

Short course on High Performance Computing

Victor Eijkhout

2018

Processor Architecture

Modern processor
Memory hierarchy: caches, register, TLB.

- Structure of a modern processor

Programming strategies for parallelism

The power question

- Memory hierarchy: caches, register, TLB.

The SIMD/MIMD/SPMD/SIMT model for parallelism

- Multicore issues

Interconnects and topologies, theoretical concepts

Programming models

- Programming strategies for performance

Load balancing; locality, space-filling curves

First we dig into bits

- The power question

Integers

Floating point numbers

Floating point math

Examples

More

Parallelism

Essential aspects of LU factorization

- Basic concepts

Sparse matrices: storage and algorithms

Collective methods, basic concepts and available methods

Collectives as building blocks; complexity

- Theoretical concepts

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

- The SIMD/MIMD/SPMD/SIMT model for parallelism

Complete SIMD and MIMD methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

- Characterization of parallelism by memory model

Multicore block algorithms

N-body problems: naive and equivalent formulations

- Interconnects and topologies, theoretical concepts

Derived datatypes

Communicator manipulation

Blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

- Load balancing, locality, space-filling curves

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Computational aspects of parallelization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix vector product

Sparse matrix vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Justification

High Performance Computing is a field that brings together algorithms, software, and hardware. This course conveys the basics of computer architecture, scientific algorithms, and how to code algorithms to make them efficient on current hardware.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More

Processor Architecture

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks, complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency Hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Justification

The performance of a parallel code has as one component the behaviour of the single processor or single-threaded code. In this section we discuss the basics of how a processor executes instructions, and how it handles the data these instructions operate on.

Processor Architecture

- **Structure of a modern processor**

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

- **Memory hierarchy: caches, register, TLB.**

Interconnects and topologies; theoretical concepts

Programming models

Load balancing, locality, space-filling curves

- **Multicore issues**

First we dig into bits

Integers

- **Programming strategies for performance**

Floating point numbers

Floating point math

Examples

More

- **The power question**

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Parallelism 85

Iterative methods: basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

- **Basic concepts**

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

- **Theoretical concepts**

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

- **The SIMD/MIMD/SPMD/SIMT model for parallelism**

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Programming models

Table of Contents

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Structure of a modern processor

more

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD SIMD model, memory issues
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More

The ideal processor:

- **(Stored program)**
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
- **An instruction contains the operation and two operand locations**
LU factorization, pivoting, complexity
Scalability analysis of dense matrix-vector product
- **Processor decodes instruction, gets operands, computes and writes back the result**
Latency hiding / communication minimizing
Incomplete factorizations, iterative methods
Parallel LU through nested dissection
- **Repeat**
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Von Neumann machine

Structure of a modern processor

Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The actual state of affairs

The SIMD/MIMD/SIMT paradigm continues

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

- Single instruction stream versus multiple cores / floating point units

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, direct methods and hybrid methods

Collectives as building blocks; complexity

- Single instruction stream versus Instruction Level Parallelism

- Unit time addressable memory versus large latencies

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Computational aspects of direct methods

Incomplete approaches to matrix factorization

Fast matrix multiplication, wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Modern processors contain lots of magic to make them seem like Von Neumann machines.

Memory management, prefetching, wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concerns
The SIMD/MIMD/SPMD/SIMT model
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Traditional: processor speed was paramount. Operation counting.

Integers
Floating point numbers
Floating point math
Examples

Nowadays: memory is slower than processors (peak performance only out of register).

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods

This course

Study data movement aspects
Dealing with latency
Algorithm design for processor reality

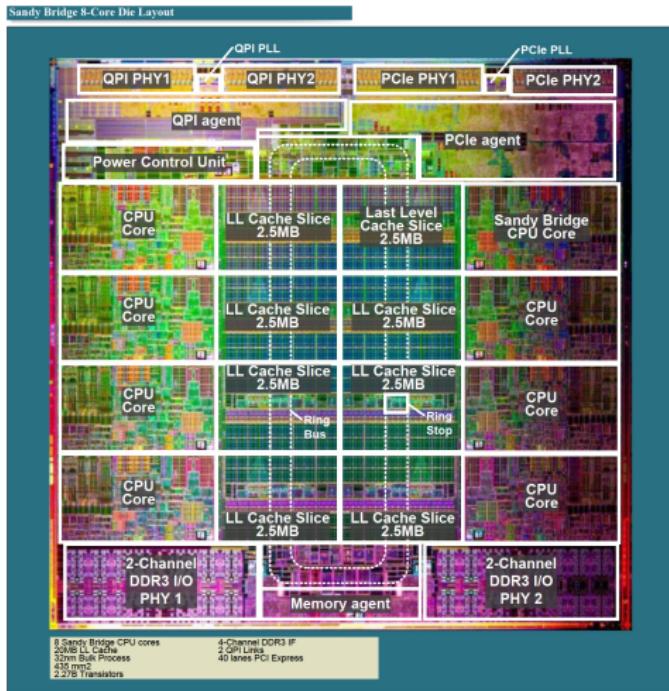
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor

Memory hierarchy: caches, register, TLB.

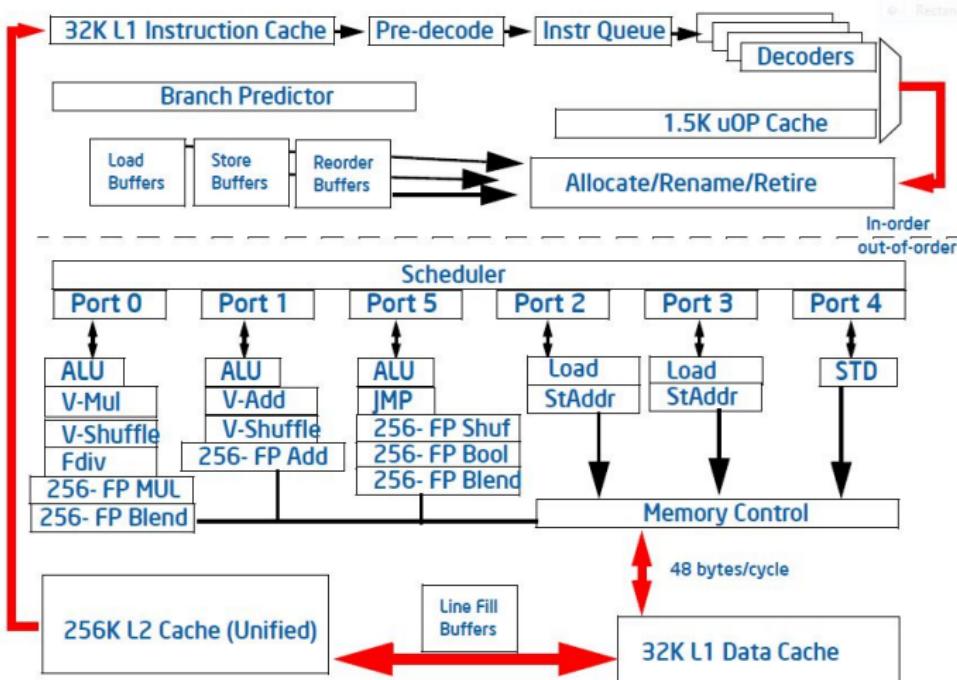
Multicore issues

A first look at a processor



Copyright (c) 2011 Hervé Goto All rights reserved.

Structure of a core



Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

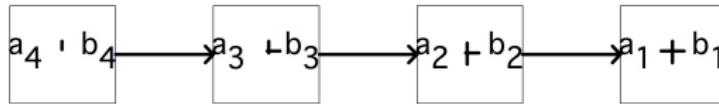
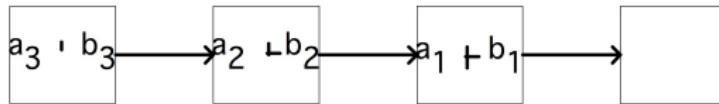
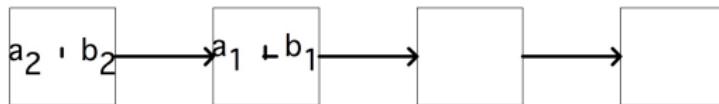
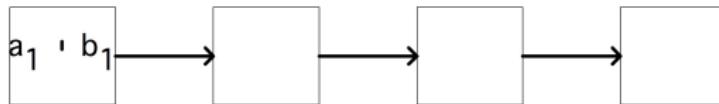
Parallel computation

The SIMD/MIMD/SPMD/SIMT model for parallelism

Pipelining, pictorially

The SIMD/MIMD/SPMD/SIMT model for parallelism

$$c_i \leftarrow a_i + b_i$$



F

Profili

1

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SPMT model and available methods
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models

- Decoding the instruction operands.

- Data fetch into register
 - Integers
 - Floating point numbers
 - Floating point math
- Aligning the exponents:
 - Examples
 - More

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks: complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product

$35 \times 10^{-1} + .6 \times 10^{-2}$ becomes
 $.35 \times 10^{-1} + .06 \times 10^{-1}$.

- Adding mantissas, giving .41.
- Normalizing the result, giving $.41 \times 10^{-1}$.

Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems, naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems
Derived datatypes

pipeline stages

Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor

Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage, arithmetic, examples

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Implementation techniques: matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics; interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

One-sided communication

$n_{1/2}$: value for which speedup is $\ell/2$

Profiling and debugging; optimization and programming strategies.

Analysis

Operation timing:

$$\left\{ \begin{array}{l} n \text{ operations} \\ \ell \text{ number of stages} \\ \tau \text{ clock cycle} \end{array} \right. \Rightarrow t(n) = n\ell\tau$$

With pipelining:

$$t(n) = [s + \ell + n - 1]\tau$$

where s is a setup cost

Implementation techniques: matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics; interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

One-sided communication

Structure of a modern processor

Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Processor design

Applicability of pipelining

The SIMD/MIMD/SPMD/SIMT model for parallelism

Processor parallelism by memory model

Interconnects and topologies, theoretical concepts

Pipelining works for: vector addition

Load balancing, locality, space-filling curves

First we dig into bits

Pipelining does not work for: recurrences

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

for (i) {
 Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

}

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Transform:
Parallelism, implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix multiplication

Derived datatypes

Communicator management

Non-blocking collectives

One-sided communication

$$x_{n+2} = a_{n+1}x_{n+1} + b_{n+1}$$

$$\text{Derived datatypes} = a_{n+1}(a_nx_n + b_n) + b_{n+1}$$

$$\text{Communicator management} = \overline{a_{n+1}}a_nx_n + a_{n+1}b_n + b_{n+1}$$

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model of parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math

- **Instruction-Level Parallelism:** more general notion of independent instructions

Essential aspects of LU factorization
Sparse Matrices: storage and algorithms

- Iterative methods, basic concepts and available methods
- Requires independent instructions

Scalability analysis of dense matrix-vector product
Sparse matrix-vector products

- As frequency goes up, pipeline gets longer: more demands on compiler

Computational aspects of iterative methods
Parallel LU through nested dissection

- Incomplete approaches to matrix factorization
- Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

- Graph analytics, interpretation as sparse matrix problems
- Derived datatypes

Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

Instruction pipeline

Structure of a modern processor

Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MMW SIMD/SIMT paradigm

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

More

- multiple-issue of independent instructions

Execution patterns of ultra-parallelization

Sparse matrices: storage and algorithms

Iterative methods: basic concepts and available methods

Collectives as building blocks, complexity

Scalability analysis of dense matrix-vector product

- out-of-order execution

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Computational aspects of iterative methods

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Instruction-Level Parallelism

- branch prediction and speculative execution
- prefetching

Problems: complicated circuitry, hard to maintain performance

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor

Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model from LISA

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix vector product

— Processor tries to predict branches

— *branch misprediction penalty:*

pipeline needs to be flushed and refilled

— avoid conditionals in inner loops!

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Implications

- Long pipeline needs many independent instructions:
demands on compiler
- Conditionals break the stream of independent instructions
 - Processor tries to predict branches
 - branch misprediction penalty:
pipeline needs to be flushed and refilled
 - avoid conditionals in inner loops!

- **Clock frequency**

- **SIMD width**

- **Load/store unit behaviour**

Behaviour (out of L1):

Processor	year	add/mult/fma units	daxpy cycles
MIPS R10000	1996	$1 \times 1 + 1 \times 1 + 0$	8/24
Alpha EV5	1996	$1 \times 1 + 1 \times 1 + 0$	8/12
IBM Power5	2004	$0 + 0 + 2 \times 1$	4/12
AMD Bulldozer	2011	$2 \times 2 + 2 \times 2 + 0$	2/4
Intel Sandy Bridge	2012	$1 \times 4 + 1 \times 4 + 0$	2/4
Intel Haswell	2014	$0 + 0 + 2 \times 4$	1/2

Processor	year	(count \times width)	(arith vs load/store)
MIPS R10000	1996	$1 \times 1 + 1 \times 1 + 0$	8/24
Alpha EV5	1996	$1 \times 1 + 1 \times 1 + 0$	8/12
IBM Power5	2004	$0 + 0 + 2 \times 1$	4/12
AMD Bulldozer	2011	$2 \times 2 + 2 \times 2 + 0$	2/4
Intel Sandy Bridge	2012	$1 \times 4 + 1 \times 4 + 0$	2/4
Intel Haswell	2014	$0 + 0 + 2 \times 4$	1/2

Floating point capabilities of several processor architectures, and DAXPY cycle number for 8 operands

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor

Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks, complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Dirty secret

Processor design is sometimes optimized for certain algorithms

In particular, DGEMM/Linpack

Favourable property: one load per operation, no stores

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Table of Contents

Processor Architecture

- Structure of a modern processor

The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model

- Memory hierarchy: caches, register, TLB.

Interconnects and topologies; theoretical concepts
Programming models

Load balancing, locality, space-filling curves

- Multicore issues

First we dig into bits

Integers

- Programming strategies for performance

Floating point numbers

Floating point math

Examples

More

- The power question

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Parallelism 85

- Parallel methods: basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

- Basic concepts

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

- Theoretical concepts

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

- The SIMD/MIMD/SPMD/SIMT model for parallelism

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

- Characterization of parallelism by memory model

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Programming models

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Memory hierarchy: caches, register, TLB.

more

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Caches

More memory system topics

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Complexities as scaling, cost, complexity

Scalability analysis of dense matrix-vector product

Computational aspects of iterative methods

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

The Big Story

- DRAM memory is slow, so let's put small SRAM close to the processor
 - More
 - Caches
 - More memory system topics
- This helps if data is reused
 - Complexity as scaling, cost, complexity
 - Scalability analysis of dense matrix-vector product
 - Computational aspects of iterative methods
 - Latency hiding / communication minimizing
 - Parallel LU through nested dissection
 - Incomplete approaches to matrix factorization
- Does the algorithm have reuse?
- Does the implementation reuse data?

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SIMD/MIMD model; parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collective communication: complexity

Scalability analysis of dense matrix-vector product

- **latency** is delay between request for data and availability

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Registers

Caches

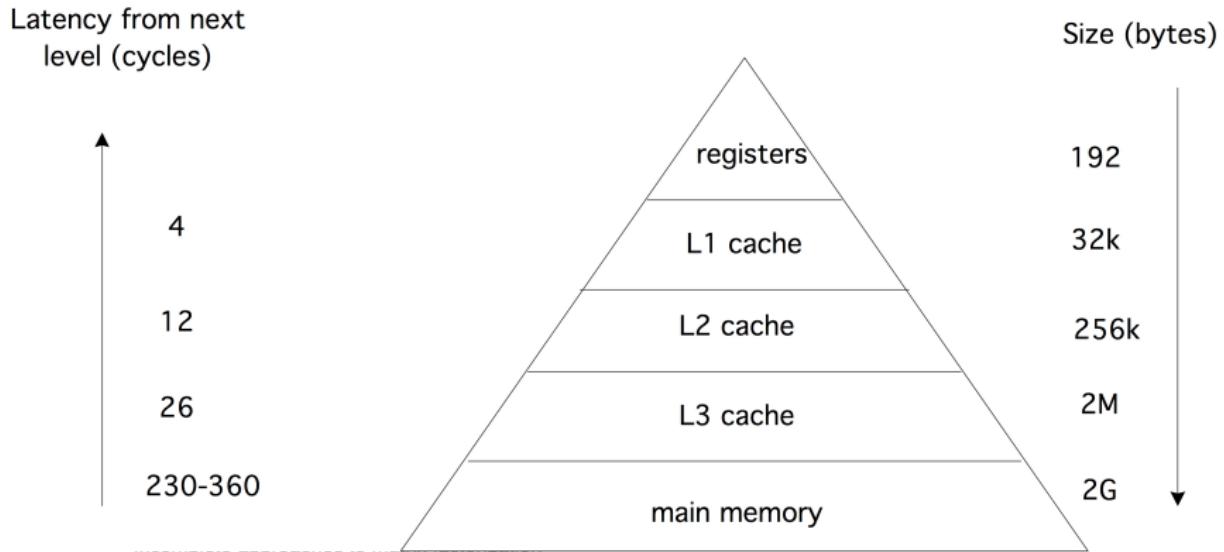
More memory system topics

Bandwidth and latency

Important theoretical concept:

- **bandwidth** is rate at which data arrives thereafter

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts



Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Registers

More
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Registers
Caches
More memory system topics

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/PD/SPMD/IMC/UMC model

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Dense matrix-vector blocking / complexity

Scalability analysis of dense matrix-vector product

Sparsity patterns: triangular factorization

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism in the implementation of sparse matrix-vector multiplication

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

$a := b + c$

Registers

Caches

More memory system topics

- load the value of b from memory into a *register*,
- Scalability analysis of dense matrix-vector product
- load the value of c from memory into another register,
- Scalability analysis of sparse matrix-vector product
- compute the sum and write that into yet another register, and
- Latency hiding / communication minimizing
- Incomplete approaches to matrix factorization
- Parallel LU through nested dissection
- write the sum value back to the memory location of a .

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model and parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Assembly code

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

• Registers are named
• Iteration space partitioning, implementable methods

Collectives as building blocks; complexity

• Scalability analysis of dense matrix-vector product
• Sparse matrix-vector product

Latency hiding / communication minimizing

• Optimizing the iteration space

• Methods

Parallel LU through nested dissection

• Iterative refinement, error control, updating

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

• N-body problems, naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

• Derived data structures

Communicator manipulation

• Blocking

One-sided communication

Profiling and debugging; optimization and programming strategies.

Register usage

Registers

Caches

More memory system topics

• Can be explicitly addressed by the programmer

• ... as opposed to caches.

• Assembly coding or inline assembly (compiler dependent)

Parallelism and implicit operations: wavefronts, approximation

• ... but typically generated by compiler

Graph analytics, interpretation as sparse matrix problems

• Derived data structures

Communicator manipulation

• Blocking

One-sided communication

Structure of a modern processor

Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Cache concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

$a := b + c$ Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

$d := a + e$ Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

$t1 = \sin(\alpha) * x + \cos(\alpha)$ Essential aspects of LU factorization

Sparse matrices: storage and algorithms

$t2 = -\cos(\alpha) * x + \sin(\alpha) * y;$ Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Parallel LU through hierarchical iteration on minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

$s = \sin(\alpha); c = \cos(\alpha);$ N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

$t1 = s * x + c * y;$ Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging, optimization and programming strategies.

often done by compiler

Examples of register usage

$a := b + c$ Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

$d := a + e$ Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

$t1 = \sin(\alpha) * x + \cos(\alpha)$ Essential aspects of LU factorization

Sparse matrices: storage and algorithms

$t2 = -\cos(\alpha) * x + \sin(\alpha) * y;$ Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Parallel LU through hierarchical iteration on minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

$s = \sin(\alpha); c = \cos(\alpha);$ N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

$t1 = s * x + c * y;$ Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging, optimization and programming strategies.

often done by compiler

Registers

Caches

More memory system topics

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

Register variables

The SIMD/MIMD/SPMD/SIMT memory models

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Exceptions

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods
register double;

Computations as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Registers

Caches

More memory system topics

Hint to the compiler: declare register variable

Declaring too many leads to register spill.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Integers

Caches

More
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Registers

Caches

More memory system topics

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Balancing

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Implementation of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Arithmetics

More

Registers

Caches

More memory system topics

Fast SRAM in between memory and registers: mostly serves data reuse

$\dots = \dots \times \dots // \text{ instruction using } x$

$\dots \dots \dots // \text{ several instructions not involving } x$

$\dots = \dots \times \dots // \text{ instruction using } x$

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

- load x from memory into cache, and from cache into register; operate on it:
 - Scalability analysis of dense matrix-vector product
 - Sparse matrix-vector product
 - Latency hiding / communication minimizing
 - Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

- do the intervening instructions;
- N-body problems: naive and equivalent formulations
- Graph analytics: incorporation of sparse matrices, hashing

Derived datatypes

Parallel primitives

Non-blocking collectives

One-sided communication

Profiling and debugging: optimization and programming strategies.

Caches are associative

Cache levels

- Levels 1,2,3,(4): L1, L2, etc
- Increasing size, increasing latency, increasing bandwidth
- (Note: L3/L4 can be fairly big; beware benchmarking)
- Cache hit, cache miss: one level is consulted, then the next
- L1 has separate data / instruction cache, other levels mixed
- Caches do not have enough bandwidth to serve the processor:
coding for reuse on all levels.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model and parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point arithmetic

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding, communication minimization

Computational aspects of iterative methods

Parallelization of iterative methods

Incomplete approaches to matrix factorization

Parallelism in MPI collectives: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Cache misses

- **Compulsory miss:** first time data is referenced
- **Capacity miss:** data was in cache, but has been flushed (overwritten) by LRU policy
- **Conflict miss:** two items get mapped to the same cache location, even if there are no capacity problems
- **Invalidation miss:** data becomes invalid because of activity of another core

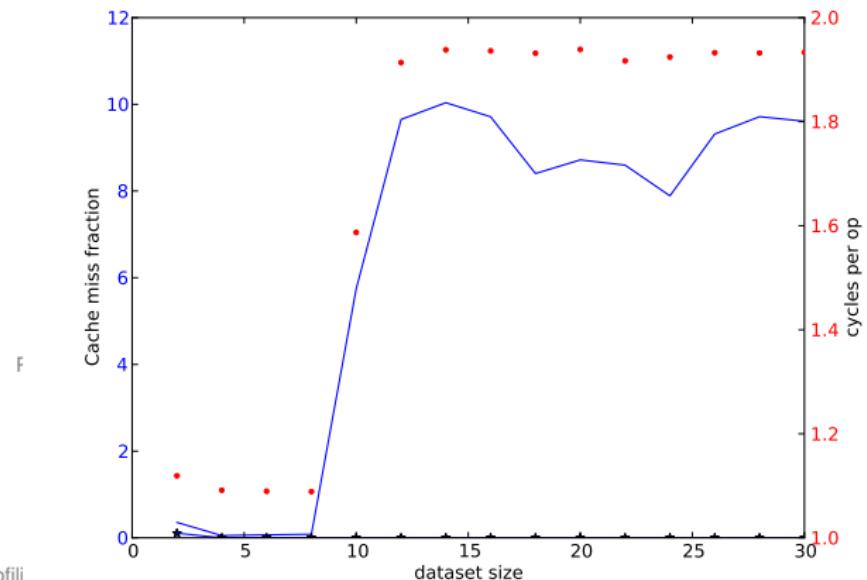
Registers

Caches

More memory system topics

```
for (i=0; i<NRUNS; i++)
    for(j=0; j<size; j++)
        array[j] = 2.3*array[j]+1.2;
    
```

Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits



Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrix factorization algorithms

Iterative methods, basic concepts and available methods

4 or 8 words

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Cache line transfer costs bandwidth

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through Nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Cache lines

- **Memory requests go by byte or word**

Registers

Caches

more memory system topics

- **Memory transfers go by cache line.**

Iterative methods, basic concepts and available methods

4 or 8 words

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

- **Cache line transfer costs bandwidth**

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through Nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor

Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

... = ... $x[i]$ First we dig into bits

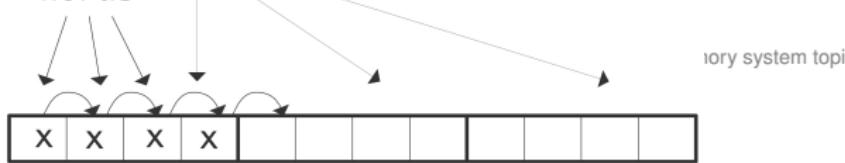
Floating point numbers

Integers

Memory system topics

for ($i=0$, $i < N$, $i++$)

words cachelines



Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and multithreaded operations: what, how, and motivation

Multicore block algorithms

... N-body problems: native and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

for ($i=0$, $i < N$, $i += \text{stride}$)

Profiling



Structure of a modern processor

Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The cache effect

Cache replacement

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory bandwidth

Interconnects and topologies, theoretical concepts
Programming models

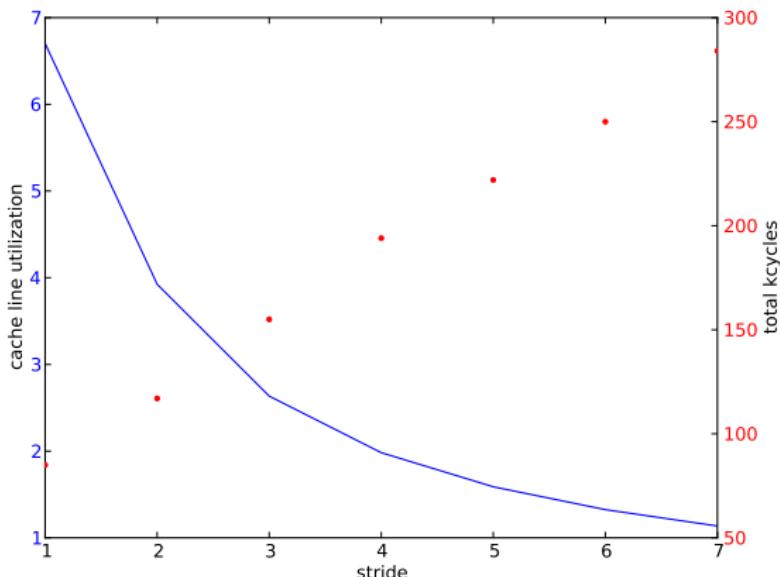
Load balancing, locality, space-filling curves

First we dig into bits

Stride effects

for ($i=0, n=0; i < 11\text{WORDS}; i++, n+=\text{stride}$)

array[n] = 2.3 * array[n] + 1.2;



Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/HMW/OMP/SIMD/multithread parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Matrix-vector multiplication, communication

Scalability analysis of dense matrix-vector product

(for instance from same cacheline)

Latency hiding / communication minimizing

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Spatial and temporal locality

**Temporal locality: use an item, use it again but from cache
efficient because second transfer cheaper.**

Registers

Registers

More memory system topics

Spatial locality: use an item, then use one 'close to it'

Scalability analysis of dense matrix-vector product

(for instance from same cacheline)

**efficient because item is already reachable even though not used
before.**

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT Model of Computation

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

- Ideal: any address can go anywhere; LRU policy for replacement

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks, complexity

Scalability analysis of dense matrix-vector product

- pro: optimal; con: slow, expensive to manufacture

Latency hiding / communication minimizing

- Simple: direct mapping by truncating addresses

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and loop iterations: cache friendly code

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Registers

Caches

More memory system topics

Cache is smaller than memory, so we need a mapping scheme

- pro: fast and cheap; con: I'll show you in a minute
- Practical: limited associativity; golden mean

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

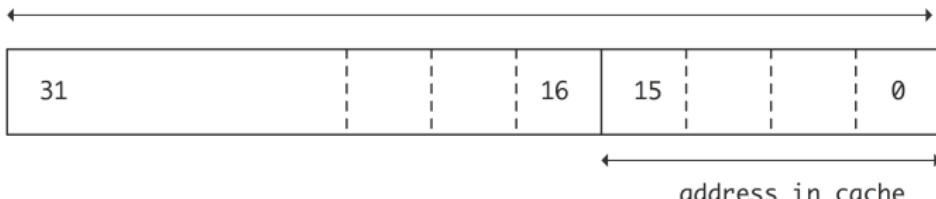
Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model of parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

address in memory



Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building-blocks: complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

- Use last number of bits to find cache address

Incomplete approaches to matrix factorization

Parallelism and implicit operations, local vs global approaches

Multicore block algorithms

N-body problems, basic and equilibrium simulations

Graph analytics, interpretation as sparse matrix problems

Memory locality, cache friendly code

Communicator manipulation

Collectives, MPI, OpenMP, OpenCL

One-sided communication

Profiling and debugging; optimization and programming strategies.

address in cache

more memory system topics

Direct mapping of 32-bit addresses into a 64K cache

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building-blocks: complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

- If (memory) addresses are cache size apart, they get mapped to the same cache location

Incomplete approaches to matrix factorization

Parallelism and implicit operations, local vs global approaches

Multicore block algorithms

N-body problems, basic and equilibrium simulations

Graph analytics, interpretation as sparse matrix problems

Memory locality, cache friendly code

Communicator manipulation

Collectives, MPI, OpenMP, OpenCL

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The problem with direct mapping

The SIMD/loop/BLAS/SVD/FFT parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

real*8 A(8192,3);

do i=1,512

 Essential aspects of LU factorization

 Sparse matrices: storage and algorithms

 Iterative methods, basic concepts and available methods

end do Collectives as building blocks; complexity

 Scalability analysis of dense matrix-vector product

 Sparse matrix-vector product

 Latency hiding / communication minimizing

 Computational aspects of iterative methods

 Parallel LU through nested dissection

 Incomplete approaches to matrix factorization

 Parallelism and implicit operations: wavefronts, approximation

low performance Multicore block algorithms

 N-body problems: naive and equivalent formulations

 Graph analytics, interpretation as sparse matrix problems

 Derived datatypes

 Communicator manipulation

 Non-blocking collectives

 One-sided communication

Profiling and debugging; optimization and programming strategies.

Registers

Caches

More memory system topics

) / 2

In each iteration 3 elements map to the same cache location:
constant overwriting ('eviction', *cache thrasing*):

low performance

 Multicore block algorithms

 N-body problems: naive and equivalent formulations

 Graph analytics, interpretation as sparse matrix problems

 Derived datatypes

 Communicator manipulation

 Non-blocking collectives

 One-sided communication

- Allow each memory address to go to multiple (but not all) cache addresses; typically 2, 4, 8
- Prevents problems with multiple arrays
- Reasonable fast, still:
- Often lower associativity for L1 than L2, L3

Associativity	L1	L2
Parallel LU through nested dissection		
Incomplete approaches to matrix factorization		

Parallelism and implicit operations: wavefronts, approximation

Intel (Woodcrest) 8 8 Multicore block algorithms

N-body problems: naive and equivalent formulations

AMD (Bulldozer) 2 8 Coarse-grained parallelism as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor

Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Two-level concepts

Illustration of associativity

{0, 12, 24, ... }

{1, 13, 25, ... }

{2, 14, 26, ... }

{3, 15, 27, ... }

{4, 16, 28, ... }

{5, 17, 29, ... }

{6, 18, 30, ... }

{7, 19, 31, ... }

{8, 20, 32, ... }

{9, 21, 33, ... }

{10, 22, 34, ... }

{11, 23, 35, ... }

{0, 12, 24, ... } {4, 16, 28, ... }

{8, 20, 32, ... }

{1, 13, 25, ... } {5, 17, 29, ... }

{9, 21, 33, ... }

{2, 14, 26, ... } {6, 18, 30, ... }

{10, 22, 34, ... }

{3, 15, 27, ... } {7, 19, 31, ... }

{11, 23, 35, ... }

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Two caches of 12 elements: direct mapped (left) and 3-way associative (right)

Graph partitioning, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

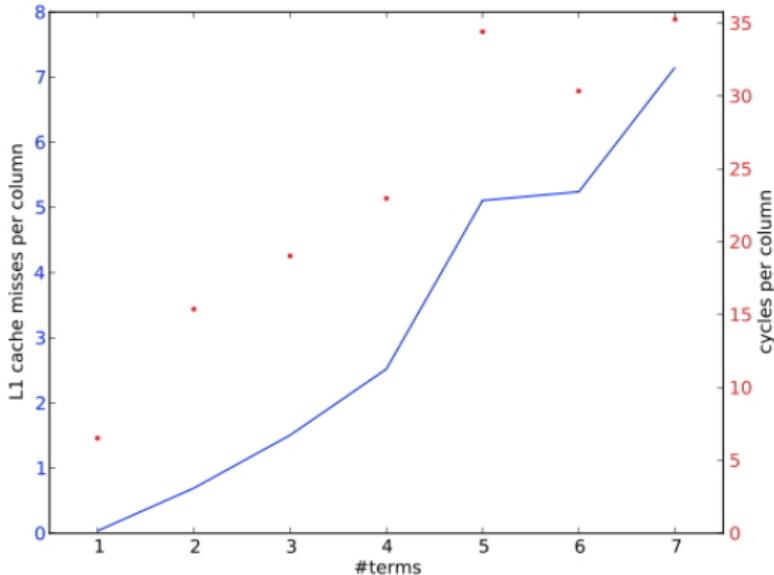
Direct map: 0–12 is conflict

Profiling and debugging: optimization and programming strategies.

Associative: no conflict

Associativity in practice

$$\forall j: y_j = y_j + \sum_{i=1}^m x_{i,j}$$



One-sided communication

Profiling and debugging; optimization and programming strategies.

The number of L1 cache misses and the number of cycles for each j

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues
Programming strategies for multicore

The power question

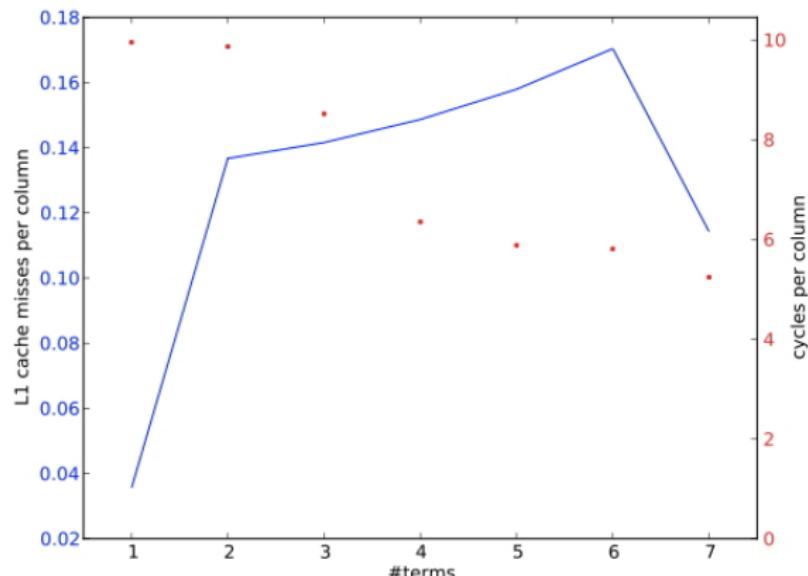
Basic concepts

Theoretical concepts

The SIMD/MIMD/COMB/CIMT model for parallelism

One remedy

Do not user powers of 2.



Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

The number of L1 cache misses and the number of cycles for each j column accumulation, vector length $4096 + 8$

Profiling and debugging, optimization, and programming strategies

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

More memory system topics

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Registers
Caches
More memory system topics

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/MPMD paradigm

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks, complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Parallel LU through recursive methods

Parallelism and implicit operations: wavefronts, approximation

- **bandwidth shared between cores**

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Bandwidth / latency

Simple model for sending n words:

$$t = \alpha + \beta n$$

Registers

Caches

More memory system topics

Quoted bandwidth figures are always optimistic:

- **bandwidth shared between cores**

Computational aspects of iterative methods

- **bandwidth wasted on coherence**

Parallel LU through nested dissection

Parallel LU through recursive methods

Parallelism and implicit operations: wavefronts, approximation

- **assumes optimal scheduling of DRAM banks**

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

Finite dig into bits
Integers

Floating point numbers

FIE, gap, width

Examples

- Do you have to wait for every item from memory?

Registers
Caches

More memory system topics

- Memory controller can infer streams: prefetch

- Sometimes controllable through assembly, directives, libraries

(AltiVec)

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

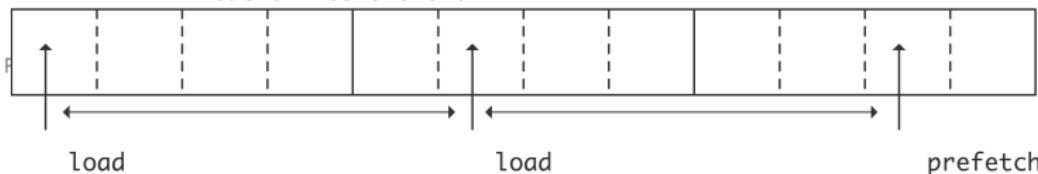
Collectives as building blocks: complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

cachelines . . .



Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model; parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

- Translation between logical address, as used by program, and physical in memory

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Matrix-vector multiplication, including

Computational aspects of iterative methods

Conjugate gradient, incomplete Cholesky

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Memory pages

Memory is organized in pages:

Registers

Caches

More memory system topics

- This serves virtual memory and relocatable code
- so we need another translation stage.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SPS model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Matrix-matrix multiplication with LU

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Lateness hiding; communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Page translation: TLB

- General page translation: slowish but extensive

Essential aspects of LU factorization

Matrix-matrix multiplication with LU

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Lateness hiding; communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Registers

Caches

More memory system topics

- Translation Look-aside Buffer (TLB) is a small list of frequently used pages

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Lateness hiding; communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

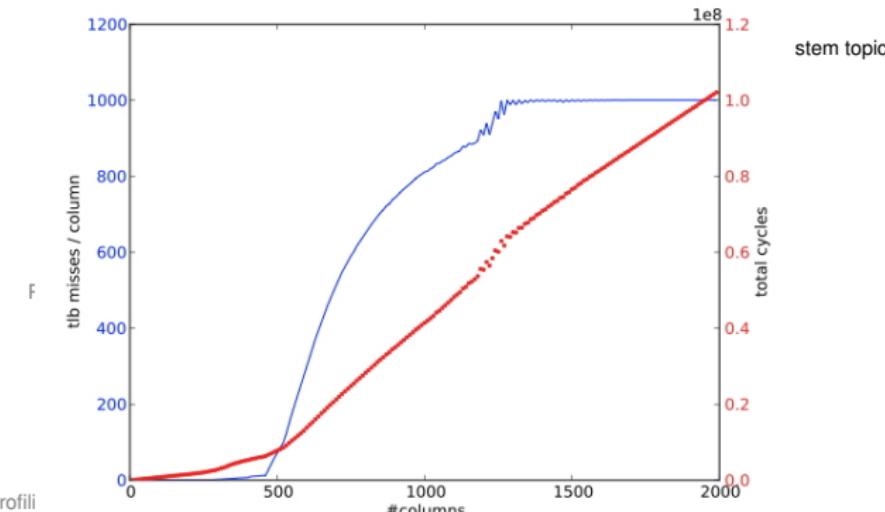
```
#define INDEX(i,j,m,n) i+j*m
```

```
array = (double*) malloc(m*n*sizeof(double));
```

```
/* traversal #2 */
```

```
for (i=0; i<m; i++)
```

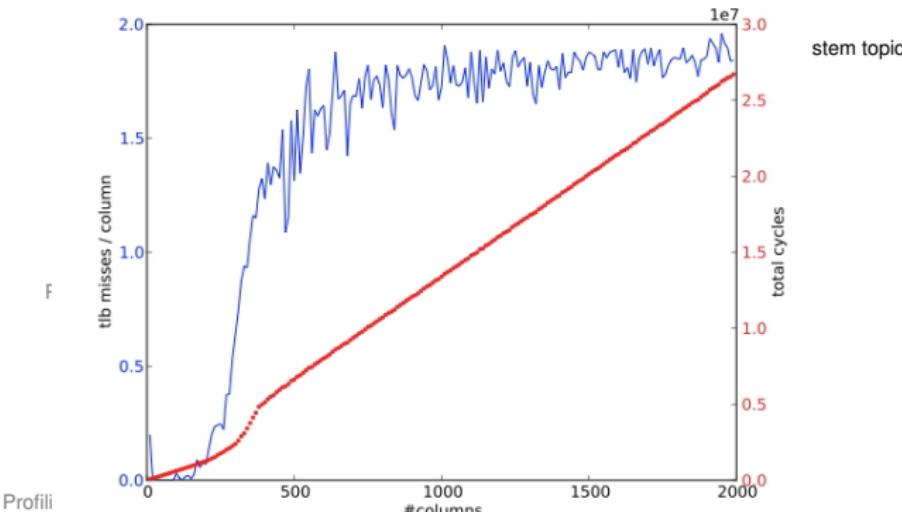
```
    array[INDEX(i,j,m,n)] |= array[INDEX(i,j,m,n)]+1;
```



TLB hits

```
#define INDEX(i,j,m,n) i+j*m
```

```
/* traversal #1 */
```



stem topics

Structure of a modern processor

Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power of SIMD

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

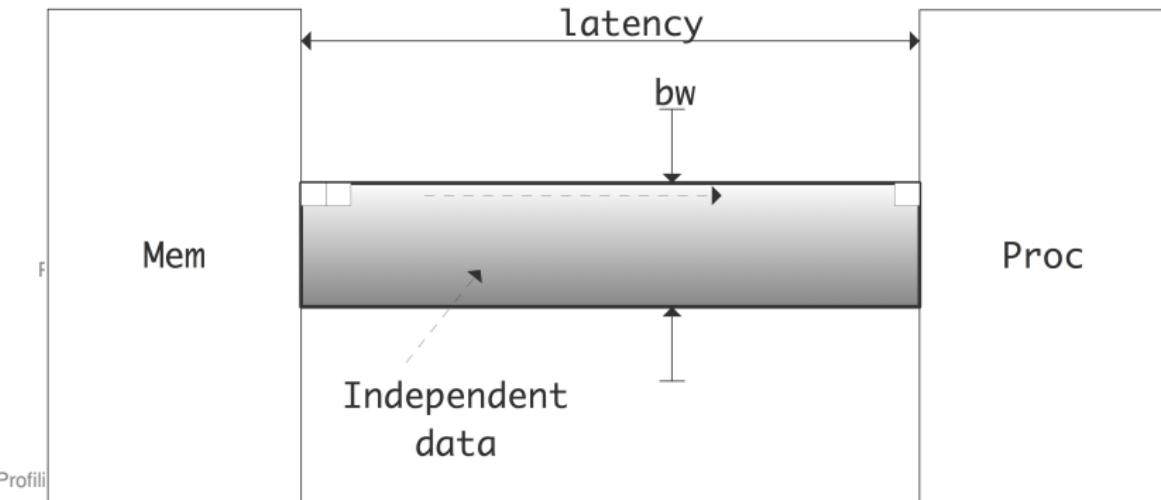
Load balancing, locality, space-filling curves

Programming models

Little's Law

- Item loaded from memory, processed, new item loaded in response
- But this can only happen after latency wait
- Items during latency are independent, therefore

$$\text{Concurrency} = \text{Bandwidth} \times \text{Latency}.$$



- Structure of a modern processor
 - The SIMD/MIMD/SPMD/SIMT model for parallelism
 - Characterization of parallelism by memory model

- Memory hierarchy: caches, register, TLB.
 - Interconnects and topologies; theoretical concepts
 - Programming models

Load balancing, locality, space-filling curves

- Multicore issues
 - First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

- The power question
 - Essential aspects of LU factorization
 - Sparse matrices: storage and algorithms

Iterative methods; basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

- Basic concepts
 - Sparse matrix-vector product
 - Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

- The SIMD/MIMD/SPMD/SIMT model for parallelism
 - Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Programming models

Processor Architecture

- Structure of a modern processor
 - The SIMD/MIMD/SPMD/SIMT model for parallelism
 - Characterization of parallelism by memory model

Interconnects and topologies; theoretical concepts

Programming models

Load balancing, locality, space-filling curves

- Multicore issues
 - First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

- The power question
 - Essential aspects of LU factorization
 - Sparse matrices: storage and algorithms

Iterative methods; basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

- Basic concepts
 - Sparse matrix-vector product
 - Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

- The SIMD/MIMD/SPMD/SIMT model for parallelism
 - Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Programming models

Parallelism

85

- Basic concepts
 - Sparse matrix-vector product
 - Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

- The SIMD/MIMD/SPMD/SIMT model for parallelism
 - Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Programming models

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers

Multicore issues

more
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model of parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Quest for higher performance:

- Two cores at half speed more energy-efficient than one at full speed.
 - Essential aspects of LU factorization
 - Sparse matrices: storage and algorithms
- Not enough instruction parallelism for long pipelines
 - Collectives as building blocks, complexity
 - Scalability analysis of dense matrix-vector product
 - Sparse matrix-vector product
 - Latency hiding / communication minimizing
 - Computational aspects of iterative methods
 - Parallel LU through nested dissection
 - Incomplete approaches to matrix factorization

Multicore solution:

- More theoretical performance
 - Parallelism and implicit parallelism before the programming time
 - Multicore block algorithms

- Burden for parallelism is now on the programmer
 - N-body problems: naive and equivalent formulations
 - Graph analysis; interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Why multicore

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

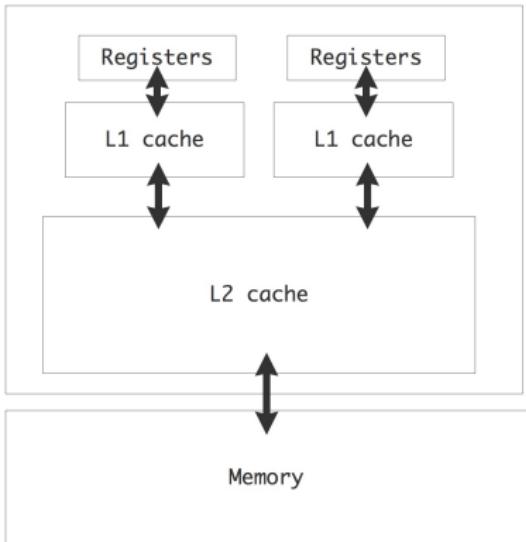
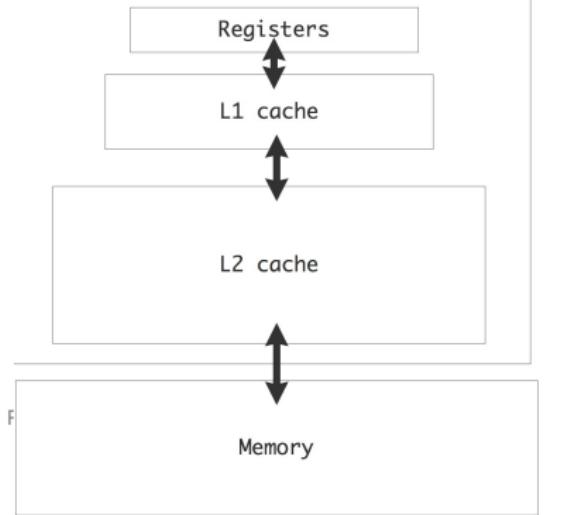
Theoretical concepts

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves



Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

IEEE standard

Examples

More

Modified-Shared-Invalid (MSI) coherence protocol:

Modified: the cacheline has been modified

Iterative methods, basic concepts and available methods

Shared: the line is present in at least one cache and is unmodified.

Scalability analysis of dense matrix-vector product

Parallel matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Cache coherence

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

Coherence issues

The SIMD/MIMD/SPMD/SIMT model and its variants

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

- Iterative methods: Jacobi, Gauss-Seidel, and conjugate gradient

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Lattices: error reduction, lattice sieving

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT memory parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing locality space filling theory

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

- Sandy Bridge core can absorb 300 GB/s

Sparse matrices: storage and algorithms

Iterative methods, matrix-vector product, collective methods

Collectives as building blocks; complexity

- It gets worse: latency 80ns, bandwidth 51 GB/s,
Little's law: parallelism 64 cache lines

Scalability analysis of sparse matrix-vector product

Sparse matrix-vector product

Latency reduction, communication minimization

Computational aspects of iterative methods

- However, each core only has 10 line fill buffers,

so we need 6–7 cores to provide the data for one core

- Power: cores are 72%, uncore 17, DRAM 11.

Parallelism and implicit operations: wavefronts, approximation

- Core power goes 40% to instruction handling, not arithmetic

N-body problems: naive and equivalent formulations

Graph analytics: interpretation, dense matrix problems

Derived datatypes

Graph structure manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Balance analysis

- Parallelism and implicit operations: wavefronts, approximation
- Core power goes 40% to instruction handling, not arithmetic
- Time for a redesign of processors and programming; see my research presentation

Processor Architecture

- Structure of a modern processor

- Memory hierarchy: caches, register, TLB.

- Multicore issues

- The power question

- Parallelism

- Basic concepts

- Theoretical concepts

- The SIMD/MIMD/SPMD/SIMT model for parallelism

- Interconnects and topologies; theoretical concepts

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues

Programming strategies for performance

- The power question
- Basic concepts
- Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

- Characterization of parallelism by memory model
- Interconnects and topologies, theoretical concepts
- Programming models
- Load balancing, locality, space-filling curves
- First we dig into bits

Programming strategies for performance

more

- Essential aspects of LU factorization
- Sparse matrices: storage and algorithms
- Iterative methods, basic concepts and available methods
 - Collectives as building blocks; complexity
 - Scalability analysis of dense matrix-vector product
 - Sparse matrix-vector product
 - Latency hiding / communication minimizing
 - Computational aspects of iterative methods
 - Parallel LU through nested dissection
 - Incomplete approaches to matrix factorization
- Parallelism and implicit operations: wavefronts, approximation
 - Multicore block algorithms
- N-body problems: naive and equivalent formulations
- Graph analytics, interpretation as sparse matrix problems
 - Derived datatypes
 - Communicator manipulation
 - Non-blocking collectives
 - One-sided communication
- Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD SIMD/SPMD model and its evolution

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

- Processor peak performance: absolute limit

Sparse matrices: storage and algorithms

- Bandwidth: linear correlation with performance

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Multi-level algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

How much performance is possible?

Performance limited by

- Processor peak performance: absolute limit

- Bandwidth: linear correlation with performance

Arithmetic intensity: ratio of operations per transfer

If AI high enough: processor-limited
otherwise: bandwidth-limited

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

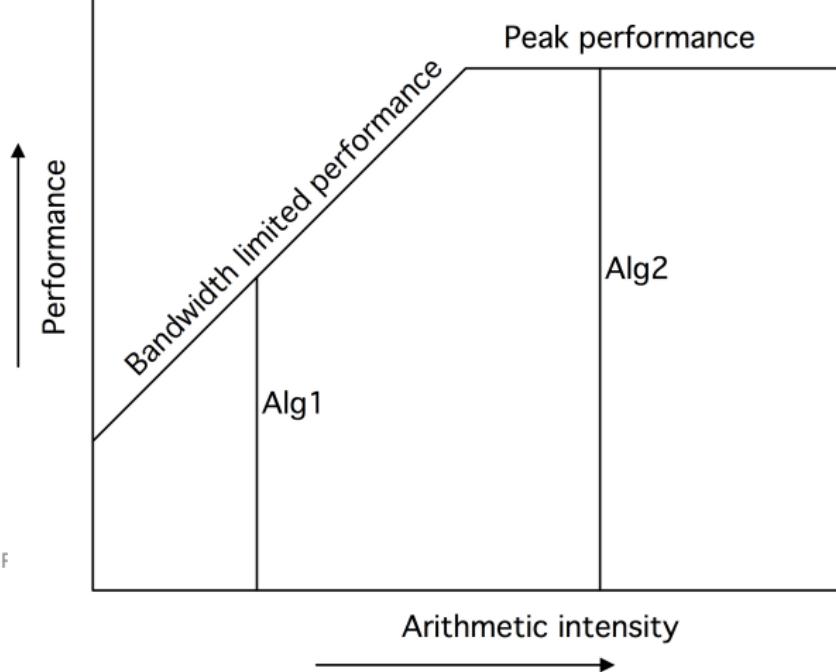
Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

Performance depends on algorithm:

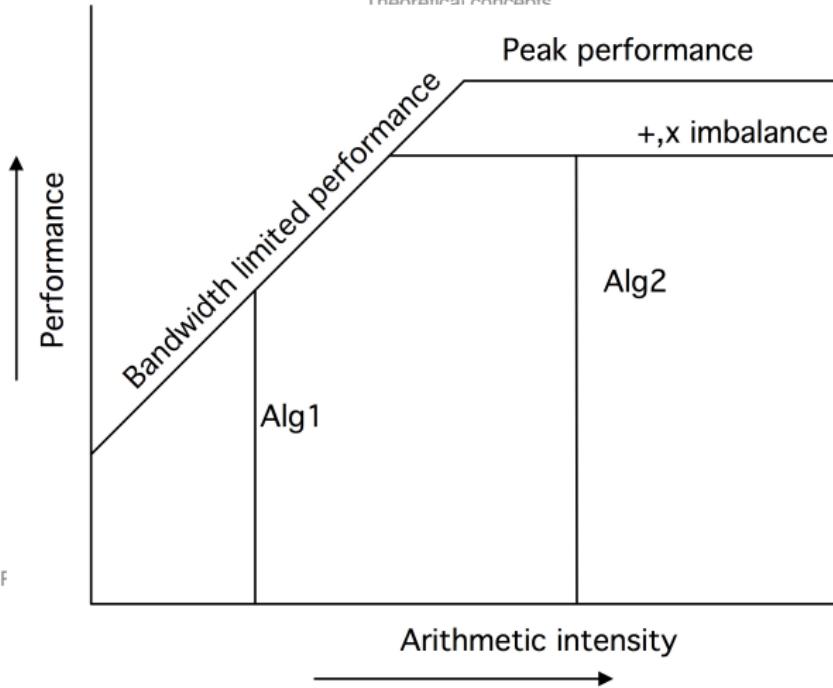
Top-down approach
Basic concepts
Theoretical concepts



Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

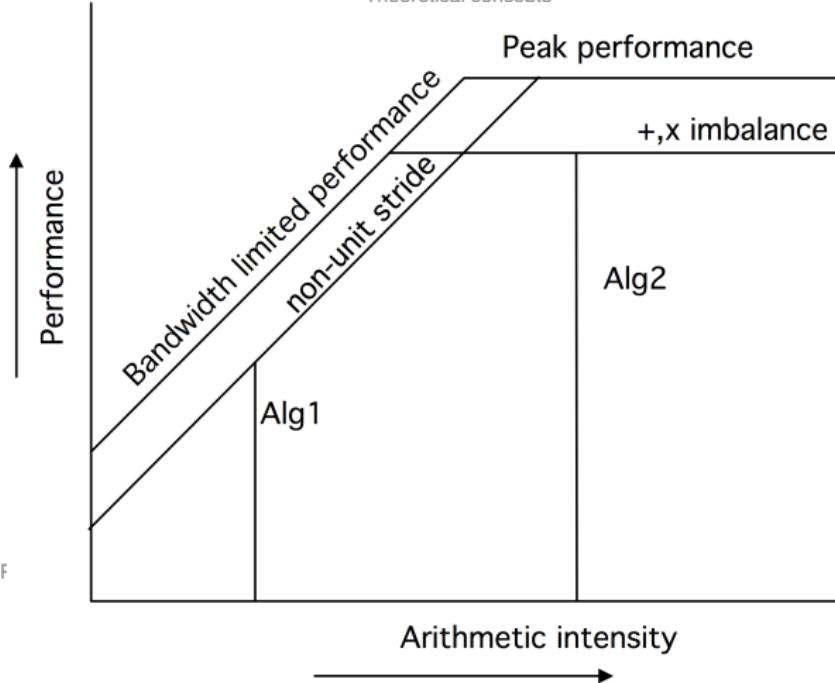
Insufficient utilization of functional units:



Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
Imperfect data transfer:
The power question
Basic concepts
Theoretical concepts



Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The GPU/IMPACT Model made for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

- Cache size: block loops

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

- pipelining and vector instructions: expose streams of instructions

Concurrent computation, loop unrolling

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU factorization, distribution

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Architecture aware programming

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

The program loop

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

for (k < small bound) Integers

Floating point numbers

for (i < N) Floating point math

Examples

x[i] = f(x[i], k, ...)

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Block to be cache contained

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

for (ii < N, ii+= blocksize)

Parallel LU through nested iteration

Incomplete approaches to matrix factorization

Parallelism and implementation: wavefront approximation

Multicore block algorithms

for (ii=ii; ii<ii+blocksize; ii++)

Graph analytics, interpretation as sparse matrix problems

x[i] = f(x[i], k, ...)

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging, multithread and multinode strategies

Loop blocking

Multiple passes over data

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

for (k < small bound) Integers

Floating point numbers

for (i < N) Floating point math

Examples

x[i] = f(x[i], k, ...)

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks: complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

for (ii < N, ii+= blocksize)

Parallel LU through nested iteration

Incomplete approaches to matrix factorization

Parallelism and implementation: wavefront approximation

Multicore block algorithms

for (ii=ii; ii<ii+blocksize; ii++)

Graph analytics, interpretation as sparse matrix problems

x[i] = f(x[i], k, ...)

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging, multithread and multinode strategies

This requires independence of operations

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

THE SIMD AND MMU model of parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage formats, algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel block and nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

• all three blocked i, j, k

Multi-level block algorithms

N-body problems: naive and equivalent formulations

• Many loop permutations, blocking factors to choose

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

The ultimate in performance programming: DGEMM

Matrix-matrix product $C = A \cdot B$

$$\sum_i \sum_j \sum_k c_{ij} + = a_{ik} b_{kj}$$

Parallelization and data distribution

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel block and nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

• Three independent loop i, j, k

Multi-level block algorithms

N-body problems: naive and equivalent formulations

• Many loop permutations, blocking factors to choose

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

The power law

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

for (k) Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

for (i) Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

for (j) Latency hiding / communication minimizing

Computational aspects of iterative methods

c[i,j] += a[i,k] * b[k,j]

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

DGEMM variant

Outer product: updates with low-rank columns-times-vector

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

for (k) Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

for (i) Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

for (j) Latency hiding / communication minimizing

Computational aspects of iterative methods

c[i,j] += a[i,k] * b[k,j]

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

Rank 1 updates

The SIMD/MIMD/SPMD/SIMT memory model, parallelism
Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

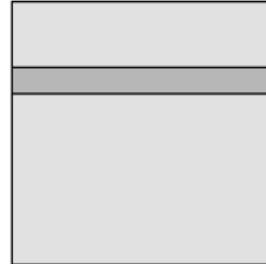
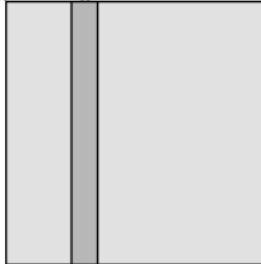
$$G_{**} = \sum_k A_{*k} B_{k*}$$

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms



N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD model for parallel computation

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

Block of A times ‘sliver’ of B



F

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model of parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

For inner i :

// compute $C[i, *]$:

for k :

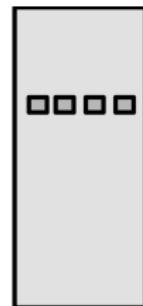
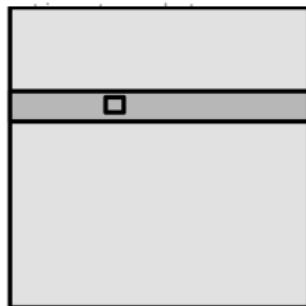
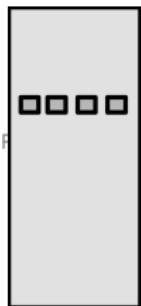
Essential aspects of LU factorization

$C[i, *] = A[i, k] * B[k, *]$

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product



One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

// compute C[i,*] : Examples
More

for k: Essential aspects of LU factorization

Sparse matrices: storage and algorithms

C[i,*] += A[i,k]* B[k,*]

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

• C[i,*] stays in register
Computing via reduction methods

Parallel LU through nested dissection

• A[i,k] and B[k,*] stream from L1

Parallelism and implicit operations: wavefronts, approximation

• blocksize of A for L2 size
Multicore block algorithms

• Multicore: involvement in quantum calculations

Graph analytics, interpretation as sparse matrix problems

• A stored by rows to prevent TLB problems
Delayed updates

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Tuning

For inner i:

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

// compute C[i,*] : Examples
More

for k: Essential aspects of LU factorization

Sparse matrices: storage and algorithms

C[i,*] += A[i,k]* B[k,*]

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

• C[i,*] stays in register
Computing via reduction methods

Parallel LU through nested dissection

• A[i,k] and B[k,*] stream from L1

Parallelism and implicit operations: wavefronts, approximation

• blocksize of A for L2 size
Multicore block algorithms

• Multicore: involvement in quantum calculations

Graph analytics, interpretation as sparse matrix problems

• A stored by rows to prevent TLB problems
Delayed updates

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SIMD/MIMD model of parallelism

Characterization of parallelism by memory model

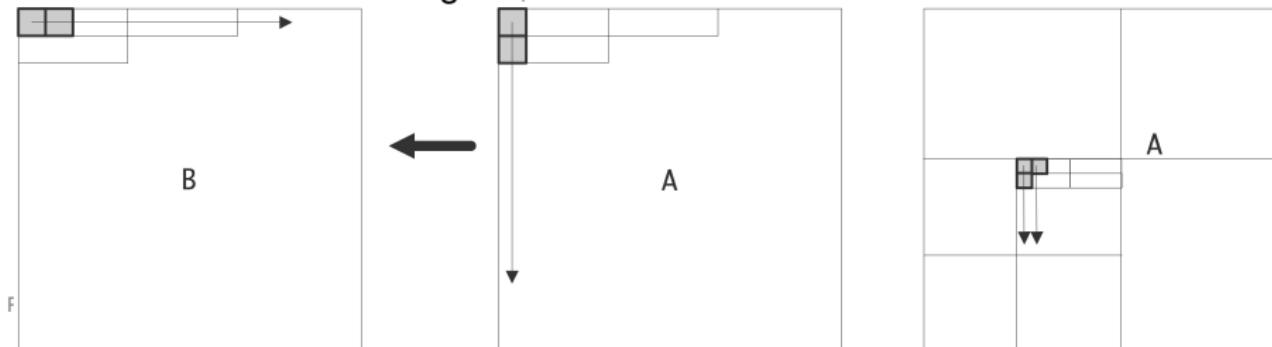
Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Observation: recursive subdivision will ultimately make a problem small / well-behaved enough



In-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Collectives as building blocks for cache

Parallel LU through nested dissection

Implementation of parallel LU factorization

with $C_{11} = A_{11}B_{11} + A_{12}B_{21}$

Recursive approach will be cache contained.

Not as high performance as being cache-aware...
Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

$$\begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}$$

Processor Architecture

Table of Contents

- Structure of a modern processor
 - The SIMD/MIMD/SPMD/SIMT model for parallelism
 - Characterization of parallelism by memory model

- Memory hierarchy: caches, register, TLB.
 - Interconnects and topologies; theoretical concepts
 - Programming models

- Multicore issues
 - Load balancing, locality, space-filling curves
 - First we dig into bits

- Programming strategies for performance
 - Integers
 - Floating point numbers
 - Floating point math
 - Examples

- The power question
 - More
 - Essential aspects of LU factorization
 - Sparse matrices: storage and algorithms

Parallelism 85

- Iterative methods: basic concepts and available methods
 - Collectives as building blocks; complexity
- Scalability analysis of dense matrix-vector product

- Basic concepts
 - Sparse matrix-vector product
 - Latency hiding / communication minimizing
 - Computational aspects of iterative methods

- Theoretical concepts
 - Parallel LU through nested dissection
 - Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

- The SIMD/MIMD/SPMD/SIMT model for parallelism
 - Multicore block algorithms

- N-body problems: naive and equivalent formulations

- Graph analytics, interpretation as sparse matrix problems

- Derived datatypes

- Communicator manipulation

- Non-blocking collectives

- One-sided communication

Profiling and debugging; optimization and programming strategies.

Programming models

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

The power question

more

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model of parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space filling curves

Scale down feature size by s :
First we dig into bits

Integers	$\sim s$
Floating point numbers	$\sim s$
Floating point math	$\sim s$
Voltage	$\sim s$
Current	$\sim s$
Frequency	$\sim s^{-1}$

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and analysis

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approximations, matrix factorizations

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: native and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

One-sided communication

Opportunity for more components, higher frequency

Profiling and debugging; optimization and programming strategies.

Dennard scaling

Feature size
Voltage
Current
Frequency

$\sim s$
 $\sim s$
 $\sim s$
 $\sim s^{-1}$

$$\text{Power} = V \cdot I \sim s^2; \text{Power density} \sim 1$$

Miracle conclusion:

Everything gets better, cooling problem stays the same

Structure of a modern processor
 Memory hierarchy: caches, register, TLB.
 Multicore issues
 Programming strategies for performance
 The power question
 Basic concepts
 Theoretical concepts
 The SIMD/MIMD/SPMD/SIMT model of parallelism
 Characterization of parallelism by memory model
 Interconnects and topologies, theoretical concepts
 Programming models
 Load balancing, locality, space-filling curves
 First we dig into bits

Charge Work Power	$q = CV$ $W = qV = CV^2$ $W/\text{time} = WF = CV^2 F$
--------------------------------	--

(1)

Sparse matrices: storage and algorithms
 Iterative methods, basic concepts and available methods
 Collectives as building blocks: complexity
 Scalability analysis of dense matrix-vector product

$$\left. \begin{aligned}
 C_{\text{multi}} &= 2C \\
 F_{\text{multi}} &= F/2 \\
 V_{\text{multi}} &= V/2
 \end{aligned} \right\} \Rightarrow P_{\text{multi}} = P/4.$$

Parallel LU through nested dissection
 Incomplete approaches to matrix factorization
 Parallelism and implicit operations: workflow approximation
 Multicore block algorithms
 N-body problems: naive and equivalent formulations
 Graph analytics: interpretation as sparse matrix problems
 Derived datatypes
 Communicator manipulation
 Non-blocking collectives
 One-sided communication
 Profiling and debugging; optimization and programming strategies.

Same computation, less power

Parallelism

- Structure of a modern processor
- Memory hierarchy: caches, register, TLB.
- Multicore issues
- Programming strategies for performance
 - The power question**
 - Basic concepts
 - Theoretical concepts
- The SIMD/MIMD/SPMD/SIMT model for parallelism
 - Characterization of parallelism by memory model
 - Interconnects and topologies, theoretical concepts
 - Programming models
 - Load balancing, locality, space-filling curves
 - First we dig into bits
 - Integers
 - Floating point numbers
 - Floating point math
 - Examples
 - More
 - Essential aspects of LU factorization
 - Sparse matrices: storage and algorithms
 - Iterative methods, basic concepts and available methods
 - Collectives as building blocks; complexity
 - Scalability analysis of dense matrix-vector product
 - Sparse matrix-vector product
 - Latency hiding / communication minimizing
 - Computational aspects of iterative methods
 - Parallel LU through nested dissection
 - Incomplete approaches to matrix factorization
 - Parallelism and implicit operations: wavefronts, approximation
 - Multicore block algorithms
 - N-body problems: naive and equivalent formulations
 - Graph analytics, interpretation as sparse matrix problems
 - Derived datatypes
 - Communicator manipulation
 - Non-blocking collectives
 - One-sided communication
 - Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and iterative Methods
Collectives as building blocks: complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency Hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Justification

Parallelism can be approached in several different ways. This session will discuss data parallelism versus instruction parallelism, issues in shared memory parallelism, parallel programming systems, the interconnects of distributed memory parallelism, scaling measures.

Processor Architecture

Table of Contents

- Structure of a modern processor

- The SIMD/MIMD/SPMD/SIMT model for parallelism

- Characterization of parallelism by memory model

- Memory hierarchy: caches, register, TLB.

- Interconnects and topologies; theoretical concepts

- Programming models

- Load balancing, locality, space-filling curves

- Multicore issues

- First we dig into bits

- Integers

- Programming strategies for performance

- Floating point numbers

- Floating point math

- Examples

- More

- The power question

- Essential aspects of LU factorization

- Sparse matrices: storage and algorithms

Parallelism 85

- Iterative methods: basic concepts and available methods

- Collectives as building blocks; complexity

- Scalability analysis of dense matrix-vector product

- Basic concepts

- Sparse matrix-vector product

- Latency hiding / communication minimizing

- Computational aspects of iterative methods

- Theoretical concepts

- Parallel LU through nested dissection

- Incomplete approaches to matrix factorization

- Parallelism and implicit operations: wavefronts, approximation

- The SIMD/MIMD/SPMD/SIMT model for parallelism

- Multicore block algorithms

- N-body problems: naive and equivalent formulations

- Graph analytics, interpretation as sparse matrix problems

- Derived datatypes

- Communicator manipulation

- Non-blocking collectives

- One-sided communication

- Profiling and debugging; optimization and programming strategies.

Programming models

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Basic concepts

more

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model of parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

- **Hardware:** vector instructions, multiple cores, nodes in a cluster.

Scalability analysis of dense matrix-vector product

- **Algorithm:** can you think of examples?

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

1 The basic idea

Parallelism is about doing multiple things at once.

Structure of a modern processor

Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The SIMD/MIMD/SPMD/SIMT model

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

for ($i=0$; $i < n$; $i++$)

Inter-elements and topologies, theoretical concepts

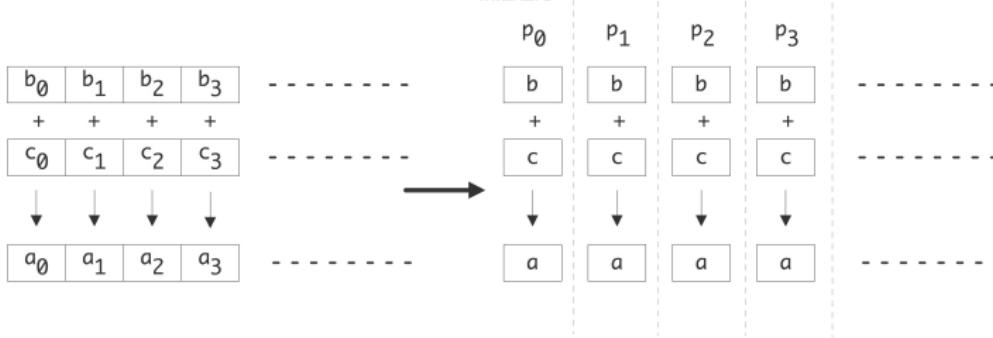
$a[i] = b[i] + c[i];$

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers



Parallel: every processing element does

for (i in my_subset_of_indices)

$a[i] = b[i] + c[i];$

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging, optimization and communication patterns

Time goes down linearly with processors

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

3 Differences between operations

The SIMD/MIMD/loop SIMD model; available parallelism
Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

for (*i*=0; *i*<*n*; *i*++)
 a[*i*] = *b*[*i*] + *c*[*i*];

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Implementation: cache blocking

Computational aspects of iterative methods

- Compare operation counts
- Compare behavior on single processor. What about multi-core?

Parallelism and implicit operations; wavefronts, approximation

- Other thoughts about parallel execution?

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

$s = 0;$

for (*i*=0; *i*<*n*; *i*++)

$s += x[i]$

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model of parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Naive algorithm

Load balancing, locality, space-filling curves

First we dig into bits

$s = 0;$ Integers

for ($i=0; i < n; i++$) Floating point numbers

$s += x[i]$ Floating point math

Examples

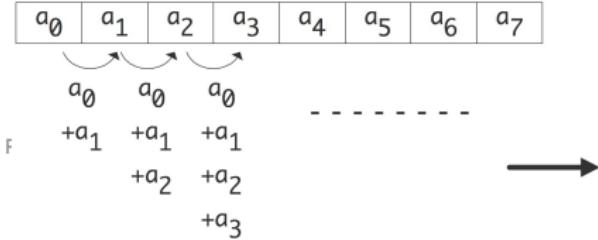
More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

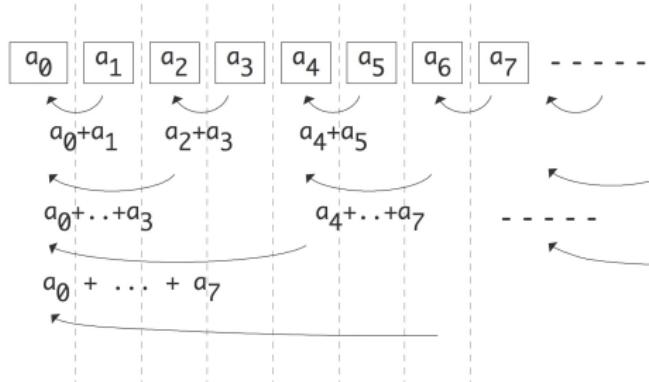
Iterative methods, basic concepts and available methods

Collectives as building blocks: complexity



Recoding

```
for (s=2; s<n; s*=2)  
  for (i=0; i<n; i+=s)  
    x[i] += x[i+s/2]
```



Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD conflict: load balancing

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

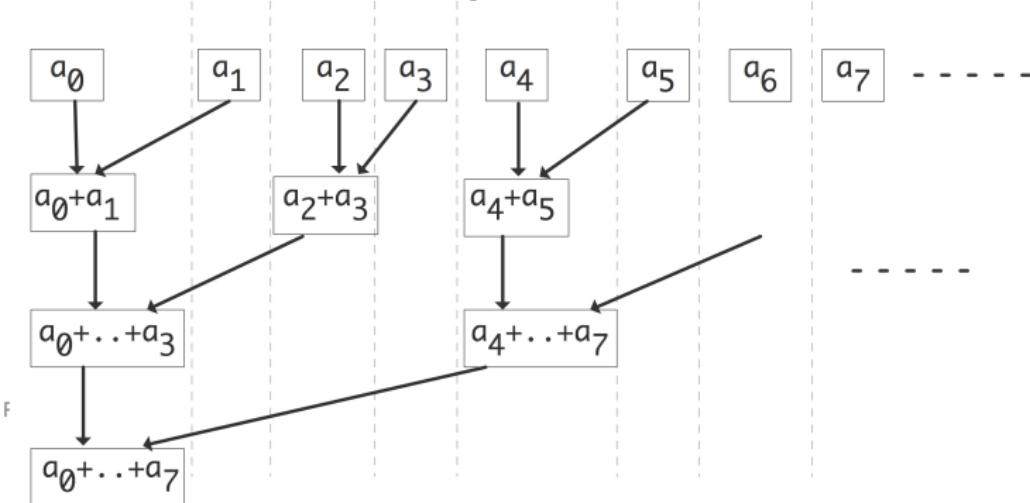
Load balancing, locality, space-filling curves

First we dig into bits

Topology of the processors

Load balancing, locality, space-filling curves

First we dig into bits



Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

increasing distance: limit on parallel speedup

Profiling and debugging; optimization and programming strategies.

Table of Contents

Processor Architecture

- Structure of a modern processor

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

- Memory hierarchy: caches, register, TLB.

Interconnects and topologies; theoretical concepts

Programming models

Load balancing, locality, space-filling curves

- Multicore issues

First we dig into bits

Integers

- Programming strategies for performance

Floating point numbers

Floating point math

Examples

Efficiency and scaling

Critical path analysis

Granularity

LU factorization analysis

- The power question

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Parallelism

85

Iterative methods: basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

- Basic concepts

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

- Theoretical concepts

Nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

- The SIMD/MIMD/SPMD/SIMT model for parallelism

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Programming models

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Theoretical concepts

more

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Critical path analysis

Granularity

LU factorization analysis

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Efficiency and scaling

more
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Critical path analysis
Granularity
LU factorization analysis

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves

- Single processor time T_1 , on p processors T_p
- speedup is $S_p = T_1/T_p$, $S_p \leq p$
- efficiency is $E_p = S_p/p$, $0 < E_p \leq 1$

Efficiency and scaling
Critical path analysis
Granularity
LU factorization analysis

But:
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product

- Is T_1 based on the same algorithm? The parallel code?
- Sometimes superlinear speedup.
- Is T_1 measurable? Can the problem be run on a single processor?

Parallelism and implicit operations: Wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

6 Speedup

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The programming model

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

Task scheduling

Integers

Floating point numbers

Floating point math

Examples

Efficiency and scaling

Critical path analysis

Granularity

T_1

More

Essential aspects of LU factorization

f_s

f_p

Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

f_s

f_p/p

P_0

Incomplete approaches to matrix factorization

Parallelism and implicit operations

P_1

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

T_p

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor

Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

8 Amdahl's law, analysis

The SIMD/MIMD/SIMD/MIMD/MIMD paradigm

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

- F_s sequential fraction, F_p parallelizable fraction
Floating point numbers
 - $F_s + F_p = 1$
Floating point math
Examples
More
 - $T_1 = (F_s + F_p)T_1$ Essential aspects of LU factorization
 $= F_s T_1 + F_p T_1$ Sparse matrices: storage and algorithms
 - Iteration methods, basic concepts and available methods
Amdahl's law: $T_p = F_s T_1 + F_p T_1 / p$
Collectives as building blocks, complexity
 - Scalability analysis of dense matrix-vector product
 $P \rightarrow \infty$: $T_p \downarrow T_1 F_s$ Sparse matrix-vector product
 - Latency hiding / communication minimizing
Parallel LU through nested dissection
Iterative approaches, matrix factorization
 - Speedup is limited by $S_P < 1/F_s$, efficiency is a decreasing function $E \sim 1/P$.

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

o you see problems with this?

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging: optimization and programming strategies.

9 Amdahl's law with communication overhead

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

How DDM, SIMD, SPU, etc. make parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Efficiency

More

- Communication independent of p : $T_p = T_c(F + F_p/P) + T_c$

- assume fully parallelizable: $F_p = 1$

Sparse matrices: storage and algorithms

- then $S_p = T_c$

Collectives as building blocks; complexity

- For reasonable speedup: $T_c \ll T_1/p$ or $p \ll T_1/T_c$:

number of processors limited by ratio of scalar execution time and communication overhead

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Efficiency and scaling

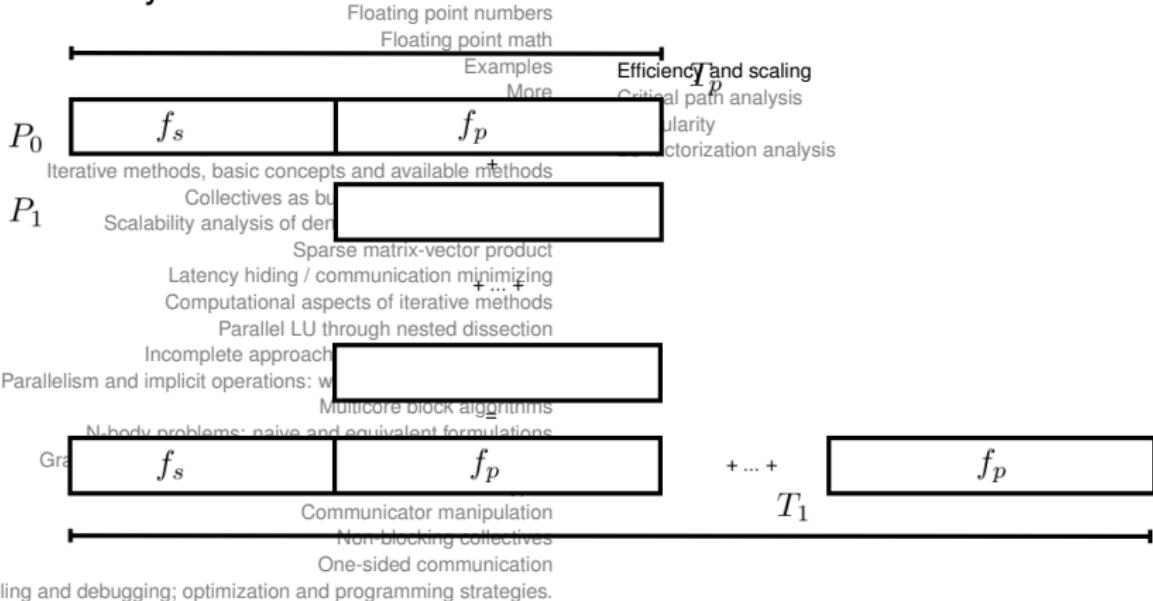
Critical path analysis

Granularity

LU factorization analysis

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models

Reconstruct the sequential execution from the parallel, then analyze efficiency.



Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

11 Gustafson's law

The SIMD/MIMD/SPMD/SIMT model, threads, tasks

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

- Let $T_p = F_s + F_p \equiv 1$ More
- then $T_1 = F_s + p \cdot F_p$ Essential aspects of LU factorization

- Speedup:
Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

$$S_p = \frac{T_1}{T_p} = \frac{F_s + p \cdot F_p}{F_s + F_p}$$

Sparse matrix-vector product

$$S_p = \frac{T_1}{T_p} = \frac{F_s + p \cdot F_p}{F_s + F_p}$$

Lattice hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism: multicore, manycore, GPU, distributed computation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Efficiency and scaling

Critical path analysis

Granularity

LU factorization analysis

$$S_p = \frac{T_1}{T_p} = \frac{F_s + p \cdot F_p}{F_s + F_p} = F_s + p \cdot F_p = p - (p-1) \cdot F_s.$$

slowly decreasing function of p

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

IEEE floating point

Examples

- Amdahl's law: **strong scaling**

same problem over increasing processors

Efficiency and scaling

Critical path analysis

Granularity

factorization analysis

- Often more realistic: **weak scaling**

Iterative methods, basic concepts and available methods

increase problem size with number of processors,

for instance keeping memory constant

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism in iterative methods: nested iteration

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

12 Scaling

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math

- Let M be the total memory needed for your problem.

More
Essential aspects of LU factorization
Iterative methods, basic concepts and available methods
 \Rightarrow memory per processor is M/P

- Let P be the number of processors
More
Scalability analysis of dense matrix-vector product
 \Rightarrow memory per processor is M/P
- What is $\lim_{P \rightarrow \infty} E_P$?
Sparse matrix-vector product
Latency hiding / communication minimizing
Incomplete approaches to matrix factorization
Parallel LU through nested dissection

Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Efficiency and scaling
Critical path analysis
Granularity
LU factorization analysis

13 Strong scaling

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math

- Let M be the memory per processor
 - More Examples
- Let P be the number of processors
 - Essential aspects of LU factorization
 - Iterative methods, basic concepts and available methods

⇒ total memory is $M \cdot P$

- Scalability analysis of dense matrix-vector product
- What is $\lim_{P \rightarrow \infty} E_P$?
 - Sparse matrix-vector product
 - Latency hiding / communication minimizing
 - Conjugate gradient, iterative methods
 - Parallel LU through nested dissection

Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

14 Weak scaling

Efficiency and scaling
Critical path analysis
Granularity
LU factorization analysis

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

Power question

Batched code

Theoretical concepts

- **Assumption:** simulated time S , running time T constant, now increase precision
 - Characterization of parallelism by memory model
 - Implementation of theoretical concepts
 - Programming models
- m memory per processor, and P the number of processors
 - Load balancing, locality, space filling curves
 - First we dig into bits
 - Integers
 - Floating point numbers
 - Floating point math
 - Examples
 - More

d the number of space dimensions of the problem, typically

2 or 3.

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Efficiency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Memory manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies

$$\Delta x = 1/M^{1/d}$$

total memory.

Efficiency and scaling

Critical path analysis

Granularity

LU factorization analysis

grid spacing.

hyperbolic case

parabolic case

With a simulated time S :

Memory manipulation

Non-blocking collectives

One-sided communication

$$k = S/\Delta t \quad \text{time steps.}$$

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Access patterns

Theoretical concepts

16 Simulation scaling con'td

- Assume time steps parallelizable

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

memory per processor goes down.

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

- Substituting $M = Pm$, we find ultimately

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

$$T = kM/P = \frac{S}{\Delta t} m.$$

Efficiency and scaling

Critical path analysis

Complexity

LU factorization analysis

$$m = C\Delta t,$$

$$m = C\Delta t = C \begin{cases} 1 / M^{1/d} & \text{hyperbolic case} \\ 1 / M^{2/d} & \text{parabolic case} \end{cases}$$

$$m = C \begin{cases} 1 / P^{1/(d+1)} & \text{hyperbolic} \\ 1 / P^{2/(d+2)} & \text{parabolic} \end{cases}$$

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Critical path analysis

more
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Critical path analysis
Granularity
LU factorization analysis

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

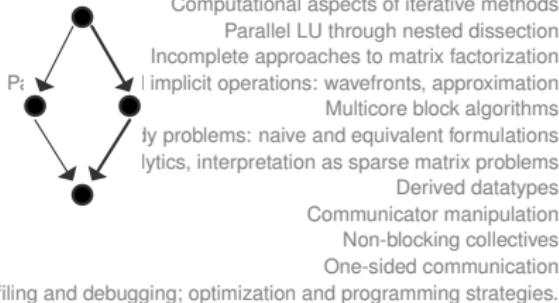
Programming models

Load balancing, locality, space-filling curves

Floating point numbers
First we dig into bits
Integers

Floating point math

- The sequential fraction contains a *critical path*: a sequence of operations that depend on each other.
- Example?
- T_{∞} = time with unlimited processors: length of critical path.

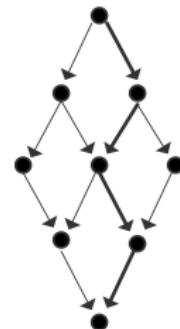


Efficiency and scaling

Critical path analysis

Granularity

Parallel matrix-vector multiplication



Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model of parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Exponents

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

LU factorization, direct methods

Incomplete approaches to matrix factorization

Parallelism and implicit operations, way forward: approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

18 Brent's theorem

Let m be the total number of tasks, p the number of processors, and t the length of a *critical path*. Then the computation can be done in

$$T \leq t + \frac{m}{p}$$

- Time equals the length of the critical path ...

- ... plus the remaining work as parallel as possible.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Granularity

more
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Critical path analysis
Granularity
LU factorization analysis

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model of parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Scalability analysis and algorithms

Iterative methods, basic concepts and available methods

LU factorization

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

19 Definition

Definition: granularity is the measure for how many operations can be performed between synchronizations

Efficiency and scaling

Critical path analysis

Granularity

LU factorization

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

20 Instruction level parallelism

The SIMD/UMD/SIMT/VM model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Efficiency and scaling

Critical path analysis

Granularity

LU factorization analysis

$$a \leftarrow b + c$$

$$d \leftarrow e * f$$

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexities

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Lateness hiding, communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

For the compiler / processor to worry about

- Structure of a modern processor
- Memory hierarchy: caches, register, TLB.
- Multicore issues
- Programming strategies for performance
 - The power question
 - Basic concepts
 - Theoretical concepts
- The SIMD/MIMD/SPMD/SIMT model of parallelism
- Characterization of parallelism by memory model
- Interconnects and topologies, theoretical concepts
- Programming models
 - Load balancing, locality, space-filling curves
 - First we dig into bits
 - Integers
 - Floating point numbers
- for** ($i=0$; $i < 1000000$; $i = i + 1$)
 - Examples
 - More
- Essential aspects of LU factorization
- Sparse matrices: storage and algorithms
- Iterative methods, basic concepts and available methods
- **Array processors, vector instructions, pipelining, GPUs**
 - Computing on arrays
 - Scalability analysis of dense matrix-vector product
- **Sometimes harder to discover**
 - Sparse matrix-vector product
 - Latency hiding / communication minimizing
 - Computational aspects of iterative methods
 - Implementation of high-level algorithm
- **Often used mixed with other forms of parallelism**
 - Incomplete approaches to matrix factorization
 - Parallelism and implicit operations: wavefronts, approximation
 - Multicore block algorithms
- N-body problems: naive and equivalent formulations
- Graph analytics, interpretation as sparse matrix problems
 - Derived datatypes
 - Communicator manipulation
 - Non-blocking collectives
 - One-sided communication
- Profiling and debugging; optimization and programming strategies.



Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SIMT/SIMT-like paradigm
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More

Efficiency and scaling
Critical path analysis
Granularity
LU factorization analysis

Example: Mandelbrot set
Matrix factorizations: LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Parameter sweep,
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MMX/SSE: how does it work and why?
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples

Mix of data parallel and task parallel

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
my_lower_bound = // some processor-dependent number
my_upper_bound = // some processor-dependent number
for (*i*=*my_lower_bound*; *i*<*my_upper_bound*; *i*++)
 // the loop body goes here
 Scalability analysis of dense matrix-vector product
 Sparse matrix-vector product
 Parallel LU through nested dissection
 Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
 Multicore block algorithms
 N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
 Derived datatypes
 Communicator manipulation
 Non-blocking collectives
 One-sided communication
Profiling and debugging; optimization and programming strategies.

Efficiency and scaling
Critical path analysis
Granularity
Task-based parallelism

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers

LU factorization analysis

more
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Critical path analysis
Granularity
LU factorization analysis

Structure of a modern processor
 Memory hierarchy: caches, register, TLB.
 Multicore issues
 Programming strategies for performance
 The power question
 Basic concepts
 Theoretical concepts
 The SIMD/MIMD/SPMD/SIMT model for parallelism
 Characterization of parallelism by memory model
 Interconnects and topologies, theoretical concepts
 Programming models
 Load balancing, locality, space-filling curves
 First we dig into bits
 Integers
 Floating point numbers
 Floating point math
 Examples
for $k = 1, n - 1$:
for $i = k + 1$ to n :
 More
 Essential aspects of LU factorization
 $a_{ik} \leftarrow a_{ik} / a_{kk}$
 Sparse matrices: storage and algorithms
 Iterative methods, basic concepts and available methods
for $i = k + 1$ to n :
 Collective operations: building blocks; complexity
 Scalability analysis of dense matrix-vector product
for $j = k + 1$ to n :
 Sparse matrix-vector product
 Latency hiding / communication minimizing
 $a_{ij} \leftarrow a_{ij} - a_{ik} * a_{kj}$
 Computational approaches: iterative methods
 Parallel LU through nested dissection
 Incomplete approaches to matrix factorization
 Parallelism and implicit operations: wavefronts, approximation
 Multicore block algorithms
 N-body problems: naive and equivalent formulations
 Graph analytics, interpretation as sparse matrix problems
 Derived datatypes
 Communicator manipulation
 Non-blocking collectives
 One-sided communication
 Profiling and debugging; optimization and programming strategies.

25 Algorithm

Efficiency and scaling
 Critical path analysis
 Granularity
LU factorization analysis

Can the k loop be done in parallel? The i, j loops?

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
26 Dependent operations
The SIMD/MIMD/BMWD/SIMD/MIMD/BMWD
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More
Essential aspects of LU factorization
Sparse matrices: storage, algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
 $a_{33} \leftarrow a_{33} - a_{32} * a_{22}^{-1} a_{23}$
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SSP paradigm: parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Example

More

Critical path analysis

Summary

LU factorization analysis

Follow this argument through: Argue that there is a non-trivial *critical path* in the sense of section ???. What is its length?

In the analysis of the critical path section, what does this critical path imply for the minimum parallel execution time and bounds on speedup?

Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/IMM model (parallelism)

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

$a_{ij} \leftarrow a_{ij} - a_{ik} * a_{kj}$

Collective as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Conjugate gradient, biconjugate gradient

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

27 Subblock update

for $i = k + 1$ to n :

 Essential aspects of LU factorization

 for $j = k + 1$ to n :

 Sparse matrices: storage and algorithms

 Iterative methods, basic concepts and available methods

$a_{ij} \leftarrow a_{ij} - a_{ik} * a_{kj}$

 Collective as building blocks; complexity

 Scalability analysis of dense matrix-vector product

 Sparse matrix-vector product

 Latency hiding / communication minimizing

 Computational aspects of iterative methods

 Conjugate gradient, biconjugate gradient

 Incomplete approaches to matrix factorization

 Parallelism and implicit operations: wavefronts, approximation

 Multicore block algorithms

 N-body problems: naive and equivalent formulations

 Graph analytics, interpretation as sparse matrix problems

 Derived datatypes

 Communicator manipulation

 Non-blocking collectives

 One-sided communication

Efficiency and scaling

Critical path analysis

Granularity

LU factorization analysis

How many processors can you use maximally in step k ?

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/MD SIMD model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Arithmetic exceptions

More

Locality, cache, communication

Sparse matrices: storage and algorithms

Iterative methods: basic concepts and available methods

Collectives as building blocks: complexity

Scalability analysis: matrix-vector, vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computing the LU factorization

Parallel LU through nested dissection

Communication avoiding LU factorization

Parallelism and implicit operations: wavefronts, approximation

Implicit operations: LU factorization

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Exercise 2: Parallel execution

Continue this reasoning. With $p = n^2$ processing elements each of the (i, j) updates in the subblock can be done simultaneously. To be precise, how long does an arbitrary k iteration take? Summing over all k , what is the resulting T_p, S_p, E_p ? How does this relate to the bounds you derived above?

Also, with $p = n$ processing elements you could let each row or column of the subblock update be done in parallel. What is now the time for the k th outer iteration? What is the resulting T_p, S_p, E_p ?,

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SIMDT paradigm

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Iterative methods, direct and indirect

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

$$T = \frac{1}{3} N^3/f,$$

$$M = N^2.$$

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Non-local problems: LU factorization

where f is processor frequency

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

28 Application scaling

Single processor.

Relating time and memory to problem size

Efficiency and scaling

Critical path analysis

Granularity

LU factorization analysis

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

Implementation of SIMD/SIMD/SIMT parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks, complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Suppose you buy a processor twice as fast, and you want to do a benchmark run that again takes time T . How much memory do you need?

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Efficiency and scaling

Critical path analysis

Granularity

LU factorization analysis

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/MPMD model of parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples

Keep frequency constant, but vary number of processors p :

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks: complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
 $T = \frac{1}{3}N^3/p$, $M = N^2$.

Each processor now stores $M_p = N^2/p$ elements.

Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theory and concepts
Programming models

Exercise 4: Memory scaling, case 2: More processors

Suppose you have a cluster with p processors, each with M_p memory, can run a Gaussian elimination of an $N \times N$ matrix in time T :

$$T = \frac{1}{3} N^3 / p.$$

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks: complexity

Scalability analysis of dense matrix-vector product
Latency hiding + communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Algorithmic building blocks, algorithms

Hint: for the extended cluster:

N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems

$$T' = \frac{1}{3} N'^3 / p'.$$

$$M'_p = N'^2 / p'.$$

Efficiency and scaling
Critical path analysis
Granularity
LU factorization analysis

The question becomes to compute M'_p under the given conditions.

Processor Architecture

Table of Contents

- Structure of a modern processor

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

- Memory hierarchy: caches, register, TLB.

Interconnects and topologies; theoretical concepts

Programming models

Load balancing, locality, space-filling curves

- Multicore issues

First we dig into bits

Integers

- Programming strategies for performance

Floating point numbers

Floating point math

Examples

More

- The power question

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Parallelism 85

Iterative methods; basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

- Basic concepts

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

- Theoretical concepts

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

- The SIMD/MIMD/SPMD/SIMT model for parallelism

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Programming models

- Structure of a modern processor
- Memory hierarchy: caches, register, TLB.
- Multicore issues
- Programming strategies for performance
 - The power question
 - Basic concepts
 - Theoretical concepts
- The SIMD/MIMD/SPMD/SIMT model for parallelism**
 - Characterization of parallelism by memory model
 - Interconnects and topologies, theoretical concepts
 - Programming models
 - Load balancing, locality, space-filling curves
 - First we dig into bits

The SIMD/MIMD/SPMD/SIMT model for parallelism

more

- Essential aspects of LU factorization
- Sparse matrices: storage and algorithms
- Iterative methods, basic concepts and available methods
 - Collectives as building blocks; complexity
 - Scalability analysis of dense matrix-vector product
 - Sparse matrix-vector product
 - Latency hiding / communication minimizing
 - Computational aspects of iterative methods
 - Parallel LU through nested dissection
 - Incomplete approaches to matrix factorization
- Parallelism and implicit operations: wavefronts, approximation
 - Multicore block algorithms
 - N-body problems: naive and equivalent formulations
 - Graph analytics, interpretation as sparse matrix problems
 - Derived datatypes
 - Communicator manipulation
 - Non-blocking collectives
 - One-sided communication
- Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts

30 Flynn Taxonomy

The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

Consider instruction stream and data stream:

First we dig into its
Integers

Floating point numbers

Floating point math

More

- SISD: single instruction single data
used to be single processor, now single core

Iterative methods, basic concepts and available methods

Sparse matrices: storage and algorithms

- MISD: multiple instruction single data
redundant computing for fault tolerance?

Collectives as building blocks, complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Compiler and application level methods

Parallel LU through nested dissection

Optimal assignments to hypercube nodes

Parallelism and implicit operations: wavefronts, approximation

- SIMD: single instruction multiple data
data parallelism, pipelining, array processing, vector instructions

N-body problems: naive and equivalent formulations

Graph partitioning, load balancing, sparse problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

- Relies on streams of identical operations

Essential concepts of HPC applications
Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

- See pipelining

Scalability analysis of sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

31 SIMD

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

32 SIMD: array processors

The SIMD/MIMD/SIMD/SIMD/MIMD hybrid paradigm

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices, storage and algorithms

Iterative methods, basic concepts and available methods

GoodYear Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Processor matrix-vector product

Major advantage: simplification of processor

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

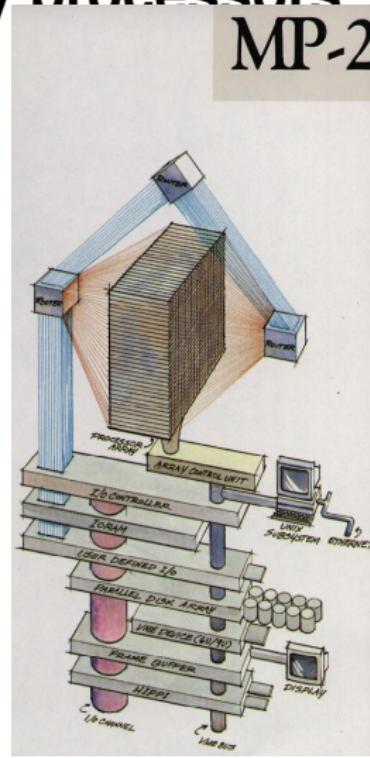
Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.



Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD (SIMD/SIMD-DIM) model of parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

- Register width multiple of 8 bytes:

Load balancing, locality, space-filling curves

First we dig into bits

- simultaneous processing of more than one operand pair

Floating point numbers

Floating point math

Examples

- SSE: 2 operands,

More

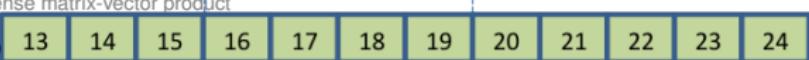
Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

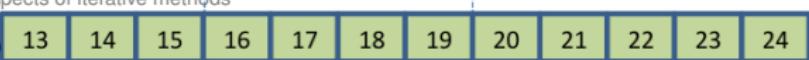
Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

array a: 

Latency hiding /

Computational aspects of iterative methods

array b: 

Parallel LU

Incomplete approach

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

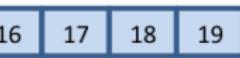
Derived datatypes

$b[i] = a[i] + b[i]$

Communication manipulation

Non-blocking collectives

One-sided communication



Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

34 Controlling vector instructions

The SIMD/MMPS SIMD SIMD model of parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

```
void func(float restrict c, float *restrict a,  
         float *restrict b, int n)
```

{

 Essential aspects of LU factorization

```
#pragma vector always
```

 Sparse matrices storage and algorithms

 Iterative methods, basic concepts and available methods

 for (int i=0; i<n; i++)

 Collectives as building blocks; complexity

 Sparcity analysis, dense matrix-vector product

}

 Sparse matrix-vector product

 Latency hiding / communication minimizing

 Computational aspects of iterative methods

 Parallel LU through nested dissection

 Incomplete approaches to matrix factorization

 Parallelism and implicit operations: wavefronts, approximation

 Multicore block algorithms

 N-body problems: naive and equivalent formulations

 Graph analytics, interpretation as sparse matrix problems

 Derived datatypes

 Communicator manipulation

 Non-blocking collectives

 One-sided communication

 Profiling and debugging; optimization and programming strategies.

This needs aligned data (posix_memalign)

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

35 New branches in the taxonomy

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

- **SPMD: single program multiple data**
 - Assume a problem is factorizable
 - Sparse matrices: storage and algorithms
 - Iterative methods: basic concepts and available methods

- **SIMT: single instruction multiple threads**
 - Collectives as building blocks; complexity
 - Locality: shared data structures for SIMD
 - Sparse matrix-vector product
 - Memory access: communication minimizing
 - Computational aspects of iterative methods

- Parallel LU through nested dissection
- Incomplete approaches to matrix factorization
- Parallelism and implicit operations: wavefronts, approximation
- Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

The pointer concepts

36 MIMD becomes SPMD

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

Floating point

Integers

Floating point numbers

Floating point math

- **MIMD: independent processors, independent instruction streams, independent data**
 - Examples
 - More
- In practice very little true independence: usually the same executable
 - Essential aspects of LU factorization
 - Sparse matrices: storage and algorithms
 - Iterative methods, basic concepts and available methods
 - Collectives as building blocks, complexity
 - Scalability analysis of dense matrix-vector product
 - Matrix-vector product
 - Latency hiding / communication minimizing
 - Communication patterns: bus, memory
 - Parallel LU through nested dissection
 - Incomplete approaches to matrix factorization

Single Program Multiple Data

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Parallel LU through nested dissection

- **Exceptional example: climate codes**
- **Old-style SPMD: cluster of single-processor nodes**
- **New-style: cluster of multicore nodes, ignore shared caches / memory**
 - Communicator manipulation
 - Non-blocking collectives
 - One-sided communication
- **(We'll get to hybrid computing in a minute)**

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MMD/SIMD/MMD model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Lockstep in thread block,

LU factorization and sparse LU factorization

Sparse matrices: storage and algorithms

LU factorization: parallel computation, algorithmic methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

(more about GPU threads later)

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

37 GPUs and data parallelism

single instruction model between streaming processors

- **Structure of a modern processor**

- **Memory hierarchy: caches, register, TLB.**

- **Multicore issues**

- **Programming strategies for performance**

- **The power question**

Parallelism 85

- **Basic concepts**

- **Theoretical concepts**

- **The SIMD/MIMD/SPMD/SIMT model for parallelism**

- **Characterization of parallelism by memory model**

- **Interconnects and topologies; theoretical concepts**

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Characterization of parallelism by memory model

more

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

38 Major types of memory organization, classic

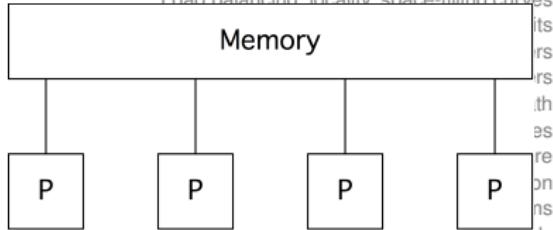
Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

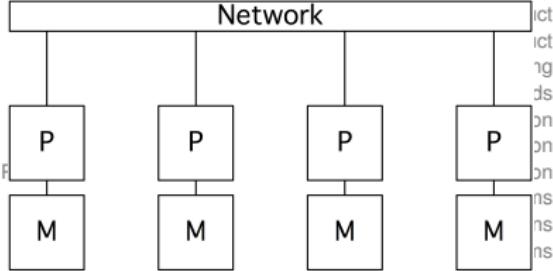
Memory



Iterative methods, basic concepts and available methods

Collectives as building blocks: complexity

Network



Derived datatypes

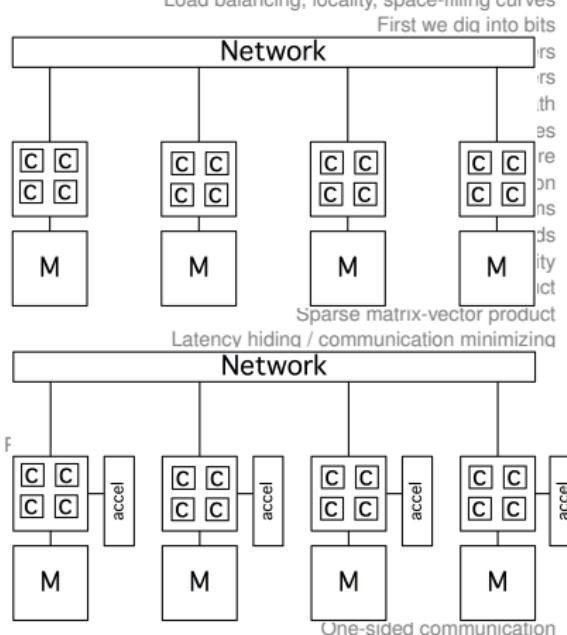
Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

39 Major types of memory organization, contemporary



Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

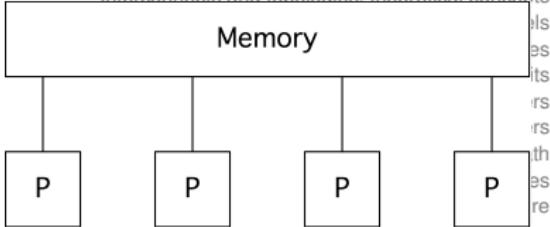
Basic concepts

Theoretical concepts

The SIMD / MIMD / SIMD-SIMD hybrid paradigm

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts



Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

- **The ideal case of shared memory:**
Scalability analysis of dense matrix-vector product
Scaling of LU factorization
Latency hiding / communication minimizing

- **This hasn't existed in a while**
Computational aspects of iterative methods
Parallel LU through nested dissection

(Tim Mattson claims Cray-2)
Parallelism and implicit operations: Wavefronts, approximation

- **Danger signs: shared memory programming pretends that memory access is symmetric**
Multicore block algorithms
Memory alignment and cache coherency issues
Graph analytics, interpretation as sparse matrix problems
Data locality types
Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

40 Symmetric multi-processing

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SPMD more parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

- Bus: all processors on the same wires to memory

Sparse matrices: storage and algorithms

- Not very scalable, requires slow processors or cache memory

Collectives as building blocks, complexity

- Cache coherence easy by 'snooping'

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

4.1 SMP, bus design

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues
Programming strategies for performance

The power question
Basic concepts

Theoretical concepts

42 Non-uniform Memory Access

The SIMD/MIMD/MPSIM model of parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Memory is equally programmable, but not equally accessible

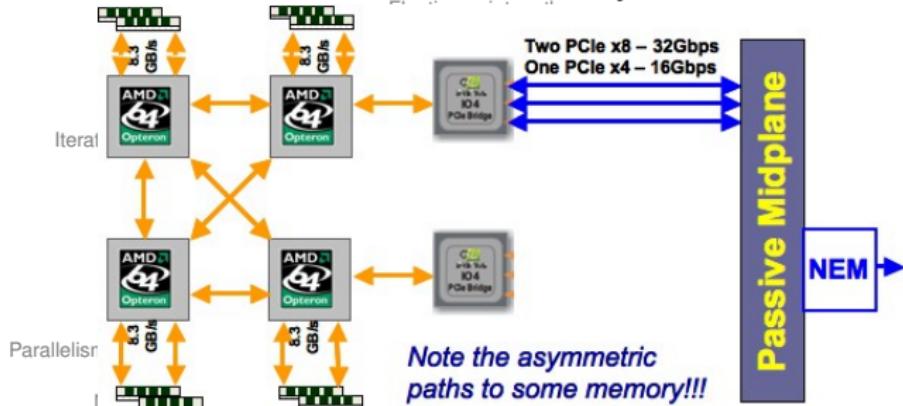
Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

- Different caches, different affinity



Graph analytics, interpretation as sparse matrix problems

- Distributed shared memory, network latency

Derived datatypes

Communicator manipulation

Non-blocking collective

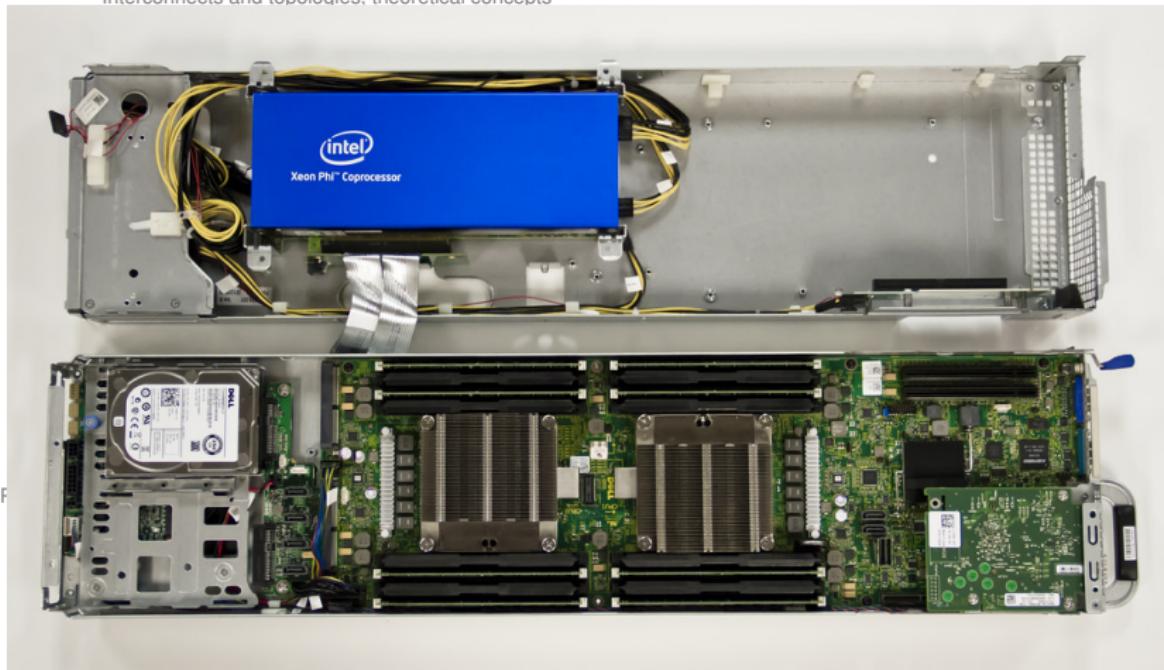
One-sided communication

ScaleMP and other products watch me not believe it

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
The local concepts
The SIMD/MIMD/SPMD/MPMD model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts

43 Picture of NUMA



One-sided communication

Profiling and debugging; optimization and programming strategies.

Table of Contents

Processor Architecture

- Structure of a modern processor

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

- Memory hierarchy: caches, register, TLB.

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

- Multicore issues

First we dig into bits

Integers

- Programming strategies for performance

Floating point numbers

Floating point math

Examples

More

- The power question

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Parallelism

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

- Basic concepts

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

- Theoretical concepts

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

- The SIMD/MIMD/SPMD/SIMT model for parallelism

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

The MPI-2 API

- Interconnects and topologies, theoretical concepts

Profiling and debugging; optimization and programming strategies.

- Structure of a modern processor
- Memory hierarchy: caches, register, TLB.
- Multicore issues
- Programming strategies for performance
 - The power question
 - Basic concepts
 - Theoretical concepts
- The SIMD/MIMD/SPMD/SIMT model for parallelism
 - Characterization of parallelism by memory model
- Interconnects and topologies, theoretical concepts**
 - Programming models
 - Load balancing, locality, space-filling curves

Interconnects and topologies, theoretical concepts

- Sparse matrices: storage and algorithms
- Iterative methods, basic concepts and available methods
 - Collectives as building blocks; complexity
 - Scalability analysis of dense matrix-vector product
 - Sparse matrix-vector product
 - Latency hiding / communication minimizing
 - Computational aspects of iterative methods
 - Parallel LU through nested dissection
 - Incomplete approaches to matrix factorization
- Parallelism and implicit operations: wavefronts, approximation
 - Multicore block algorithms
 - N-body problems: naive and equivalent formulations
 - Graph analytics, interpretation as sparse matrix problems
 - Derived datatypes
 - Communicator manipulation
 - Non-blocking collectives
 - One-sided communication
- Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

44 Topology concepts

The SIMD/MIMD/SIMT model and parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

- **Hardware characteristics**

Fuse architecture, tile optimization

Sparse matrices: storage and algorithms

- **Software requirement**

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector multiplication

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts

The SIMD/MIMD/SPMD/SIMT paradigm follows here

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More

Essential aspects of LU factorization

- **Degree:** number of connections from one processor to others

Sparse matrix-vector product, complexity

Iterative methods, basic concepts and available methods

- **Diameter:** maximum minimum distance (measured in hops)

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

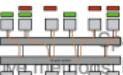
45 Graph theory

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model of parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts

Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math

Examples

- Bandwidth per wire is nice, adding over all wires is nice, but...

 Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative refinement: basic concepts and available methods

Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product

- Bisection width: minimum number of wires through a cut
- Bisection bandwidth: bandwidth through a bisection

Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms

N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems

Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

46 Bandwidth

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
The set of concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts

Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More

Essential aspects of LU factorization
Sparse matrices: storage and algorithms

Already discussed; simple design, does not scale very far

Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

47 Design 1: bus

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/MOD/IMOD model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

- **Degree 2, diameter P , bisection width 1**

Associated properties of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

- **Scales nicely!**

Scalability analysis of matrix-matrix product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

48 Design 2: linear arrays

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD (SIMT) Model: more for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Parallel LU methods, block LU, incomplete LU, iterative methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Flip last bit, flip one before, . . .

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

49 Design 3: 2/3-D arrays

The SIMD/MIMD/P2P/SIMD/MIMD hybrid paradigm

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

- Degree $2d$, diameter $P^{1/d}$

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

- Iterative methods: variants and their relation

Collectives as building blocks; complexity

- Scalability analysis of dense matrix-vector products

Sparse matrix-vector product

LU, Cholesky / incomplete LU, scaling

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

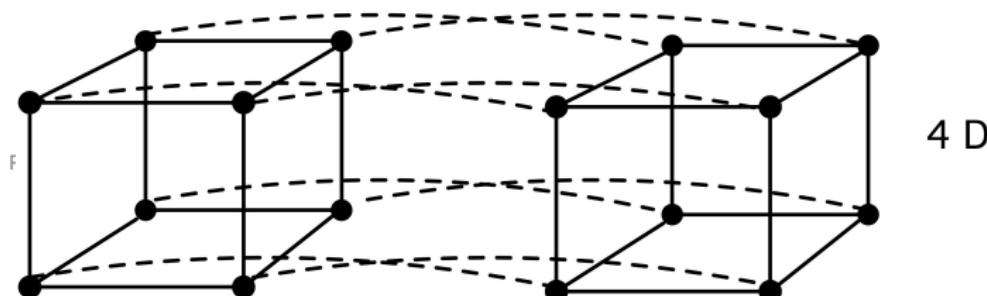
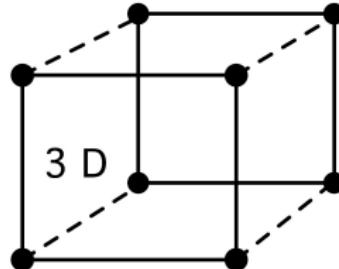
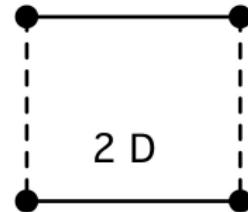
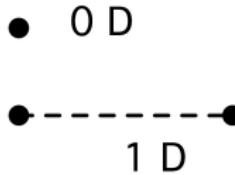
50 Design 3: Hypercubes

The SIMD/MIMD-SIMD/SIMD-MIMD model of parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models



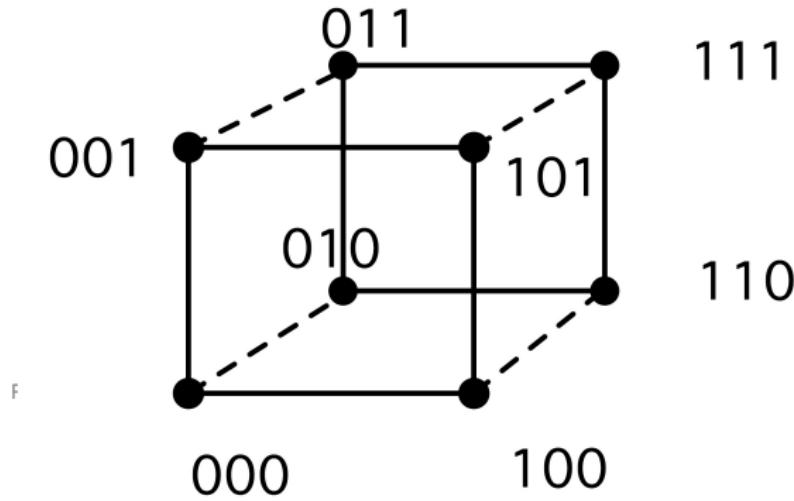
Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/PIMD/SIMD model of parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Naive numbering:



Profiling and debugging; optimization and programming strategies.

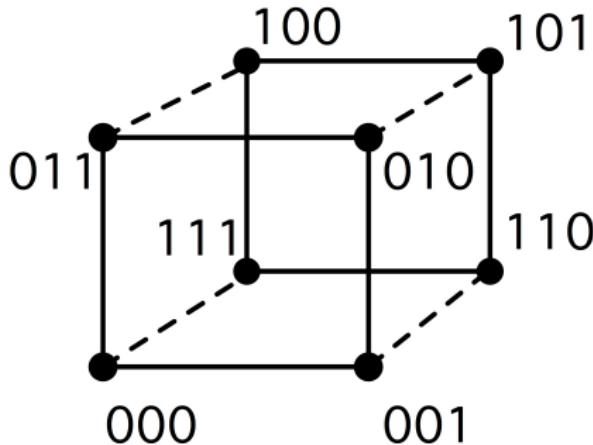
Communicator manipulation
Non-blocking collectives
One-sided communication

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves

52 Gray codes

Theoretical concepts
First we dig into bits
Integer

Embedding linear numbering in hypercube:



Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SIMT/SPMD memory model

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

1D Gray code :

Integers

0 1

Floating point numbers

Floating point math

2D Gray code :

1D code and reflection:

More

0 1 : 1 0

Essential aspects of LU factorization

Sparse matrices: append 0 and 1 bit:

0 0 : 1 1

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis: the matrix-free approach

2D code and reflection:

0 1 1 0 : 0 1 1 0

Sparse matrix-vector product

Coefficient hiding / communication minimizing

Computational aspects of iterative methods

0 0 1 1 : 1 1 0 0

3D Gray code :

Parallel LU through nested dissection

Incomplete approaches: serially and parallel

append 0 and 1 bit:

0 0 0 0 : 1 1 1 1

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

53 Binary reflected Gray code

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMV/AMM model of computation
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts

Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples

- Solution to all-to-all connection
 - Essential aspects of LU factorization
- (Real all-to-all too expensive)
 - Sparse matrices: storage and algorithms
 - Iterative methods; basic concepts and available methods
 - Collectives as building blocks; complexity
 - Sparse matrix-vector product
 - Sparse matrix-vector product
- Typically layered
 - Computational aspects of iterative methods
 - Parallel LU through nested dissection
 - Incomplete approaches to matrix factorization
- Switching elements: easy to extend
 - Parallelism and implicit operations: wavefronts, approximation
 - Multicore block algorithms
 - N-body problems: naive and equivalent formulations
 - Graph analytics, interpretation as sparse matrix problems
 - Derived datatypes
 - Communicator manipulation
 - Non-blocking collectives
 - One-sided communication

Profiling and debugging; optimization and programming strategies.

54 Switching networks

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

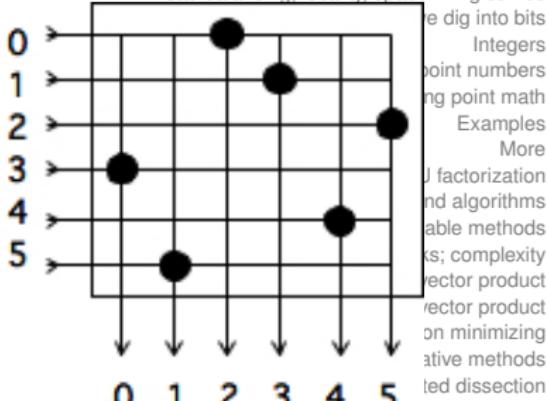
The SIMD/MIMD/SPMD/SIMT model; Parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves



We dig into bits

Integers

point numbers

point math

Examples

More

LU factorization

and algorithms

stable methods

complexity

vector product

vector product

on minimizing

active methods

ted dissection

x factorization

Parallelism and implicit operations: wavefronts, approximation
Advantage: non-blocking

N-body problems: naive and equivalent formulations

Sparse matrix problems: sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

56 Butterfly exchange

The SIMD/MIMD/SPMD model. The butterfly exchange

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits
integers

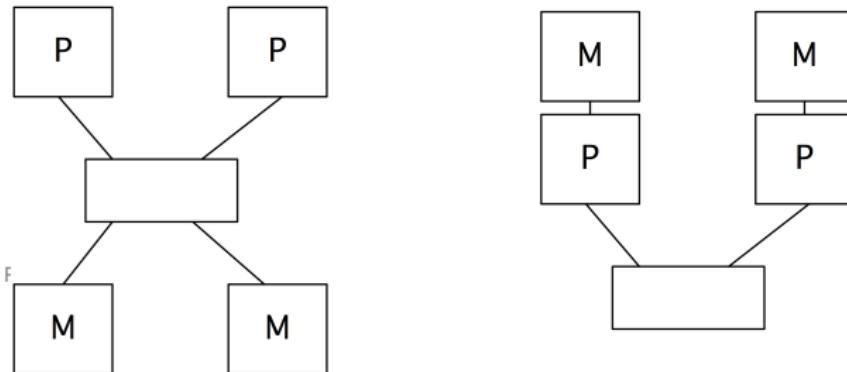
Floating point numbers

Floating point math

Examples

..

Process to segmented pool of memory, or between processors with private memory:



Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/PMD/SIMD/MIMD/PMD parallelism

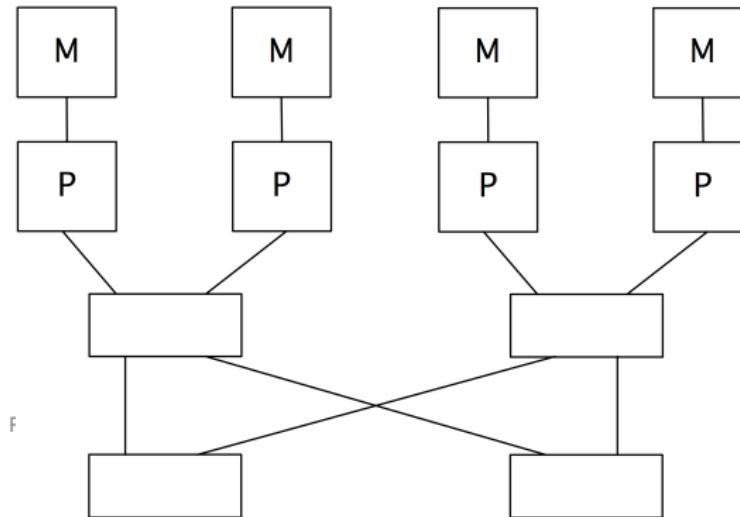
Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

57 Building up butterflies



Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

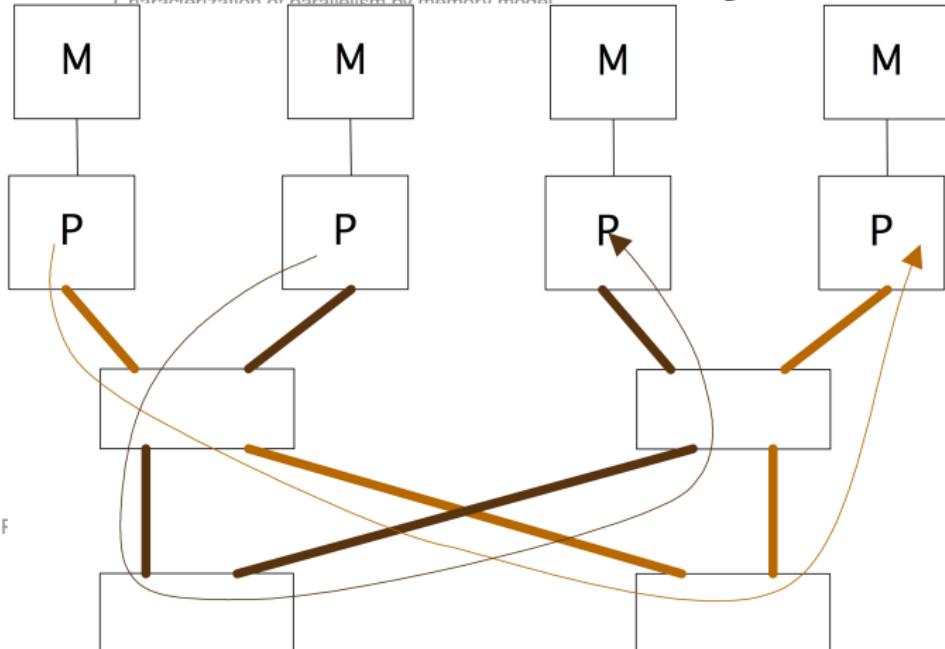
Basic concepts

Theoretical concepts

58 Uniform memory access

The SIMD/MIMD/SPMD/IMC model of parallelism

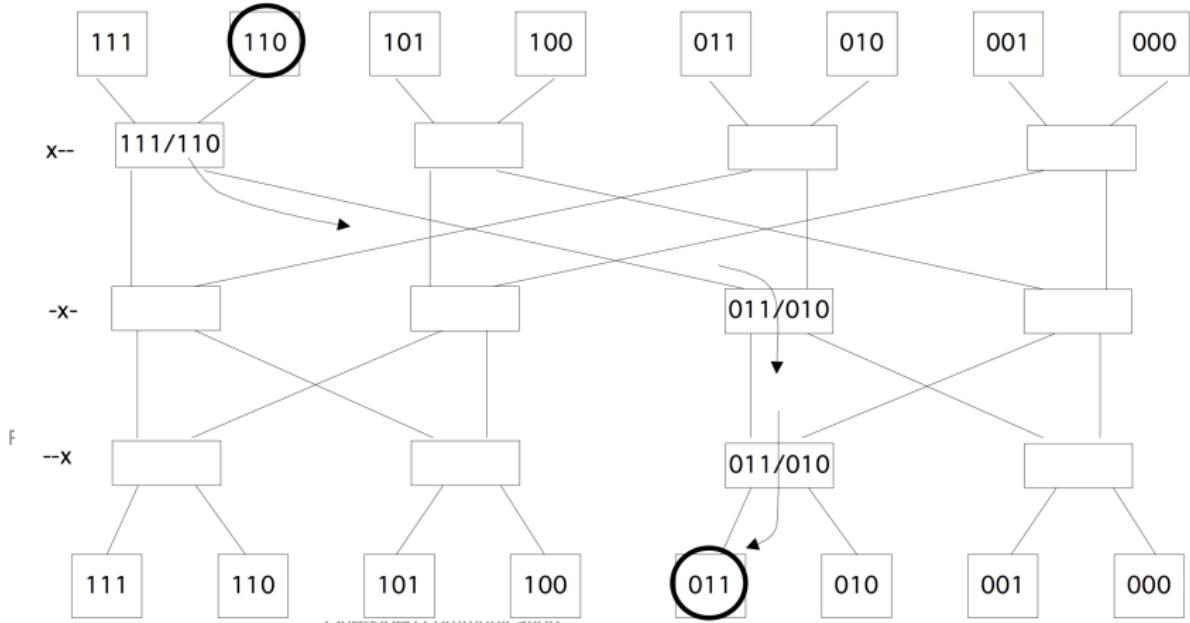
Characterization of parallelism by memory model



Contention possible
Profiling and debugging optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD paradigm of parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts

59 Route calculation



Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

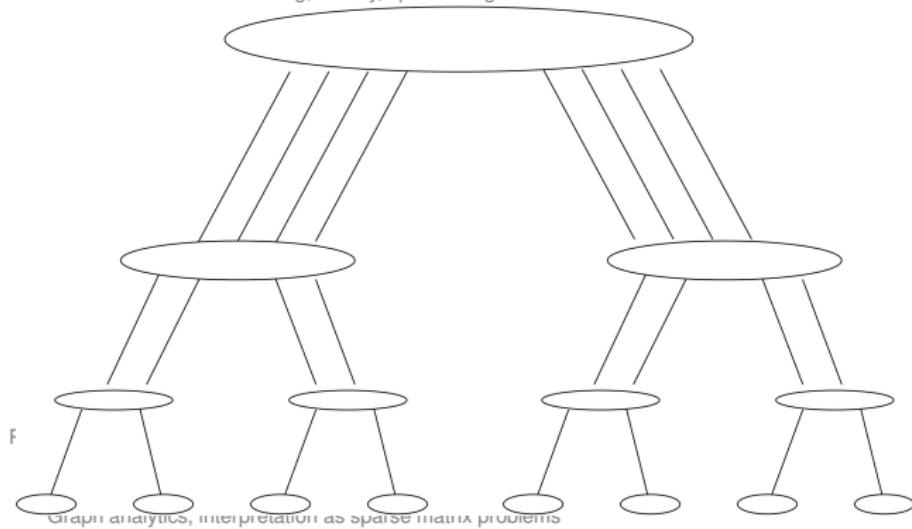
The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves



Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.



Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

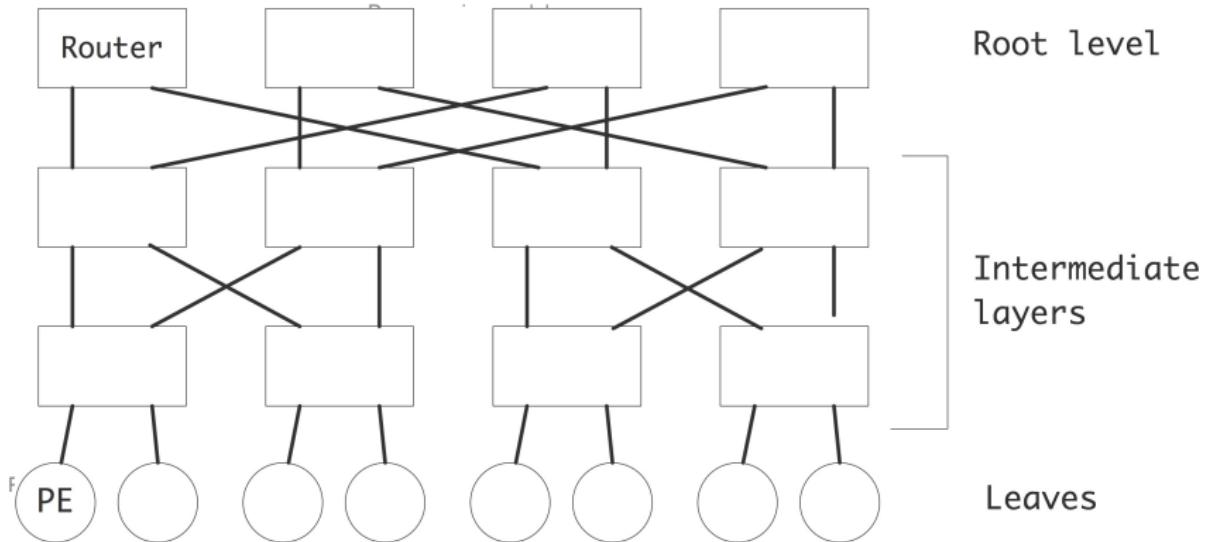
Theoretical concepts

The SIMD/MDP/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

61 Fat trees from switching elements



Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

(Clos network)

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples



Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

62 Fat tree clusters



Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD model / SIMD model of parallelism

Characterization of parallelism by memory model

Interactions and influences theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Floating point numbers

IEEE standard for floating point

Examples

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic ideas and available methods

Collectives as building blocks for parallelism

Scalability analysis of dense matrix-vector product

Memory hierarchy and memory access patterns for product

Memory hierarchy and memory access patterns for LU

Computational aspects of iterative methods

Parallel LU through nested dissection

Multicore block algorithms

N-body problems: naive and equivalent formulations

10-port switch



Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMD+MIMD model
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves.



Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD model (for parallelism)

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

- **Core level: private cache, shared cache**

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

- **Node level: numa**

Collectives as building blocks

Scalability analysis of LU factorization

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

64 Levels of locality

<p>Structure of a modern processor</p> <p>Memory hierarchy: caches, register, TLB</p> <p>The SIMD/MIMD/SPMD/SIMT model for parallelism</p> <p>Characterization of parallelism by memory model</p> <p>Interconnects and topologies; theoretical concepts</p> <p>Programming models</p> <p>Load balancing, locality, space-filling curves</p> <p>Multicore issues</p> <p>Floating point numbers</p> <p>Floating point math</p> <p>Examples</p> <p>More</p> <p>Essential aspects of LU factorization</p> <p>Sparse matrices: storage and algorithms</p> <p>Iterative methods: basic concepts and available methods</p> <p>Collectives as building blocks; complexity</p> <p>Scalability analysis of dense matrix-vector product</p> <p>Sparse matrix-vector product</p> <p>Latency hiding / communication minimizing</p> <p>Computational aspects of iterative methods</p> <p>Parallel LU through nested dissection</p> <p>Incomplete approaches to matrix factorization</p> <p>Parallelism and implicit operations: wavefronts, approximation</p> <p>Multicore block algorithms</p> <p>N-body problems: naive and equivalent formulations</p> <p>Graph analytics, interpretation as sparse matrix problems</p> <p>Derived datatypes</p> <p>Communicator manipulation</p> <p>Non-blocking collectives</p> <p>One-sided communication</p> <p>Profiling and debugging; optimization and programming strategies.</p>	<h1>Table of Contents</h1> <h2>Processor Architecture</h2> <ul style="list-style-type: none"> Structure of a modern processor Memory hierarchy: caches, register, TLB Multicore issues Programming strategies for performance The power question Basic concepts Theoretical concepts Interconnects and topologies; theoretical concepts Programming models <h2>Parallelism</h2> <ul style="list-style-type: none"> Basic concepts Theoretical concepts The SIMD/MIMD/SPMD/SIMT model for parallelism Characterization of parallelism by memory model Interconnects and topologies; theoretical concepts Programming models
--	---

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Programming models

more

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Hybrid/heterogeneous parallelism
Design patterns
What's left

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The parallel execution model

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

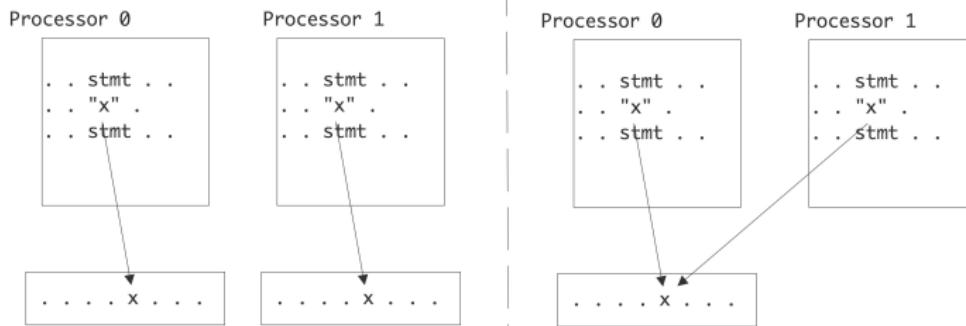
Characterization of parallelism by memory access

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

Different memory models:



Parallel I/O through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

- **Shared memory: synchronization problems such as critical sections**
 - Communicator manipulation
 - Non-blocking collectives
 - One-sided communication

- Profiling and debugging tools for parallel programming
- **Distributed memory: data motion**

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers

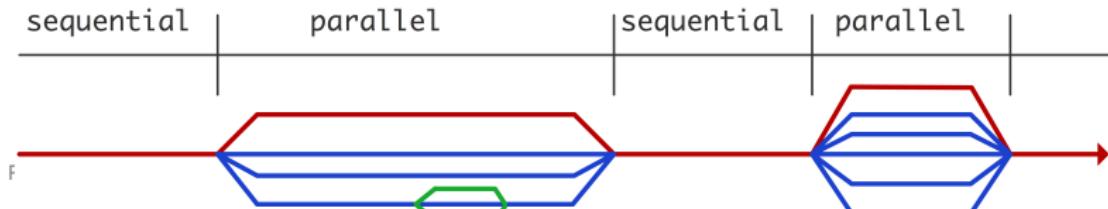
Thread parallelism

More
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Hybrid/heterogeneous parallelism
Design patterns
What's left

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
The basic concepts
The SIMD/MIMD/SPMD/SIMT model of parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models

- **Process:** code, heap, stack
 - Load balancing, locality, space-filling curves
 - Memory management
 - Integers
 - Floating point numbers
 - Floating point math
 - Examples
 - More
- **Thread:** same code but private program counter, stack, local variables
 - Thread parallelism
 - Distributed memory parallelism
 - Hybrid/heterogeneous parallelism
 - Debugging
 - What's left
- **dynamically (even recursively) created: fork-join**
 - Essential aspects of LU factorization
 - Sparse matrices, storage and algorithms
 - Iterative methods, basic concepts and available methods
 - Collectives as building blocks: complexity



Graph analytics, interpretation as sparse matrix problems

Derived datatypes
Data manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theorems/concepts
The SIMD/MIMD/SPMD/SIMD model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Precision
More

- Private data (stack, local variables) is called ‘thread context’
Thread parallelism
Distributed memory parallelism
Hybrid/heterogeneous parallelism
Timing synchronization
What's left
- Context switch: switch from one thread execution to another
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
- context switches are expensive; alternative hyperthreading
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
- Intel Xeon Phi: hardware support for 4 threads per core
Computational aspects of iterative methods
Modeling the memory hierarchy
Incomplete approaches to matrix factorization
- GPUs: fast context switching between many threads
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

67 Thread context

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/MOD/SIMD/MIMD/MOD parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Pthreads

```
pthread_t threads[NTHREADS];  
printf("forking\n");  
for (i=0; i<NTHREADS; i++)  
    if (pthread_create(&threads[i], NULL, &adder, NULL) !=0)  
        return i;  
printf("joining\n");  
for (i=0; i<NTHREADS; i++)  
    if (pthread_join(threads[i], NULL) !=0)  
        return NTHREADS+i;
```

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Thread parallelism

Distributed memory parallelism

Hybrid/heterogeneous parallelism

Design patterns

What's left

Parallel LU through nested dissection

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Iterative methods, basic concepts and available methods

Computational aspects of LU factorization

Parallel LU through nested dissection

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Implicit operations: wavefronts, approximation

Multidimensional algorithms

Parallelism and implicit operations: wavefronts, approximation

Multi-level block algorithms

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts

Programming models

Init: $I=0$ Load balancing, locality, space-filling curves
First we dig into bits

process 1: $I=I+2$ Integers

process 2: $I=I+3$ Floating point numbers
Floating point math

Examples
More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms
Iterative methods: concepts and available methods

Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing
Computational aspects of iterative methods

read $I=0$ **read** $I=0$ **read** $I=0$
set $I=2$ **set** $I=3$ **set** $I=2$

Parallel LU through nested dissection
Incomplete approaches to matrix factorization

write $I=2$ Parallel and implicit operations: wavefronts, approximation

Multicore block algorithms
N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems
Derived datatypes

Communicator manipulation

Non-blocking collectives

I = 3 **One-sided communication** $I=2$

Profiling and debugging: optimization and programming strategies

Thread parallelism

Distributed memory parallelism
Hybrid/heterogeneous parallelism

Design patterns
What's left

scenario 3.

read $I=0$ **read** $I=0$

set $I=3$ **set** $I=2$

write $I=3$ **write** $I=2$

read $I=2$

set $I=5$

write $I=5$

I = 5

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

70 Dealing with atomic operations

The SIMD/MMX/FPU model parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Physical vs logical LU factorizations

Semaphores, locks, mutexes, critical sections, transactional memory

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Parallel matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Thread parallelism

Distributed memory parallelism

Hybrid/heterogeneous parallelism

What's left

Software / hardware

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

• **Directive-based** storage and algorithms

Iterative methods, basic concepts and available methods

• **Parallel sections, parallel loops, tasks**

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

72 OpenMP

Thread parallelism

Distributed memory parallelism

Hybrid/heterogeneous parallelism

Design patterns

What's left

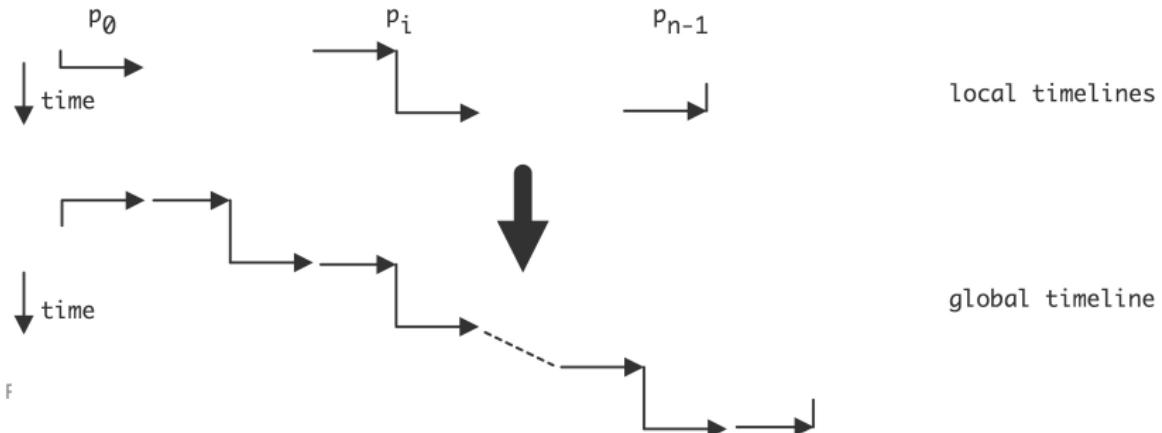
Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers

Distributed memory parallelism

more
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

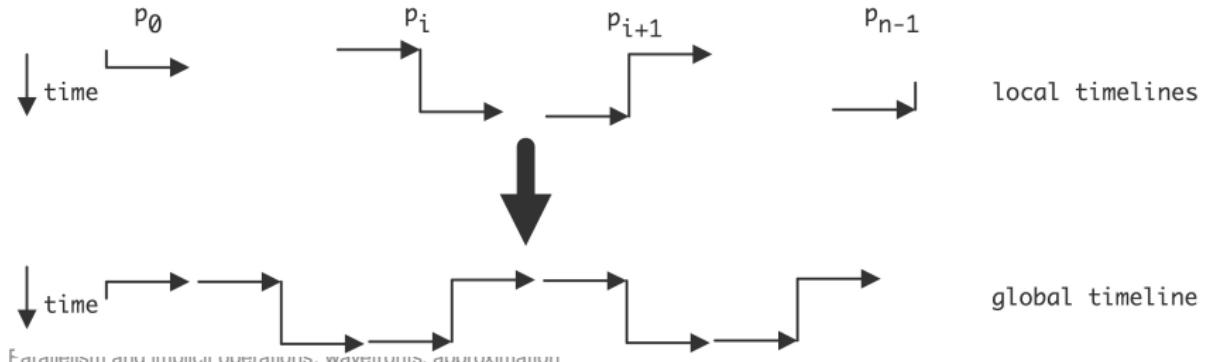
Hybrid/heterogeneous parallelism
Design patterns
What's left

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits



IN-DOMAIN problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
Model formalism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers



Parallelism and implicit operations. Waveforms, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SWIM memory mechanism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Implementation strategy and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

76 Better approaches

Thread parallelism

Distributed memory parallelism

Hybrid/heterogeneous parallelism

Design patterns

What's left

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Hybrid/heterogeneous parallelism

MORE
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Hybrid/heterogeneous parallelism
Design patterns
What's left

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

77 Hybrid computing

The SIMD/MIMD/SPMD SIMD model parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

Thread parallelism

Distributed memory parallelism

Hybrid/heterogeneous parallelism

Design patterns

Matrix multiplication

- **Use MPI between nodes, OpenMP inside nodes**
 - More
Essential aspects of LU factorization
- **alternative: ignore shared memory and MPI throughout**
 - Sparse matrices: storage and algorithms
 - Iterative methods, basic concepts and available methods
 - Collectives as building blocks; complexity
 - Parallel matrix-vector product
 - Sparse matrix-vector product
 - Latency hiding / communication minimizing
 - Computational aspects of iterative methods
 - Parallel LU through nested dissection
 - Incomplete approaches to matrix factorization
- **you save buffers and copying**
- **bundling communication, load spread**

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/MPS/Multithreaded Model

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

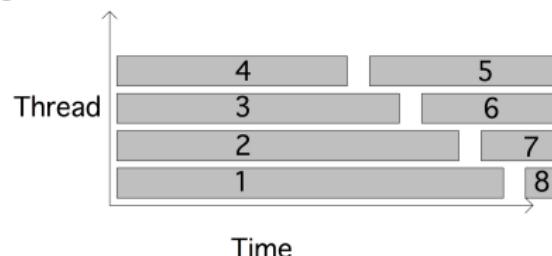
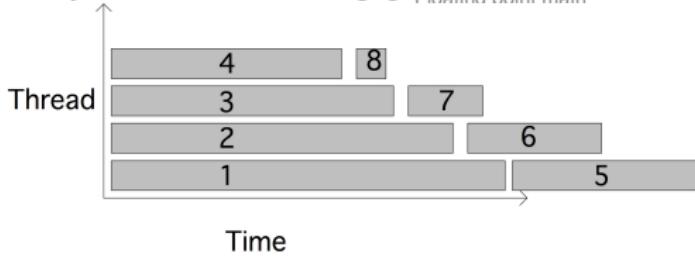
First we dig into bits

Integers

Floating point numbers

Floating point math

Dynamic scheduling gives load balancing



Hybrid is possible improvement over strict-MPI

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

79 Amdahl's law for hybrid programming

The evolution of HPC: Shared memory parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

- p nodes with c cores each

Theoretical parallelism

Distributed memory parallelism

Hybrid/heterogeneous parallelism

Design patterns

What's left

- F_p core-parallel fraction, assume full MPI parallel

- ideal speedup pc , running time $T_1/(pc)$, actually:

Iterative methods, basic concepts and available methods

Collectives as building blocks: complexity

Scalability analysis of dense matrix-vector product

$T_{p,c} = T_1 \left(\frac{F_s + F_p}{p + pc} \right) = \frac{T_1}{pc} (F_s c + F_p) = \frac{T_1}{pc} (1 + F_s(c - 1))$.

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

- $T_1/T_{p,c} \approx p/F_s$, hybrid programming $S_p < 1/F_s$

N-body problems, naive and equivalent formulations

Multicore block algorithms

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers

Design patterns

More
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Hybrid/heterogeneous parallelism
Design patterns
What's left

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

struct { **int** number; **double** xcoord,ycoord; } **_Node**;

Thread parallelism

struct { **double** xtrans,ytrans } **Vector**;

Distributed memory parallelism

typedef struct **_Node*****_Node**;

Hybrid/heterogeneous parallelism

typedef struct **Vector*****Vector**;

Design patterns

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Node ***nodes** = (**node**) **malloc**(**n_nodes*****sizeof**(**struct** **_Node**)

Latency hiding / communication minimizing

) ; Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

80 Array of Structures

Thread parallelism

Distributed memory parallelism

Hybrid/heterogeneous parallelism

Design patterns

What's left

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

Thread parallelism

Distributed memory parallelism

Hybrid/heterogeneous parallelism

Design patterns

What's left

void shift(node the_point, vector by) {

 the_point->xcoord += by->xtrans;

 the_point->ycoord += by->ytrans;

} Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

shift(nodes[i], shift_vector);

Multicore block algorithms

}

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

81 Operations

Operate

in a loop

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

shift(nodes[i], shift_vector);

Multicore block algorithms

}

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SIMT model, parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

Thread parallelism

`node_numbers = (int*) malloc(n_nodes*sizeof(int));`

`node_xcoords = // et cetera`

Distributed memory parallelism

`node_ycoords = // et cetera`

Hybrid/heterogeneous parallelism

Design patterns

What's left

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

`for (i=0; i<n_nodes; i++) {`

Computational aspects: iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

`node_xcoords[i] += shift_vector->xtrans;`

Multicore block algorithms

}

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

82 Along come the 80s

Vector operations

`node_numbers = (int*) malloc(n_nodes*sizeof(int));`

`node_xcoords = // et cetera`

`node_ycoords = // et cetera`

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

`for (i=0; i<n_nodes; i++) {`

Computational aspects: iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

`node_xcoords[i] += shift_vector->xtrans;`

Multicore block algorithms

}

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

83 and the wheel of reinvention turns further

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

PGM/MPI/IMC/SPMV/SIMD/Vector parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Collective operations: MPI, OpenMP, SIMD, CUDA

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Thread parallelism

Distributed memory parallelism

Hybrid/heterogeneous parallelism

Design patterns

What's left

The original design was better for MPI in the 1990s

except when vector instructions (and GPUs) came along in the 2000s

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

- **Memory and network are slow, prevent having to wait for it**

Recursive elimination and pivoting

Iterative methods, basic concepts and available methods

- **Hardware magic, out-of-order execution, caches, prefetching**

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Thread parallelism

Distributed memory parallelism

Hybrid/heterogeneous parallelism

Design patterns

What's left

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

85 Explicit latency hiding

The SIMD/MIMD (SPMD/SIMT) model of parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

$$\forall i \in l_p : y_i = \sum_j a_{ij} x_j.$$

Floating point numbers

Floating point math

Examples

More

Matrix vector product

x needs to be gathered:

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimization

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete and direct matrix factorization

Overlap loads and local operations

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Possible in MPI and Xeon Phi offloading,

very hard to do with caches

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Thread parallelism

Distributed memory parallelism

Hybrid/heterogeneous parallelism

Design patterns

What's left

$$y_i = \left(\sum_{\text{local}} + \sum_{j \text{ not local}} \right) a_{ij} x_j.$$

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers

What's left

More
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Distributed memory parallelism
Hybrid/heterogeneous parallelism
Design patterns
What's left

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/OMT model of parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

- **Co-array Fortran: extensions to the Fortran standard**

- **X10**

Essential aspects of LU factorization

- **Chapel**

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

- **UPC**

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

- **BSP**

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

- **MapReduce**

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

- **Pregel, ...**

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

86 Parallel languages

Thread parallelism

Distributed memory parallelism

Hybrid/heterogeneous parallelism

Design patterns

What's left

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

#define N 100*THREADS Integers

Floating point numbers

Floating point math

Examples

shared int v1[N], v2[N], v1plusv2[N];

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

{ Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

int i; Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

v1plusv2[i]=v1[i]+v2[i];

Incomplete approaches to matrix factorization

}

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

87 UPC example

Thread parallelism

Simultaneous memory parallelism

Hybrid/heterogeneous parallelism

Design patterns

What's left

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/DIPO/SPMD/IMPI model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

More examples

Explicit dimension for ‘images’

Real, dimension(100), codimension[], as X*

Sparse matrices: storage and algorithms

Iteration methods for linear systems and available methods

Collectives as building blocks: complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel iterative methods

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

88 Co-array Fortran example

Thread parallelism

Distributed memory parallelism

Hybrid/heterogeneous parallelism

High performance X

What's left

determined by runtime environment

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

89 Grab bag of other approaches

The GRAB BAG OF APPROACHES

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

- OS-based, data movement induced by cache misses

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives, including sparse vectors

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Thread parallelism

Distributed memory parallelism

Hybrid/heterogeneous parallelism

Design patterns

What's left

- Active messages: application level Remote Procedure Call

(see: Charm++)

Collectives, including sparse vectors

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Table of Contents

Processor Architecture

- Structure of a modern processor

- The SIMD/MIMD/SPMD/SIMT model for parallelism

- Characterization of parallelism by memory model

- Memory hierarchy: caches, register, TLB.

- Interconnects and topologies; theoretical concepts

- Programming models

- Load balancing, locality, space-filling curves

- Multicore issues

- First we dig into bits

- Integers

- Floating point numbers

- Floating point math

- Examples

- The power question

- Essential aspects of LU factorization

- Sparse matrices: storage and algorithms

Parallelism

- Parallelism

- Iterative methods; basic concepts and available methods

- Collectives as building blocks; complexity

- Scalability analysis of dense matrix-vector product

- Basic concepts

- Sparse matrix-vector product

- Latency hiding / communication minimizing

- Computational aspects of iterative methods

- Theoretical concepts

- Parallel LU through nested dissection

- Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

- The SIMD/MIMD/SPMD/SIMT model for parallelism

- Multicore block algorithms

- N-body problems: naive and equivalent formulations

- Graph analytics, interpretation as sparse matrix problems

- Derived datatypes

- Communicator manipulation

- Non-blocking collectives

- One-sided communication

Profiling and debugging; optimization and programming strategies.

Programming models

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Load balancing, locality, space-filling curves

more

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Space-filling curves

Domain partitioning by Fiedler vectors

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD (IMD/PMD) memory model

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

More

Essential aspects of LU factorization

- Application load can change dynamically

e.g., mesh refinement, time-dependent problems

Space-filling curves

Domain partitioning by Fiedler vectors

- Splitting off and merging loads

Sparse matrices: storage and algorithms

Iterative solvers: parallelization and communication

Collectives as building blocks; complexity

- No real software support: write application anticipating load management

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Iteration efficient / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to load balancing

- Initial balancing: graph partitioners

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

90 The load balancing problem

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

91 Load balancing and performance

Le SIMD/MIMD/MPS/MPS/MIMD parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Graph partitioning

Domain partitioning by Fiedler vectors

- Assignment to arbitrary processor violates locality

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods: building blocks, parallelization

Collectives as building blocks; complexity

Locality analysis of the sparse vector product

Sparse matrix-vector product

- Need a dynamic load assignment scheme that preserves locality under load migration

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Space-filling curves

more
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Space-filling curves

Domain partitioning by Fiedler vectors

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

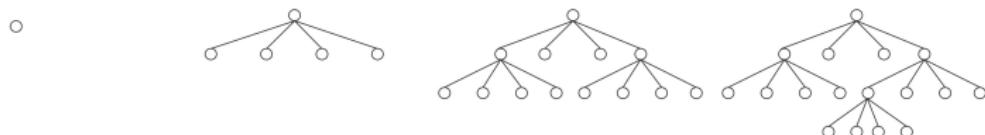
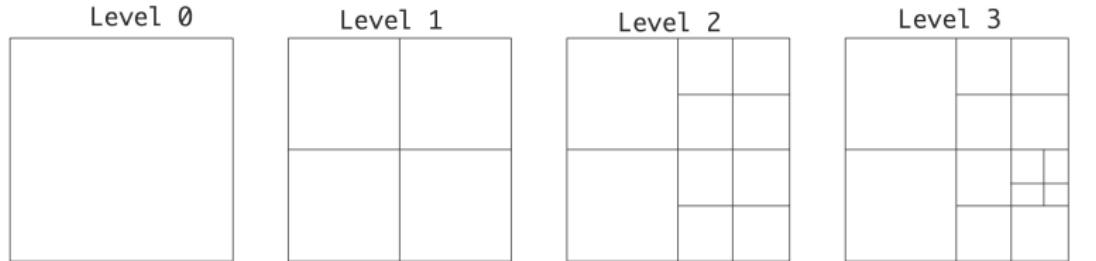
Multicore issues

Programming strategies for performance

The power question

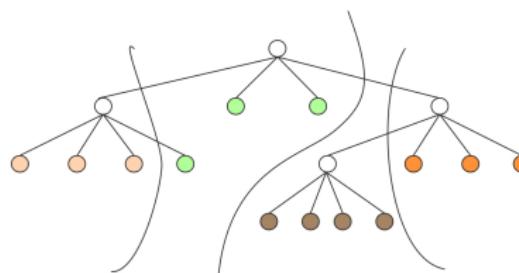
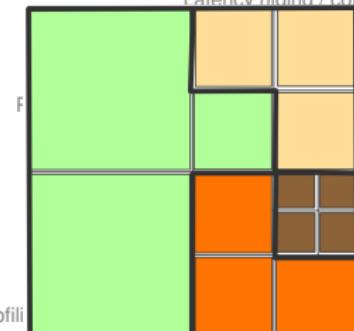
Load assignment

92 Adaptive refinement and load assignment



Sparse matrix-vector product

Latency hiding / communication minimization



Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

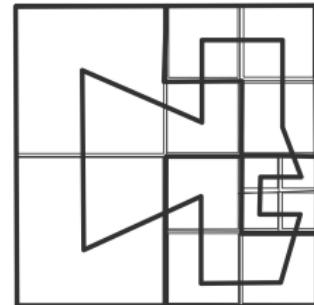
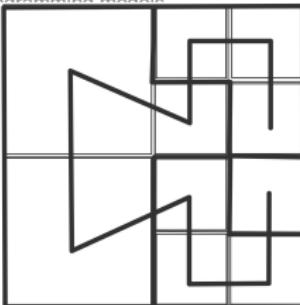
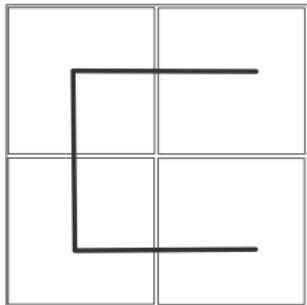
Theoretical concepts

93 Assignment through Space-Filling Curve

Characterization of parallelism by memory model

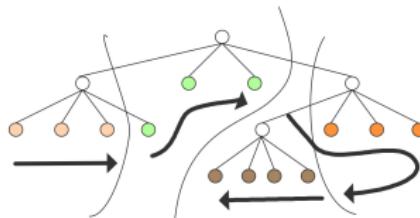
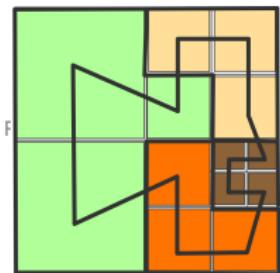
Interconnects and topologies, theoretical concepts

Programming models



Scalability analysis of dense matrix-vector product

Sparse matrix-vector product



Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers

Domain partitioning by Fiedler vectors

more
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Space-filling curves
Domain partitioning by Fiedler vectors

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

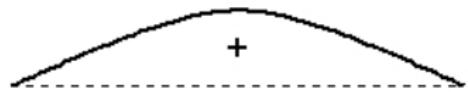
Theoretical concepts

94 Inspiration from physics

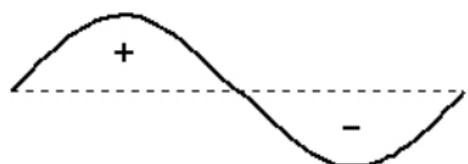
The SIMD/MIMD/PMD SIMD-like parallelism

Characterization of parallelism by memory model

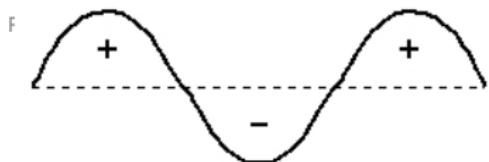
Modes of a Vibrating String



Lowest Frequency $\lambda(1)$



Second Frequency $\lambda(2)$



Third Frequency $\lambda(3)$

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

95 Graph laplacian

The SIMD/MIMD/SPMD/SIMD model, basic ideas

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

- Set $G_{ij} = -1$ if edge (i,j)

Examples
More

Space-filling curves

- Set G_{ii} positive to give zero rowsums

Essential aspects of LU factorization
Sparse matrices: storage and algorithms

Domain partitioning by Fiedler vectors

Iterative methods, basic concepts and available methods

- First eigenvector is zero, positive eigenvector

Scalability analysis of dense matrix-vector product

- Second eigenvector has pos/neg, divides in two

Sparse matrix-vector product
Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel computation through Krylov subspace

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

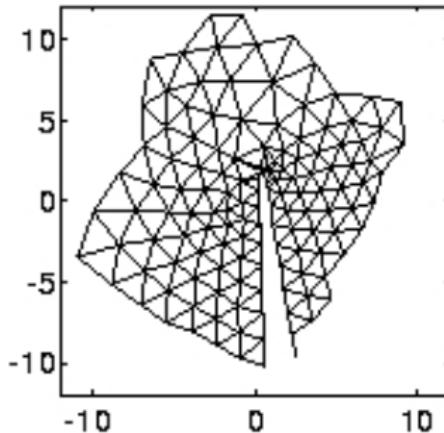
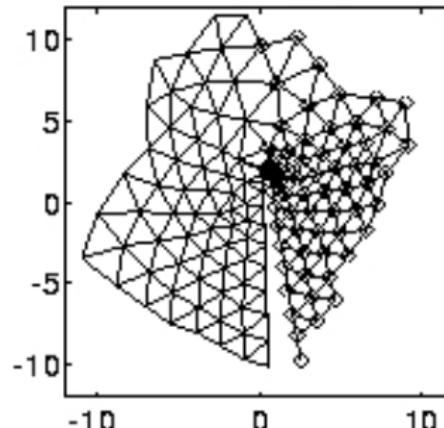
Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

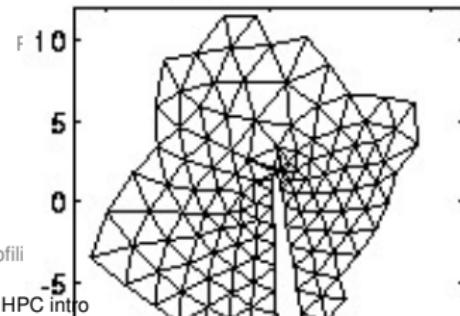
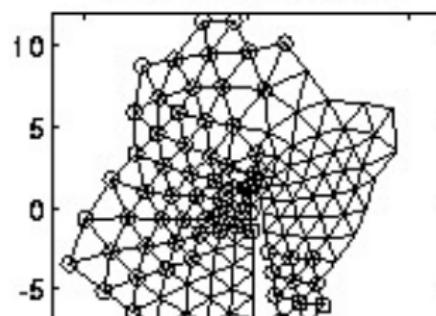
96 Fiedler in a picture

Original FE mesh

Circle node i if $v_2(i) > 0$ 

Latency hiding / communication minimizing

Original FE mesh

Circle node i if $v_4(i) > 0$ 

Computer arithmetic

Structure of a modern processor

Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point FP

Example

More

Space-filling curves

Domain partitioning by Fiedler vectors

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Some practical approaches

Iterative methods, basic concepts and available methods

Conjugate gradient method, conjugate residual

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Justification

This short session will explain the basics of floating point arithmetic, mostly focusing on round-off and its influence on computations.

Space-filling curves

Domain partitioning by Fiedler vectors

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

97 Numbers in scientific computing

SIMD, MIMD, SIMD/DS, SIMD for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

- Integers: $\dots, -2, -1, 0, 1, 2, \dots$

Examples

Space-filling curves

- Rational numbers: $1/3, 22/7$: not often encountered

Essential aspects of LU factorization

LU factorization, storage and algorithms

Iterative methods, basic concepts and available methods

Domain partitioning by Fiedler vectors

- Real numbers $0, 1, -1.5, 2/3, \sqrt{2}, \log 10, \dots$

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Communication avoiding computation methods

Parallel LU through nested dissection

Indirect and direct matrix factorization

Parallel LU through blockwise pivoting

Computers use a finite number of bits to represent numbers,
so only a finite number of numbers can be represented, and no
irrational numbers (even some rational numbers).

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Table of Contents

Processor Architecture

- Structure of a modern processor

The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model

- Memory hierarchy: caches, register, TLB.

Interconnects and topologies; theoretical concepts
Programming models

Load balancing, locality, space-filling curves

- Multicore issues

First we dig into bits

Integers

- Programming strategies for performance

Floating point numbers

Floating point math

Examples

More

- The power question

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Parallelism

85

Iterative methods; basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

- Basic concepts

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

- Theoretical concepts

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

- The SIMD/MIMD/SPMD/SIMT model for parallelism

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

- Characterization of parallelism by memory model

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Programming models

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

First we dig into bits

more

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models

98 Bit operations

	boolean	bitwise (C)	bitwise (F)	bitwise (Py)
and	& &	& Floating point numbers Integers	and	&
or		Floating point math Examples	or	
not	!	More		~
xor		Essential aspects of LU factorization Sparse matrices: storage and algorithms Iterative methods, basic concepts and available methods	leor	

Bit shift operations in C:
Scalability analysis of dense matrix-vector product
Collective matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods

Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
left shift `<<` Multicore block algorithms
right shift `>>` N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

Fortran: mvbits

Communicator manipulation
Non-blocking collectives
One-sided communication

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/MC paradigm introduction

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

More examples

i_times_2 = i<<1; Essential aspects LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

i_mod_8 = i&7 Memory hiding / communication minimizing

Computational aspects of iterative methods

(How does that last one work?)
Parallel LU through nested dissection
Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

99 Arithmetic with bit ops

- Left-shift is multiplication by 2:

i_mod_8 = i&7 Essential aspects LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

i_mod_8 = i&7 Memory hiding / communication minimizing

Computational aspects of iterative methods

(How does that last one work?)
Parallel LU through nested dissection
Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD model, parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More
Essential aspects of LU factorization

Use bit operations to test whether a number is odd or even.

Can you think of more than one way?

Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Table of Contents

Processor Architecture

- Structure of a modern processor

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

- Memory hierarchy: caches, register, TLB.

Interconnects and topologies; theoretical concepts

Programming models

Load balancing, locality, space-filling curves

- Multicore issues

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

- The power question

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Parallelism

85

Iterative methods; basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

- Basic concepts

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

- Theoretical concepts

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

- The SIMD/MIMD/SPMD/SIMT model for parallelism

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

- Interconnects and topologies; theoretical concepts

Profiling and debugging; optimization and programming strategies.

Programming models

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Integers

more

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model of parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More

Scientific computation mostly uses real numbers. Integers are mostly used for array indexing.

We look at
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through block-diagonalization
Incomplete approaches to matrix factorization

- 1. integers as supported by the hardware;**
- 2. integers as they exist in programming languages;**
- 3. (and not software defined integers)**

Parallelism in high level programming languages
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

100 Integers

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/PIMD/SIMT model (parallelism)

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Arithmetic operations

Floating point math

IEEE standard

More

Essential aspects of LU factorization

Sparse matrices storage and algorithms

Iterative methods, basic concepts and available methods

Objectives: speed, memory locality

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Parallel sparse matrix-vector product

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

integer(2) :: i2

integer(4) :: i4

integer(8) :: i8

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communications

Specify the number of decimal digits with selected_int_kind(n).

Profiling and debugging; optimization and programming strategies.

C:

101 In C/C++ and Fortran

- A short int is at least 16 bits;
- An integer is at least 16 bits, but often 32 bits;
- A long integer is at least 32 bits, but often 64;
- A long long integer is at least 64 bits.

Fortran uses kinds, not necessarily equal to number of bytes:

integer(2) :: i2

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

integer(4) :: i4

integer(8) :: i8

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communications

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/Task-based parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Print 2^n for $n = 0, \dots, 31$. There are at least two ways of generating these powers.

Essential aspects of LU and QR factorization

Sparse matrices: storage and algorithms

Collectives as building blocks; basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Also print the bit pattern. What is unexpected?

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Exercise 8: Powers of two

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model; parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models

Problem: Load balancing, locality, space-filling curves
First we dig into bits

Integers

Floating point numbers
Signed integers
Examples

- How do we represent them?
- How do we do efficient arithmetic on them?

Essential aspects of LU factorization
Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product
Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods
Parallel LU factorization: use of

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Non-blocking operations

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

102 Negative integers

$$\text{rep}: \mathbb{Z} \rightarrow 2^n$$

'representation of the number $N \in \mathbb{Z}$ as bitstring of length n '

$$\text{int}: 2^n \rightarrow \mathbb{Z}$$

'interpretation of the bitstring of length n as number $N \in \mathbb{Z}$ '

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts

103 Negative integers

The SIMD/MIMD/SPMD/SIMT model; data parallelism
Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models
Load balancing, locality, space-filling curves

First we dig into bits

Use of sign bit: typically first bit

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization
Sparse matrices: storage and algorithms

s	i ₁	... i _n
---	----------------	--------------------

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithm

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

$$\begin{cases} n \geq 0 & \text{rep}(n) = 0, i_1, \dots, i_{31} \\ n < 0 & \text{rep}(-n) = 1, i_1, \dots, i_{31} \end{cases}$$

Structure of a modern processor
 Memory hierarchy: caches, register, TLB.
 Multicore issues
 Programming strategies for performance
 The power question
 Basic concepts
 Theoretical concepts
 The SIMD/MIMD/SPMD/SIMT model for parallelism
 Characterization of parallelism by memory model
 Interconnects and topologies, theoretical concepts
 Programming models
 Load balancing, locality, space-filling curves
 First we dig into bits
Integers
 Floating point numbers
 Floating point math
 Examples
 More

Interpretation

	Essential aspects of LU factorization					
bitstring	Sparse matrices: storage and algorithms Iterative methods, basic concepts and available methods	00 ··· 0	01 ··· 1	10 ··· 0	...	11 ··· 1
as unsigned int	Collectives as building blocks: complexity Scalability analysis of dense matrix-vector product	0	$2^{31} - 1$	2^{31}	...	$2^{32} - 1$
as naive signed	Sparse matrix-vector product Latency filling, communication minimizing	0	$2^{31} - 1$	-0	...	$-2^{31} + 1$

Computational aspects of iterative methods
 Parallel LU through nested dissection
 Incomplete approaches to matrix factorization
 Parallelism and implicit operations: wavefronts, approximation
 Multicore block algorithms
 N-body problems: naive and equivalent formulations
 Graph analytics, interpretation as sparse matrix problems
 Derived datatypes
 Communicator manipulation
 Non-blocking collectives
 One-sided communication
 Profiling and debugging; optimization and programming strategies.

104 Sign bit

Structure of a modern processor
 Memory hierarchy: caches, register, TLB.
 Multicore issues
 Programming strategies for performance
 The power question
 Basic concepts
 Theoretical concepts
 The SIMD/MIMD/SPMD/SIMT model for parallelism
 Characterization of parallelism by memory model
 Interconnects and topologies, theoretical concepts
 Programming models
 Load balancing, locality, space-filling curves
 First we dig into bits

105 Shifting

Interpret unsigned number n as $n - B$

More

Essential aspects of LU factorization

bitstring	00 ··· 0	01 ··· 1	10 ··· 0	...	11 ··· 1
as unsigned int	0	$2^{31} - 1$	2^{31}	...	$2^{32} - 1$
as shifted int	-2^{31}	-1	0	...	$2^{31} - 1$

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

106 2's Complement

The SIMD/MIMD/SPMD/SIMT molecular parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First weird in bits
Integers

Floating point numbers

Floating point math

Examples

More

- If $0 \leq m < 2^{31}$, the **normal bit pattern** for m is used, that is

Essential aspects of floating-point

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

- For $-2^{31} \leq n \leq -1$, n is represented by the bit pattern for

$2^{32} - |n|$.

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

$-2^{31} \leq n \leq -1 \Rightarrow \text{rep}(m) = 2^{32} - |n|$.

Structure of a modern processor
 Memory hierarchy: caches, register, TLB.
 Multicore issues
 Programming strategies for performance
 The power question
 Basic concepts
 Theoretical concepts
 The SIMD/UMD/SPIR/SD/SOT model of parallelism
 Characterization of parallelism by memory model
 Interconnects and topologies, theoretical concepts
 Programming models
 Load balancing, locality, space-filling curves
 First we dig into bits
Integers
 Floating point numbers
 Floating point math
 Examples
 More

Essential aspects of LU factorization

bitsring	00 ··· 0	... 01 ··· 1	10 ··· 0	... 11 ··· 1
Iterative methods, basic concepts and available methods				
as unsigned int	0 ···	$2^{31} - 1$	2^{31}	... $2^{32} - 1$

Scalability analysis of dense matrix-vector product
 Sparse matrices: storage and algorithms
 Collective as building blocks; complexity
 Scalability analysis of dense matrix-vector product
as 2's comp. integer
 Sparse matrix-vector product
 Latency Hiding & Communication minimizing
 Computational aspects of iterative methods
 Parallel LU through nested dissection
 Incomplete approaches to matrix factorization
 Parallelism and implicit operations: wavefronts, approximation
 Multicore block algorithms
 N-body problems: naive and equivalent formulations
 Graph analytics, interpretation as sparse matrix problems
 Derived datatypes
 Communicator manipulation
 Non-blocking collectives
 One-sided communication
 Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts

The SIMD/MIMD/SPMD/All mode for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Integers

Floating point numbers
Floating point arithmetic examples
More

Problem: processor is very good at arithmetic on (unsigned) bit strings.

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods: basic concepts and available methods
Collectives as building blocks, complexity

Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization

Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms

N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

$$\text{int}(\text{rep}(x) * \text{rep}(y)) \stackrel{?}{=} x * y$$

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More

Add $m + n$, where m, n are representable:

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
More

The easy case is $0 \leq m, n$, as long as there is no overflow.

Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/Vector and GPU model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models

Case $m > 0, n < 0$, and $m + n > 0$. Then $\text{rep}(m) = m$ and
 $\text{rep}(n) = 2^{32} - |n|$, so the unsigned addition becomes

$$\text{rep}(m) + \text{rep}(n) = m + (2^{32} - |n|) = 2^{32} + m - |n|.$$

Since $m - |n| > 0$, this result is $> 2^{32}$.

$m + n = m - |n|$

Parallel addition through nested dissection
Complete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms

N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems

Derived datatypes
Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

However, this is basically $m + n$ with the overflow bit set.

Non-blocking communication
Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD debate: from parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Subtraction $m - n$: Floating point numbers

Floating point math

Examples

More

- Case: $m < n$. Observe that $-n$ has the bit pattern of $2^{32} - n$.

Essential: Laplace transform on

Sparse matrices: storage and algorithms

Iteration, local vs global concepts and available methods

Collectives as building blocks; complexity

Computability analysis of dense matrix multiplication

$2^{32} - (n + m)$ is the 2's complement bit pattern of $m - n$.

Sparse matrix-vector product

Latency hiding / communication minimization

Computational aspects of iterative methods

Parallel algorithm without nested iterations

Incomplete approaches to matrix factorization

Parallelism and interleaving: see for example [1], [2]

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

111 Subtraction in 2's complement

The SIMD/MIMD debate: from parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

- Case: $m < n$. Observe that $-n$ has the bit pattern of $2^{32} - n$.

Essential: Laplace transform on

Sparse matrices: storage and algorithms

Iteration, local vs global concepts and available methods

Collectives as building blocks; complexity

Computability analysis of dense matrix multiplication

$2^{32} - (n + m)$ is the 2's complement bit pattern of $m - n$.

Sparse matrix-vector product

Latency hiding / communication minimization

Computational aspects of iterative methods

Parallel algorithm without nested iterations

Incomplete approaches to matrix factorization

Parallelism and interleaving: see for example [1], [2]

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

There is a limited number of bits, so numbers that are too large in absolute value can not be represented.

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Overflow
Numerical analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Cooperative nature of floating-point

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

112 Overflow

This is not a fatal error: your program continues with the wrong result.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MMI model; SIMD and vector parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Communication efficient LU, generalizations

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Exercise 9: Integer overflow

Investigate what happens when you perform an integer calculation that leads to overflow. What does your compiler say if you try to write down a nonrepresentable number explicitly, for instance in a declaration or assignment statement?

Language lawyer remark: signed integer overflow is Undefined Behavior in C/C++.

Table of Contents

Processor Architecture

- Structure of a modern processor

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

- Memory hierarchy: caches, register, TLB.

Interconnects and topologies; theoretical concepts

Programming models

Load balancing, locality, space-filling curves

- Multicore issues

First we dig into bits

Integers

- Programming strategies for performance

Floating point numbers

Floating point math

Examples

More

- The power question

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Parallelism

85

Iterative methods; basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

- Basic concepts

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

- Theoretical concepts

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

- The SIMD/MIMD/SPMD/SIMT model for parallelism

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

- Interconnects and topologies; theoretical concepts

Profiling and debugging; optimization and programming strategies.

Programming models

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Floating point numbers

more

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MM/MIMD/Multicore parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

113 Floating point math is hard!

And the consequences if you get it wrong can be considerable.

Upvote icon r/formula1 · Posted by u/dogryan100 · Aston Martin · 11 months ago · 5 comments

2.0k [OT Roborace] Driverless racecar drives straight into a wall
clips.twitch.tv/FunAma...



F

255 Comments Award Share Save Hide Report 98% Upvoted

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SIMD/SIMD in OpenMP parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Parallelism vs. sequentiality

First we dig into bits

Integers

$$x = \pm \sum_{i=0}^{t-1} d_i \beta^{-i} \beta^e$$

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Analysis of dense matrix-vector product

Sparse matrix-vector product

Implementation / fine-grained mapping

Computational aspects of iterative methods

- sign bit
- β is the base of the number system

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation
one digit before the radix point, so mantissa < β

- $0 \leq d_i \leq \beta - 1$ the digits of the mantissa:
- $0 \leq d_i \leq \beta - 1$ the digits of the mantissa:
 $f_l(L) = 0$

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

114 Floating point numbers

Analogous to scientific notation $x = 6.022 \cdot 10^{23}$:

First we dig into bits

Integers

$$x = \pm \sum_{i=0}^{t-1} d_i \beta^{-i} \beta^e$$

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Analysis of dense matrix-vector product

Sparse matrix-vector product

Implementation / fine-grained mapping

Computational aspects of iterative methods

- sign bit
- β is the base of the number system

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation
one digit before the radix point, so mantissa < β

- $0 \leq d_i \leq \beta - 1$ the digits of the mantissa:
- $0 \leq d_i \leq \beta - 1$ the digits of the mantissa:
 $f_l(L) = 0$

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/UMA/SPM/SHM models for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

	β	t	L	U
IEEE single (32 bit)	2	23	-126	127
		Integers floating point numbers		
IEEE double (64 bit)	2	53	-1022	1023
		Floating point math Examples		
Old Cray 64bit	2	48	-16383	16384
		More Essential aspects of LU factorization		
IBM mainframe 32 bit	16	6	-64	63
		Sparse matrices: storage and algorithms Iterative methods, basic concepts and available methods		
packed decimal	10	50	-999	999
		Collectives as building blocks: complexity Scalability analysis of dense matrix-vector product		

Sparse matrix-vector product

Latency hiding / communication minimizing

Communication locality: locality methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics; interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

One-sided collective collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

115 Examples of floating point systems

BCD is tricky: 3 decimal digits in 10 bits

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics; interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

One-sided collective collectives

One-sided communication

Internal processing in 80 bit

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts

Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples

Overflow: more than $\beta(1 - \beta^{-t+1})\beta^U$ or less than $-\beta(1 - \beta^{-t+1})\beta^U$

Essential aspects of LU factorization
Sparse matrices: storage and algorithms

Underflow: positive numbers less than β^L

Iterative methods: basic concepts and available methods
Collectives as building blocks, complexity

Scalability analysis of dense-matrix vector product
Sparse matrix-vector product

Gradual underflow: $\beta^{t+1} \cdot \beta^L$
Latency hiding / communication minimizing
Computational aspects of iterative methods

Parallel LU through nested dissection
Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms

N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems

Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

116 Limitations

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

Theorem 10.1: The SISIM model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Example

More

For real numbers x, y , the quantity $g = \sqrt{(x^2 + y^2)/2}$ satisfies

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Exercise 10: Floating point overflow

so it is representable if x and y are. What can go wrong if you compute g using the above formula? Can you think of a better way?

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD SIMD Thread Library

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Inf - Inf → NaN

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

also 0/0 or √1

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

117 Other exceptions

Overflow: Inf

Essential aspects of LU factorization

Inf - Inf → NaN

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

also 0/0 or √1

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/Single-processor like

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Implementation of collective operations

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

118 The normalization problem

Do we allow

$$1 \cdot 100 \cdot 10^0, \quad 0 \cdot 110 \cdot 10^1, \quad 0.011 \cdot 10^2?$$

This makes testing for equality hard.

Solution: normalized numbers have one nonzero before the radix point.

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

heterogeneous memory management

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices, storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks, complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding, communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism in linear algebra, applications

Multicore block algorithms

Matrix multiplication, linear algebra, applications

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

119 Normalized floating point numbers

Require first digit in the mantissa to be nonzero.

Equivalent: mantissa part $1 \leq x_m < \beta$

Unique representation for each number,

also: in binary this makes the first digit 1, so we don't need to store that.

(do you see a problem?)

With normalized numbers, underflow threshold is $1 \cdot \beta^L$;

'gradual underflow' possible, but usually not efficient.

Structure of a modern processor
 Memory hierarchy: caches, register, TLB.
 Multicore issues
 Programming strategies for performance
 The power question
 Basic concepts
 Theoretical aspects
 The SIMD/MIMD/BFVb/SIMT model for parallelism
 Characterization of parallelism by memory model
 Interconnects and topologies, theoretical concepts
 Programming models
 Load balancing, locality, space-filling curves
 First we dig into bits
 Integers
Floating point numbers
 Floating point math
 Examples

sign	exponent	mantissa
	More Essential aspects of LU factorization	
p Iterative methods, basic concepts and available methods	$e = e_1 \dots e_8$	$s = s_1 \dots s_{23}$
31 Scalability analysis of dense matrix-vector product	30 Collective vector building blocks; complexity	$22 \dots 0$
\pm Latency hiding / communication minimizing Computational aspects of iterative methods	$2e^{-127}$ Sparse matrix-vector product (except $e = 0, 255$)	$s_1 \cdot 2^{-1} + \dots + s_{23} \cdot 2^{-23}$

Parallel LU through nested dissection
 Incomplete approaches to matrix factorization
 Parallelism and implicit operations: wavefronts, approximation
 Multicore block algorithms
 N-body problems: naive and equivalent formulations
 Graph analytics, interpretation as sparse matrix problems
 Derived datatypes
 Communicator manipulation
 Non-blocking collectives
 One-sided communication
 Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretic concepts

The SIMD/MIMD/MPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

$(e_1 \dots e_8)$

numerical value

integers

Floating point numbers

$s_1 \cdot s_{23} \times 2^{-126}$

Floating point math

Examples

Matrix

$(0 \dots 0) = 0$

$\pm 0 \cdot s_1 \cdot s_{23} \times 2^{-126}$

$(0 \dots 01) = 1$

$\pm 1 \cdot s_1 \cdot s_{23} \times 2^{-126}$

$(0 \dots 010) = 2$

$\pm 1 \cdot s_1 \cdot s_{23} \times 2^{-125}$

Iterative methods, basic concepts and available methods

Collectives as building blocks: complexity

Stability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

$(01111111) = 127$

LU through nested iteration

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

$(11111110) = 254$

Multi-block sparse matrix

$(11111111) = 255$

Non-prime problems, naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

$s_1 \dots s_{23} = 0 \Rightarrow \infty$

$s_1 \dots s_{23} \neq 0 \Rightarrow \text{NaN}$

range

$$s = 0 \dots 01 \Rightarrow 2^{-23} \cdot 2^{-126} = 2^{-149} \approx 10^{-45}$$

$$s = 1 \dots 11 \Rightarrow (1 - 2^{-23}) \cdot 2^{-126}$$

$$s = 0 \dots 01 \Rightarrow 1 \cdot 2^{-126} \approx 10^{-37}$$

$$s = 0 \dots 00 \Rightarrow 1 \cdot 2^0 = 1$$

$$s = 0 \dots 01 \Rightarrow 1 + 2^{-23} \cdot 2^0 = 1 + \varepsilon$$

$$s = 1 \dots 11 \Rightarrow (2 - 2^{-23}) \cdot 2^0 = 2 - \varepsilon$$

$$s = 0 \dots 00 \Rightarrow 1 \cdot 2^1 = 2$$

et cetera

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts

The SIMD/MIMD/SPMD shared memory parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Note that the exponent doesn't come at the end. This has an interesting consequence.

Integers
Floating point numbers
Floating point math
Examples
More

Essential aspects of LU factorization
Parallelization of linear algebra algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallelization through nested dissection
Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

0...0111

as int? What as float?
What is the largest integer that is representible as float?

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD SIMD model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

- There is a 64-bit format, with 53 bits mantissa.

- IEEE envisioned a sliding scale of precisions: see Intel 80-bit registers

- Half precision, and recent invention bfloat16

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

FP32	s	8 bit exp	23 bit mantissa
FP16	s	5 bit exp	10 bit mantissa
BF16	s	8 bit exp	7 bit mantissa

BFP10001

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Table of Contents

Processor Architecture

- Structure of a modern processor

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

- Memory hierarchy: caches, register, TLB.

Interconnects and topologies; theoretical concepts

Programming models

Load balancing, locality, space-filling curves

- Multicore issues

First we dig into bits

Integers

- Programming strategies for performance

Floating point numbers

Floating point math

Examples

More

- The power question

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Parallelism

85

Iterative methods; basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

- Basic concepts

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

- Theoretical concepts

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

- The SIMD/MIMD/SPMD/SIMT model for parallelism

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

- Interconnects and topologies; theoretical concepts

Profiling and debugging; optimization and programming strategies.

Programming models

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Floating point math

more

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/MIMD/MIMD/MIMD
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math

Error between number x and representation \tilde{x} :

absolute $|x - \tilde{x}|$ or $|\tilde{x} - x|$ More
Essential aspects of LU factorization

relative $\frac{|\tilde{x} - x|}{x}$ or $|\frac{\tilde{x} - x}{x}|$ Examples
Sparse matrix storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity

Equivalent: $\tilde{x} = x + \epsilon \Leftrightarrow |x - \tilde{x}| \leq \epsilon \Leftrightarrow \tilde{x} \in [x - \epsilon, x + \epsilon]$.

Also: $\tilde{x} = x(1 + \epsilon)$ often shorthand for $|\frac{\tilde{x} - x}{x}| \leq \epsilon$

Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms

N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

123 Representation error

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model of parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Precision with
Examples

Decimal, $t = 3$ digit mantissa: let $x = 1.256$, $\tilde{x}_{\text{round}} = 1.26$,
 $\tilde{x}_{\text{truncate}} = 1.25$ More

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods: matrix elements and available methods

Error in the 4th digit.
Collectives as building blocks; complexity
Scalability analysis of these matrix-vector product
Sparse matrix-vector product

Latency hiding / communication minimizing
Computational aspects of iterative methods

Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

124 Example

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SIMD/MC paradigm
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
 ↳ IEEE
Floating point numbers
 ↳ IEEE
Floating point math
 ↳ IEEE
Examples
More
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
 ↳ Sparse matrix-vector product
 ↳ stencil hiding / communication minimizing
 ↳ Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: vector-only approximations
 ↳ Multicore block algorithms
 ↳ N-body problems: on-the-fly and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
 ↳ Derived datatypes
 ↳ Communicator manipulation
 ↳ Non-blocking collectives
 ↳ One-sided communication
Profiling and debugging; optimization and programming strategies.

Exercise 12: Round-off

The number $e \approx 2.72$, the base for the natural logarithm, has various definitions. One of them is

$$e = \lim_{n \rightarrow \infty} (1 + 1/n)^n. \quad (2)$$

Write a single precision program that tries to compute e in this manner. (Do not use the `pow` function: code the power explicitly.)

Evaluate the expression for an upper bound $n = 10^k$ for some k . (How far do you let k range?) Explain the output for large n . Comment on the behavior of the error.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/IMD model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Memory access locality, space-time access

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Parallel sparse matrix-vector product algorithms

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

32-bit single precision: $mp \approx 10^7$

Latency hiding / communication minimizing

64-bit double precision: $mp \approx 10^{-16}$

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multigrid and multilevel algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Another definition of machine precision: smallest number ϵ such that

$1 + \epsilon > 1$.

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

125 Machine precision

Any real number can be represented to a certain precision:

$\tilde{x} = x(1 + \epsilon)$ where

truncation: $\epsilon = \beta^{-t+1}$

rounding: $\epsilon = \frac{1}{2}\beta^{-t+1}$

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Parallel sparse matrix-vector product algorithms

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

32-bit single precision: $mp \approx 10^7$

Latency hiding / communication minimizing

64-bit double precision: $mp \approx 10^{-16}$

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multigrid and multilevel algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Another definition of machine precision: smallest number ϵ such that

$1 + \epsilon > 1$.

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD MMV (VLIW) model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

More

Factorizations: LU factorization

Sparse matrices: storage and algorithms

Iterative methods: Jacobi, Gauss-Seidel, GMRES

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Lateness hiding; communication minimizing

Computational aspects of iterative methods

Parallel LU through Nested Dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Exercise 13: Machine epsilon

Write a small program that computes the machine epsilon for both single and double precision. Does it make any difference if you set the compiler optimization levels low or high?

(For C++ programmers: can you write a templated program that works for single and double precision?)

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves

1. align exponents First we dig into bits
 Integers
2. add mantissas Floating point numbers
 Floating point math
 Examples
3. adjust exponent to normalize Essential aspects of LU factorization

Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Example: $1.00 + 2.00 \times 10^{-2} = 1.00 + .02 = 1.02$. This is exact, but
what happens with $1.00 + 2.55 \times 10^{-2}$?
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through recursive dissection
Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
Memory problems: available, equivalent, invariants

Example: $5.00 \times 10^1 + 5.04 = (5.00 + 0.504) \times 10^1 \rightarrow 5.50 \times 10^1$

Graph analytics, interpretation as sparse matrix problems
Derived data types
Communicator manipulation
Non-blocking collectives
One-sided communication
Any error comes from limiting the mantissa: if x is the true sum and \tilde{x} the computed sum, then $\tilde{x} = x(1 + \varepsilon)$ with $|\varepsilon| < 10^{-2}$

Profiling and debugging; optimization and programming strategies.

126 Addition

127 The ‘correctly rounded arithmetic’ model

Assumption (enforced by IEEE 754):

The numerical result of an operation is the rounding of the exactly computed result.

$$\text{Scalability analysis of dense matrix-vector product} \quad (x_1 \odot x_2)(1 + \varepsilon)$$

where $\odot = +, -, *, /$

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Note: this holds only for a single operation!

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model of parallelism

Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First field into bits
Integers

Floating point numbers

Floating point math

Example

More

Correctly rounding is not trivial, especially for subtraction.

Example: $t = 2, \beta = 10: 1.0 - 9.5 \times 10^{-1}$, exact result

$0.05 = 5.0 \times 10^{-2}$,
Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

- Simple approach:
Scalability analysis of dense matrix-vector product

$$1.0 - 9.5 \times 10^{-1} = 1.0 - 0.9 = 0.1 = 1.0 \times 10^{-1}$$

- Using 'guard digit':
Computational aspects of iterative methods
Incomplete LU, high nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations, wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

128 Guard digits

$1.0 - 9.5 \times 10^{-1} = 1.0 - 0.95 = 0.05 = 5.0 \times 10^{-2}$, exact.

In general 3 extra bits needed

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMV/UMV model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, load distribution, load leveling

First we dig into bits

Integers

Floating point numbers

$c \leftarrow a * b + c$

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Matrix-matrix multiplication, matrix-vector multiplication

Sparse matrix-vector product

Inter-core hiding communication, reductions

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analysis: computation, storage, matrix-vector multiplication

Derived datatypes

Collective manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

129 Fused Mul-Add instructions

(also ‘fused multiply-accumulate’)

First we dig into bits

Integers

Floating point numbers

$c \leftarrow a * b + c$

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Matrix-matrix multiplication, matrix-vector multiplication

Sparse matrix-vector product

Inter-core hiding communication, reductions

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analysis: computation, storage, matrix-vector multiplication

Derived datatypes

Collective manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

- Addition plus multiplication, but not independent

- Processors can have dedicated hardware for FMA (also IEEE 754-2008)

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analysis: computation, storage, matrix-vector multiplication

Derived datatypes

Collective manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

- Internally evaluated in higher precision: 80-bit.

- Very useful for certain linear algebra (which?) Not for other

operations (examples?)

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The lower question

Basic concepts

Theoretical concepts

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

$$\begin{aligned} (4 \cdot 10^0 + 6 \cdot 10^0) + 7 \cdot 10^0 &\xrightarrow{\text{Integers}} 10 \cdot 10^0 + 7 \cdot 10^0 \\ &\xrightarrow{\text{Floating point numbers}} 1 \cdot 10^1 + 7 \cdot 10^0 \\ &\xrightarrow{\text{Floating point math}} 1.0 \cdot 10^1 + 0.7 \cdot 10^1 \\ &\xrightarrow{\text{Examples}} 1.0 \cdot 10^1 + 0.7 \cdot 10^1 \\ &\xrightarrow{\text{More}} 1.7 \cdot 10^1 \\ &\xrightarrow{\text{Essential aspects of LU factorization}} 1.7 \cdot 10^1 \\ &\xrightarrow{\text{Sparse matrices: storage and algorithms}} 1.7 \cdot 10^1 \\ &\xrightarrow{\text{Iterative methods, basic concepts and available methods}} 2 \cdot 10^1 \\ &\xrightarrow{\text{Collectives as building blocks; complexity}} 2 \cdot 10^1 \\ &\xrightarrow{\text{Scalability analysis of dense matrix-vector product}} 2 \cdot 10^1 \\ &\xrightarrow{\text{Sparse matrix-vector product}} 2 \cdot 10^1 \end{aligned}$$

addition

rounding

using guard digit

rounding

On the other hand, evaluation right-to-left gives:

$$\begin{aligned} 4 \cdot 10^0 + (6 \cdot 10^0 + 7 \cdot 10^0) &\xrightarrow{\text{Incomplete approaches to matrix factorization}} 4 \cdot 10^0 + 13 \cdot 10^0 \\ &\xrightarrow{\text{Parallel LU through nested dissection}} 4 \cdot 10^0 + 1 \cdot 10^1 \\ &\xrightarrow{\text{Incomplete approaches to matrix factorization}} 4 \cdot 10^0 + 1 \cdot 10^1 \\ &\xrightarrow{\text{Parallelism and implicit operations: wavefronts, approximation}} 4 \cdot 10^0 + 1 \cdot 10^1 \\ &\xrightarrow{\text{Multicore block algorithms}} 4 \cdot 10^0 + 1 \cdot 10^1 \\ &\xrightarrow{\text{N-body problems: naive and equivalent formulations}} 0.4 \cdot 10^1 + 1.0 \cdot 10^1 \\ &\xrightarrow{\text{Graph analytics, interpretation as sparse matrix problem}} 0.4 \cdot 10^1 + 1.0 \cdot 10^1 \\ &\xrightarrow{\text{Derived datatypes}} 1.4 \cdot 10^1 \\ &\xrightarrow{\text{Communicator manipulation}} 1.4 \cdot 10^1 \\ &\xrightarrow{\text{Non-blocking collectives}} 1 \cdot 10^1 \\ &\xrightarrow{\text{One-sided communication}} 1 \cdot 10^1 \\ &\xrightarrow{\text{Profiling and debugging; optimization and programming strategies.}} 1 \cdot 10^1 \end{aligned}$$

addition

rounding

using guard digit

rounding

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MMD/SIMD typical case

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

Let $s = x_1 + x_2$, and $x = \tilde{s} = \tilde{x}_1 + \tilde{x}_2$ with $\tilde{x}_i = x_i(1 + \varepsilon_i)$

First we dig into bits
Integers

Floating point numbers

$\tilde{x} = \tilde{s}(1 + \varepsilon_3)$

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks: complexity

Scalability analysis of dense matrix-vector product

$\Rightarrow \tilde{x} = s(1 + 2\varepsilon)$

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

Graph problems, divide and conquer, multi-level

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

131 Error propagation under addition

The SIMD/MMD/SIMD typical case

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits
Integers

Floating point numbers

$\tilde{x} = \tilde{s}(1 + \varepsilon_3)$

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks: complexity

Scalability analysis of dense matrix-vector product

$\Rightarrow \tilde{x} = s(1 + 2\varepsilon)$

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

Graph problems, divide and conquer, multi-level

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
Memory access patterns
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples

1. add exponents More
Essential aspects of LU factorization
2. multiply mantissas Sparse matrices: storage and algorithms
Iterative Methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
3. adjust exponent Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of LU method
Parallel LU through nested dissection
Incomplete approaches to matrix factorization

Example: $.123 \times .567 \times 10^1 = .069741 \times 10^1 \rightarrow .69741 \times 10^0 \rightarrow .697 \times 10^0.$

What happens with relative errors?

N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

132 Multiplication

Table of Contents

Processor Architecture

- Structure of a modern processor

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

- Memory hierarchy: caches, register, TLB.

Interconnects and topologies; theoretical concepts

Programming models

Load balancing, locality, space-filling curves

- Multicore issues

First we dig into bits

Integers

- Programming strategies for performance

Floating point numbers

Floating point math

Examples

More

- The power question

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Parallelism

Iterative methods; basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

- Basic concepts

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

- Theoretical concepts

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

- The SIMD/MIMD/SPMD/SIMT model for parallelism

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

- Interconnects and topologies; theoretical concepts

Profiling and debugging; optimization and programming strategies.

Programming models

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Examples

more

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models

Correct rounding only applies to a single operation.

Example: $1.24 - 1.23 = 0.01 \rightarrow 1. \times 10^{-2}$:
result is exact, but only one significant digit.

What if $1.24 = f(1.244)$ and $1.23 = f(1.225)$? Correct result 1.9×10^{-2} ; almost 100% error.

- **Cancellation leads to loss of precision**
 - Computational aspects of iterative methods
 - Incomplete LU through nested dissection
 - Incomplete approaches to matrix factorization
 - **subsequent operations with this result are inaccurate**
 - Multicore block algorithms
 - Minsky problem; naive and equivalent formulations
 - Graph analytics, interpretation as sparse matrix problems
 - Derived datatypes
 - Locality-aware computation
 - Non-blocking collectives
 - One-sided communication
 - \Rightarrow avoid subtracting numbers that are likely close.
- Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT memory parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More
Example: $ax^2 + bx + c = 0 \rightarrow x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector products
Better: compute $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ and use $x_+ \cdot x_- = -c/a$.
Sparse matrix-vec product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

134 ABC-formula

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Technical concepts

The SIMD/MIMD/SPMD SIMD mode is dominant

Characterization of parallelism by memory model

Multicore aspects and topologies, technical concepts

Programming models

Load balancing, locality, space filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices storage and algorithms

Iterative methods: basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

n^2 Sparse matrix-vector product

Latency hiding / communication minimizing

$(n-1)^2 = n^2 - 2n + 1$

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Randomized iterative solvers, frontends, approximation

Multicore block algorithms

$n-1 \cdot 00 \cdots 0 \quad 10 \cdots 00$

N-body problems: native and equivalent reformulations

Graph analytics: interpretation as sparse matrix problems

$n \cdot 00 \cdots 0 \quad 10 \cdots 01 \quad 0 \cdots 0$

Derived datatypes

Communication minimization

Non-blocking collectives

One-sided communication

Profiling and prefetching, optimization analysis, tuning strategies

135 Serious example

Evaluate $\sum_{n=1}^{10000} \frac{1}{n^2} = 1.644834$

in 6 digits: machine precision is 10^{-6} in single precision

First term is 1, so partial sums are > 1 , so $1/n^2 < 10^{-6}$ gets ignored, \Rightarrow last 7000 terms (or more) are ignored \Rightarrow sum is 1.644725: 4 correct digits

Solution: sum in reverse order; exact result in single precision

Why? Consider ratio of two terms:

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

n^2 Sparse matrix-vector product

Latency hiding / communication minimizing

$(n-1)^2 = n^2 - 2n + 1$

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Randomized iterative solvers, frontends, approximation

Multicore block algorithms

$n-1 \cdot 00 \cdots 0 \quad 10 \cdots 00$

N-body problems: native and equivalent reformulations

Graph analytics: interpretation as sparse matrix problems

$n \cdot 00 \cdots 0 \quad 10 \cdots 01 \quad 0 \cdots 0$

Derived datatypes

Communication minimization

Non-blocking collectives

One-sided communication

Profiling and prefetching, optimization analysis, tuning strategies

$K = \log(n/2)$ positions

The last digit in the smaller number is not lost if $n < 2/\epsilon$

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

Consider $y_n = \int_0^1 \frac{x^n}{x-5} dx = \frac{1}{n} - 5y_{n-1}$ (monotonically decreasing)

$$y_0 = \ln 6 - \ln 5.$$

Floating point numbers

Floating point math

Examples

In 3 decimal digits:

More

computation

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

$$y_0 = \ln 6 - \ln 5 = .1821322 \times 10^1 \dots$$

Collectives as building blocks; complexity

$$y_1 = .900 \times 10^{-1}$$

Some variants of dense matrix-vector product

$$y_2 = .500 \times 10^{-1}$$

Sparse matrix-vector product

$$y_3 = .830 \times 10^{-1}$$

Computational aspects of iterative methods

$$y_4 = -.165$$

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Multicore block algorithms

Parallelism and implicit operations: wavefronts, approximation

Numpy: object-oriented and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

$$\tilde{y}_n = 1/n - 5\tilde{y}_{n-1} = 1/n + 5n_{n-1} + 5\varepsilon_{n-1} = y_n + 5\varepsilon_{n-1}$$

Non-blocking collectives

One-sided communication

Profiling and debugging: optimization and root-cause analysis.

so $\varepsilon_n \geq 5\varepsilon_{n-1}$: exponential growth.

correct result

1.82

.884

.0580

going up? .0431

negative? .0343

Reason? Define error as $\tilde{y}_n = y_n + \varepsilon_n$, then

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Non-blocking collectives

One-sided communication

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/IMC and SIMD model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

Problem: solve $Ax = b$, where b is exact.

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrix-vector multiplication algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Lateness hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Distributed matrices

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

137 Stability of linear system solving

Since $Ax = b$, we get $A\Delta x = \Delta b$. From this,

$$\left\{ \begin{array}{l} Ax = b \\ \Delta x = A^{-1}\Delta b \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} \|A\| \|x\| \geq \|b\| \\ \|\Delta x\| \leq \|A^{-1}\| \|\Delta b\| \end{array} \right.$$

$$\frac{\|\Delta x\|}{\|x\|} \leq \|A\| \|A^{-1}\| \frac{\|\Delta b\|}{\|b\|}$$

'Condition number'. Attainable accuracy depends on matrix properties

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD divide in memory hierarchies
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Multiplication and addition are not associative: problems for parallel computations.

Floating point numbers
Floating point arithmetic
Examples
More
Essential aspects of LU factorization
Source matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks for complexity
Scalability analysis of dense matrix-vector product

compute $a+b+c+d$

sequential **parallel**

Sparse matrix-vector product

Laziness hiding computation by communication
Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Randomized linear computation: randomization

Multicore block algorithms

Matrix inversion at the turn of the century

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Operations with “same” outcomes are not equally stable: matrix inversion is unstable, elimination is stable

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/Vector/Pipeline model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Consider the iteration

Integers
Floating point numbers
Floating point math

Examples

More

Essential aspects of linearization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Residual minimization, splitting, domain decomposition

Multicore block algorithms

Graph problems, sparse matrix computations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Exercise 14: Fixed-point iteration

$$x_{n+1} = f(x_n) = \begin{cases} 2x_n & \text{if } 2x_n < 1 \\ 2x_n - 1 & \text{if } 2x_n \geq 1 \end{cases}$$

Does this function have a fixed point, $x_0 \equiv f(x_0)$, or is there a cycle

$x_1 = f(x_0), x_0 \equiv x_2 \equiv f(x_1)$ et cetera?

Now code this function and see what happens with various starting

points x_0 . Can you explain this?

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/Vector/Pipeline model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Integers
Floating point numbers
Floating point math

Examples

More

Essential aspects of linearization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Residual minimization, splitting, domain decomposition

Multicore block algorithms

Graph problems, sparse matrix computations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Table of Contents

Processor Architecture

- Structure of a modern processor
 - The SIMD/MIMD/SPMD/SIMT model for parallelism
 - Characterization of parallelism by memory model
- Memory hierarchy: caches, register, TLB.
- Multicore issues
 - Load balancing, locality, space-filling curves
 - First we dig into bits
 - Integers
- Programming strategies for performance
- The power question
 - Essential aspects of LU factorization
 - Sparse matrices: storage and algorithms
 - Iterative methods: basic concepts and available methods
 - Collectives as building blocks; complexity
 - Scalability analysis of dense matrix-vector product

Parallelism 85

- Basic concepts
 - Sparse matrix-vector product
 - Latency hiding / communication minimizing
 - Computational aspects of iterative methods
 - Parallel LU through nested dissection
 - Incomplete approaches to matrix factorization
- Theoretical concepts

Parallelism and implicit operations: wavefronts, approximation

- The SIMD/MIMD/SPMD/SIMT model for parallelism
 - Multicore block algorithms
 - N-body problems: naive and equivalent formulations
 - Graph analytics, interpretation as sparse matrix problems
- Characterization of parallelism by memory model
 - Derived datatypes
 - Communicator manipulation
 - Non-blocking collectives
 - One-sided communication
- Interconnects and topologies: theoretical concepts

Profiling and debugging; optimization and programming strategies.

Programming models

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers

More

More

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

139 Complex numbers

The SIMD/MIMD/SPMD/CP model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Storage:

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks: complexity

Scalability analysis of dense matrix-vector product

- **Store real/imaginary adjacent: easy to pass address of one number**

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel iterative methods: parallel matrix-vector product

Incomplete approaches to matrix factorization

Parallelism in sparse direct solvers: Multifrontal method

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MMX/SSE/MVSB/AVX vector parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
More

Some compilers support higher precisions.

Essential aspects of LU factorization
Arbitrary precision: GMPlib
Sparse matrices and algorithms
Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Interval arithmetic
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods

Half precision **bfloat16**
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms

N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

140 Other arithmetic systems

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Random numbers

More

Partial Differential Equations

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Partial differential equations are an important source of large-scale engineering problems. Here we take a look at their computational aspects.

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods: basic concepts and available methods

Collectives as building blocks, complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Justification

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/MISW view of parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

$$\begin{cases} -u''(x) = f(x, u, u') & x \in [a, b] \\ u(a) = u_a, u(b) = u_b \end{cases}$$

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Boundary value problems

Consider in 1D

in 2D:

$$\begin{cases} -u_{xx}(\bar{x}) - u_{yy}(\bar{x}) = f(\bar{x}) & \bar{x} \in \Omega = [0, 1]^2 \\ u(\bar{x}) = u_0 & \bar{x} \in \delta\Omega \end{cases}$$

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

Approximation of 2nd order derivatives

Taylor series (write h for δx):

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

$$u(x+h) = u(x) + u'(x)h + u''(x)\frac{h^2}{2!} + u'''(x)\frac{h^3}{3!} + u^{(4)}(x)\frac{h^4}{4!} + u^{(5)}(x)\frac{h^5}{5!} + \dots$$

Load balancing, locality, space-filling curves

First we dig into bits

and

Integers

Floating point numbers

Floating point math

$$u(x-h) = u(x) - u'(x)h + u''(x)\frac{h^2}{2!} - u'''(x)\frac{h^3}{3!} + u^{(4)}(x)\frac{h^4}{4!} - u^{(5)}(x)\frac{h^5}{5!} + \dots$$

Essential aspects of LU factorization

Subtract: Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product
Sparse matrix-vector product

Latency hiding / communication minimizing

so Computational aspects of iterative methods

Parallel LU (high-level description)

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems

$$\frac{u(x+h) + 2u(x) + u(x-h)}{h^2} - \frac{u^{(4)}(x)h^4}{12} = f(x, u(x), u'(x))$$

Derived datatypes

(communicator manipulation)

Non-blocking collectives

One-sided communication

Profiling and debugging, optimization and programming strategies.
(2nd order PDEs are very common!)

Structure of a modern processor
 Memory hierarchy: caches, register, TLB.
 Multicore issues
 Programming strategies for performance
 The power question
 Basic concepts
 Theoretical concepts
 The SIMD/MIMD/SIMD/MIMD parallelism
 Characterization of parallelism by memory model
 Interconnects and topologies, theoretical concepts
 Programming models
 Load balancing, locality, space-filling curves

$$-u_{xx} = f \rightarrow \frac{2u(x) - u(x+h) - u(x-h)}{h^2} = f(x, u(x), u'(x))$$

First we get into bits
Integers
Floating point numbers
Floating point math

Equally spaced points on $[0, 1]$: $x_k = kh$ where $h = 1/(n+1)$, then

$$-u_{k+1} + 2u_k - u_{k-1} = 1/h^2 f(x_k, u_k, u'_k) \quad \text{for } k = 1, \dots, n$$

Essential aspects of LU factorization
 Sparse matrices: storage and algorithms
 Iterative methods: basic concepts and available methods
 Collectives as building blocks; complexity
 Scalability analysis of dense matrix-vector product
 Sparse matrix-vector product
 Latency hiding by communication minimizing
 Computational aspects of iterative methods
 Parallel LU through nested dissection
 Incomplete approaches to matrix factorization
 Parallelism and implicit operations: wavefronts, approximation
 N-body problems: naive and equivalent formulations
 Graph analytics, interpretation as sparse matrix problems
 Derived datatypes
 Communicator manipulation
 Non-blocking collectives
 One-sided communication

Profiling and debugging; optimization and programming strategies.

$$\begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & 2 & -1 & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix} = \begin{pmatrix} f_1 + u_0 \\ f_2 \\ \vdots \\ f_n \end{pmatrix}$$

Written as matrix equation:

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

Matrix properties

The SIMD/MIMD/SPMD/SIMT paradigm

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

- **Very sparse, banded**
- **Symmetric (only because 2nd order problem)**

Sparse matrices: storage and algorithms

- **Sign pattern: positive diagonal, nonpositive off-diagonal (true for many second order methods)**

Collectives as building blocks; complexity

Parallel matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Parallel Cholesky factorization

- **Positive definite (just like the continuous problem)**

- **Constant diagonals (from constant coefficients in the DE)**

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SIMT paradigm for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves

Sparse matrices so far were tridiagonal: only in 1D case.

Two-dimensional: $-u_{xx} - u_{yy} = f$ on unit square $[0, 1]^2$
Integers
Floating point numbers
Examples
More

Difference equation:
Four first aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts approximation

$$4u(x, y) - u(x+h, y) - u(x-h, y) - u(x, y+h) - u(x, y-h) = h^2 f(x, y)$$

$$4u_k - u_{k+1} - u_{k-1} - u_{k-n} - u_{k+n} = f_k$$

Consider a graph where $\{u_k\}_K$ are the edges
and (u_i, u_j) is an edge iff $a_{ij} \neq 0$

Multicore block algorithms
Graph analytics, interpretation as sparse matrix problems
derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SIMT/DIMT model of parallelism

Characterization of parallelism by memory model

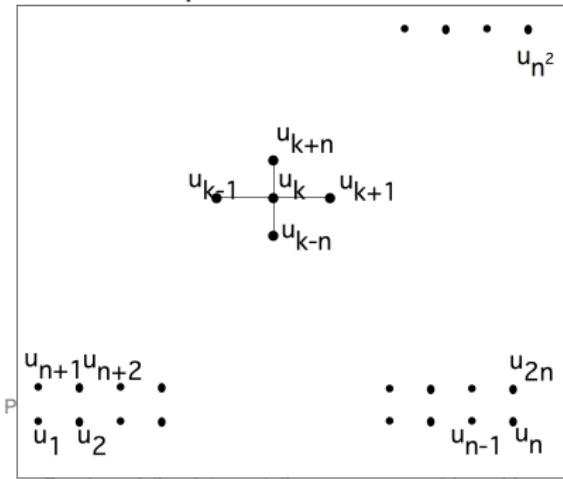
Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Poisson eq:



This is a graph!

This is the (adjacency) graph of a sparse matrix.

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model; parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

- **Very sparse, banded**

We dig into bits

Integers

Floating point numbers

Floating point math

Examples

Matrix

Essential aspects of LU factorization

Sparse matrix-vector multiplication algorithms

- **Symmetric (only because of the second order problem)**

Iterative methods, basic concepts and available methods

Conjugate gradient, biconjugate gradient, complexity

Scalability analysis of dense matrix-vector product

- **Positive definite (just like the continuous problem)**

Latency hiding / communication minimizing

Computational aspects of iterative methods

Conjugate gradient, biconjugate gradient

Incomplete approaches to matrix factorization

Parallelism: load balancing, communication, approximation

Multicore block algorithms

- **Factorization: lower complexity than dense, recursion length less than N .**

Derived datatypes

Communicator manipulation

Non-blocking collectives

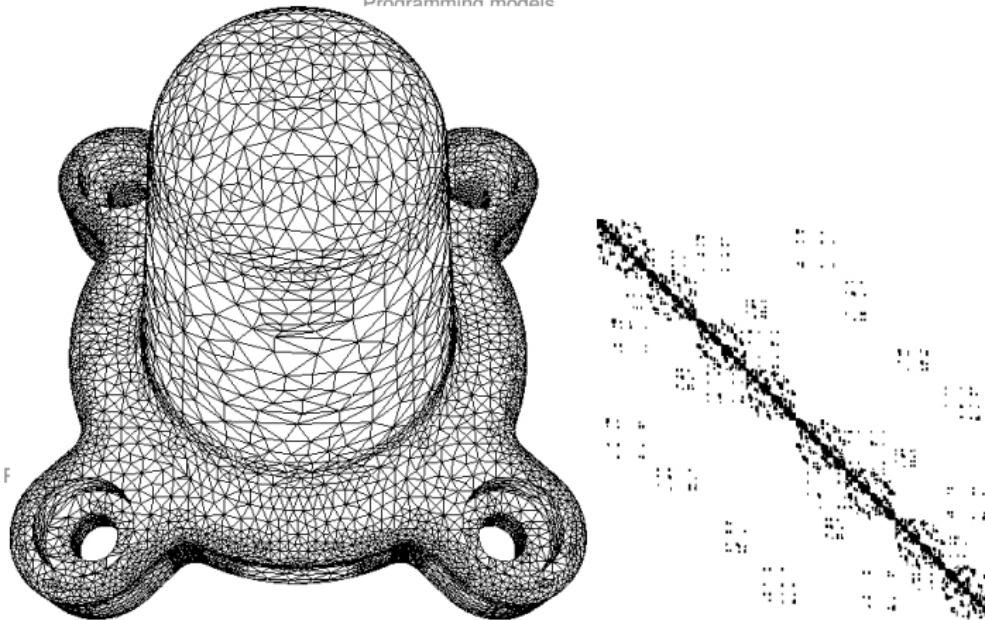
One-sided communication

Profiling and debugging; optimization and programming strategies.

Matrix properties

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT memory model
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models

Realistic meshes



Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Linear algebra

- More
 - Essential aspects of LU factorization
 - Sparse matrices: storage and algorithms
 - Iterative methods, basic concepts and available methods
 - Collectives as building blocks; complexity
 - Scalability analysis of dense matrix-vector product
 - Sparse matrix-vector product
 - Latency hiding / communication minimizing
 - Computational aspects of iterative methods
 - Parallel LU through nested dissection
 - Incomplete approaches to matrix factorization
 - Parallelism and implicit operations: wavefronts, approximation
 - Multicore block algorithms
 - N-body problems: naive and equivalent formulations
 - Graph analytics, interpretation as sparse matrix problems
 - Derived datatypes
 - Communicator manipulation
 - Non-blocking collectives
 - One-sided communication
 - Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of matrix factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks: complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency Hiding / Communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Justification

Many algorithms are based in linear algebra, including some non-obvious ones such as graph algorithms. This session will mostly discuss aspects of solving linear systems, focusing on those that have computational ramifications

- Structure of a modern processor

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

- Memory hierarchy: caches, register, TLB.

Load balancing, locality, space-filling curves

- Multicore issues

First we dig into bits

Integers

- Programming strategies for performance

Floating point numbers

Floating point math

Examples

More

- The power question

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Parallelism 85

- Basic concepts

Iterative methods: basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

- The SIMD/MIMD/SPMD/SIMT model for parallelism

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

- Interconnects and topologies; theoretical concepts

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Essential aspects of LU factorization

more

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model of parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math

- Mathematical aspects: mostly linear system solving
 - More
- Practical aspects: even simple operations are hard
 - Essential aspects of LU factorization
 - Sparse matrices: storage and algorithms
 - Iterative methods, basic concepts and available methods
- Dense matrix-vector product: scalability aspects
 - Collectives as building blocks, complexity
- Sparse matrix-vector: implementation
 - Sparse matrix-vector product
 - Latency hiding / communication minimizing
 - Computational aspects of iterative methods
 - Implementation of LU: nested dissection
 - Incomplete approaches to matrix factorization

Let's start with the math...
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Linear algebra

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

In SIMD, MIMD, GPU, cloud parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Solve $Ax = b$

Load balancing, locality, space-filling curves

First we dig into bits

Direct methods:

Integers

Floating point numbers

Floating point math

Examples

More

- **Deterministic**

Essential aspects of LU factorization

- **Exact up to machine precision**

Sparse matrices: storage and algorithms

- **Expensive (in time and space)**

Iterative methods: basic concepts and available methods

Collectives as building blocks; complexity

Sparsification, nested dissection, domain partitioning

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Iterative methods:

LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

- **Only approximate**

N-body problems, naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

- **Cheaper in space and (possibly) time**

Communicator manipulation

Non-blocking collectives

- **Convergence not guaranteed**

One-sided communication

Profiling and debugging; optimization and programming strategies.

Two approaches to linear system solving

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

Vector/MV/SMV/Optimal parallelization

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks: complexity

Scalability analysis of dense matrix-vector product

$x_i = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix} / |A|$

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Time complexity $O(n^3)$ Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/MC/MIMC/MC model: MIMD parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More

$$Ax = b$$

Essential aspects of LU factorization

Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity

- Compute explicitly A^{-1} ,
Scalability analysis of first matrix-vector product
Sparse matrix-vector product
Latency hiding, communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
- then $x \leftarrow A^{-1}b$
Implementation of matrix inversion
- Numerical stability issues.

Parallelism and implicit operations: wavefronts, approximation

- Amount of work?
 N -body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes

Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

Not a good method either

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More
Essential aspects of LU decomposition
Sparse matrices: storage, reordering, etc.
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

A close look linear system solving: direct methods

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

Gaussian elimination

The SIMD/MIMD/SPMV vs. LU factorization list
Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point path

Examples

Core

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

$\begin{bmatrix} 6 & -2 & 1 & | & 16 \\ 12 & -8 & 6 & | & 26 \\ 3 & -13 & 3 & | & -19 \end{bmatrix}$

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

$\begin{bmatrix} 6 & -2 & 2 & | & 16 \\ 0 & -4 & 2 & | & -6 \\ 0 & 12 & 2 & | & -27 \end{bmatrix}$

Latency hiding + communication minimizing

Computational aspects of iterative methods

Solve x_3 , then x_2 , then x_1 through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

6, -4, -4 are the ‘pivots’ Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

Gaussian elimination, step by step

The LU factorization in memory parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Locality, scaling, locality, space-filling curves

First we dig into bits

Integers

\langle eliminate values in column k \rangle

Floating point numbers

Floating point math

Examples

\langle eliminate values in column k \rangle

More
Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

\langle compute multiplier for row i \rangle

Scalability analysis of dense matrix-vector product

\langle update row i \rangle

Latency hiding / communication minimizing

\langle compute multiplier for row i \rangle

Computational aspects of iterative methods

Parallelization through nested dissection

Incomplete approaches to matrix factorization

Parallelism in implicit operations: wavefronts, approximation

$a_{ik} \leftarrow a_{ik}/a_{kk}$

Multicore block algorithms

\langle update row i \rangle

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

\langle for $j = k + 1$ to n : \rangle

Derived datatypes

Communicator manipulation

$a_{ij} \leftarrow a_{ij} - a_{ik} * a_{kj}$

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD model and its shortcomings

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

(LU factorization): First we dig into bits

Integers

for $k = 1, n - 1$: Floating point numbers

Floating point math

for $i = k + 1$ to n : Examples

More

$a_{ik} \leftarrow a_{ik} / a_{kk}$ Essential steps of LU factorization

Sparse matrices: storage and algorithms

Iterative methods: basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

$n-1$ Multicore block $n+1$ algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

$\sum_{k=1}^{n-1} \sum_{i,j>k} 1 = \sum_k (n-k)^2 \approx \sum_k k^2 \approx n^3/3$

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Gaussian elimination, all together

(LU factorization): First we dig into bits

Integers

for $k = 1, n - 1$: Floating point numbers

Floating point math

for $i = k + 1$ to n : Examples

More

$a_{ik} \leftarrow a_{ik} / a_{kk}$ Essential steps of LU factorization

Sparse matrices: storage and algorithms

Iterative methods: basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

$n-1$ Multicore block $n+1$ algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

$\sum_{k=1}^{n-1} \sum_{i,j>k} 1 = \sum_k (n-k)^2 \approx \sum_k k^2 \approx n^3/3$

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor

Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

More

Elementary matrix and factorization

Sparse matrices: storage and algorithms

Algorithmic issues, systematics, parallelization

Collectives as building blocks; complexity

Sparse matrix-vector product

Timing / communication minimizing

Computational aspects of iterative methods

Conjugate gradient, GMRES, BiCGSTAB, etc.

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Pivoting

If a pivot is zero, exchange that row and another.

(there is always a row with a nonzero pivot if the matrix is nonsingular)

best choice is the largest possible pivot

in fact, that's a good choice even if the pivot is not zero:

partial pivoting

(full pivoting would be row and column exchanges)

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model; parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space filling curves

First we dig into bits

$$\begin{pmatrix} \varepsilon & 1 \\ 1 & 1 \end{pmatrix}^x = \begin{pmatrix} 1 + \varepsilon \\ 2 \end{pmatrix}$$

Floating point numbers

Floating point math

Examples

More

Consider

with solution $x = (1, 1)^t$

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

We can now solve x_2 and from it x_1 :

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

$$\left\{ \begin{array}{l} x_2 = (1 - \varepsilon^{-1}) / (1 - \varepsilon^{-1}) = 1 \\ \text{Derived datatypes} \\ \text{Communicator manipulation} \end{array} \right.$$

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Roundoff control

$\left\{ \begin{array}{l} x_1 = \varepsilon^{1/2} (1 + \varepsilon - x_2) = 1 \\ \text{Non-blocking collectives} \end{array} \right.$

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Batch processing

Theoretical concepts

Roundoff 2

If $\varepsilon < \varepsilon_{\text{mach}}$, then in the rhs $1 + \varepsilon \rightarrow 1$, so the system is:

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First welding into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Eliminating:

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

$\begin{pmatrix} \varepsilon & 1 \\ 0 & 1 \end{pmatrix} x = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$ Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelizing matrix operations: wavefront computation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

$$\left\{ \begin{array}{l} x_2 = \varepsilon^{-1}/\varepsilon^{-1} = 1 \\ x_1 = \varepsilon^1(1 - 1 \cdot x_2) = \varepsilon^{-1} \cdot 0 = 0, \end{array} \right.$$

Derived datatypes

Communication minimization

Non-blocking collectives

One-sided communication

Profiling and debugging, optimization and programming practices

so x_2 is correct, but x_1 is completely wrong.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models

Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples

$$\begin{pmatrix} 1 & 1 \\ \epsilon & 1 \end{pmatrix} x = \begin{pmatrix} 2 \\ 1+\epsilon \end{pmatrix}$$

More
Essential aspects of LU factorization
Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product
Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

$$x_2 = \frac{1-\epsilon}{1-\epsilon} = 1.$$

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Roundoff 3

Pivot first:

$$x_1 = 2 - x_2 = 1$$

Now we get, regardless the size of epsilon:

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Locality, memory locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Example

More

$A = \begin{pmatrix} 6 & -2 & 2 \\ 12 & -8 & 6 \\ 3 & -13 & 3 \end{pmatrix}$

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks, complexity

Scalability analysis of dense matrix-vector product

2nd row minus $2 \times$ first; 3rd row minus $1/2 \times$ first;
equivalent to

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

$L_1 Ax = L_1 b$ multicomputer, algorithm

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

(elementary reflector) non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

$$A = \begin{pmatrix} 6 & -2 & 2 \\ 12 & -8 & 6 \\ 3 & -13 & 3 \end{pmatrix}$$

$$L_1 = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -1/2 & 0 & 1 \end{pmatrix}$$

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Define $U = L_2 L_1 A$, then $A = LU$ with $L = L_1^{-1} L_2^{-1}$

Incomplete LU factorization

Incomplete approaches to matrix factorization

Parallelism in matrix operations: elements per iteration

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

LU 2

$$L_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -3 & 1 \end{pmatrix}$$

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

Observe:

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

$$L_1 = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -1/2 & 0 & 1 \end{pmatrix}$$

Load balancing, locality, space-filling curves
First word in bits
Integers
Floating point numbers
Floating point math

Examples

More

Likewise

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches: matrix factorization

Even more remarkable:

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communication manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging, optimization and problem solving strategies

Can be computed in place! (pivoting?)

LU 3

$$L_1^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1/2 & 0 & 1 \end{pmatrix}$$

$$L_2^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 3 & 1 \end{pmatrix}$$

Lower triangular!

$$L_1^{-1} L_2^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1/2 & 3 & 1 \end{pmatrix}$$

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model of parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

$Ax = b \rightarrow LUx = b$ solve in two steps:

$Ly = b$, and $Ux = y$

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithm

Iterative methods, basic concepts and available methods

Collective re-building blocks; complexity

Scalability analysis of dense matrix-vector product

Dense matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations, wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Solve LU system

Forward sweep:

$$\begin{pmatrix} 1 & & & \\ l_{21} & 1 & & \\ l_{31} & l_{32} & 1 & \\ \vdots & & \ddots & \\ l_{n1} & l_{n2} & \cdots & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model of parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

$Ax = b \rightarrow LUx = b$ solve in two steps:

$Ly = b$, and $Ux = y$

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithm

Iterative methods, basic concepts and available methods

Collective re-building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations, wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

$$y_1 = b_1, \quad y_2 = b_2 - \ell_{21}y_1, \dots$$

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

$$\begin{pmatrix} 1 & & & \\ \ell_{21} & 1 & & \\ \ell_{31} & \ell_{32} & 1 & \\ \vdots & & \ddots & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

$$y_1 = b_1, \quad y_2 = b_2 - \ell_{21}y_1, \dots$$

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

$(\begin{matrix} u_{11} & u_{12} & \dots & u_{1n} \\ u_{21} & u_{22} & \dots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ u_{n1} & u_{n2} & \dots & u_{nn} \end{matrix})$

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Solve LU 2

Backward sweep:

$$\left(\begin{array}{cccc} u_{11} & u_{12} & \dots & u_{1n} \\ u_{21} & u_{22} & \dots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ u_{n1} & u_{n2} & \dots & u_{nn} \end{array} \right) \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

Structure of a modern processor

Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

$(\begin{matrix} u_{11} & u_{12} & \dots & u_{1n} \\ u_{21} & u_{22} & \dots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ u_{n1} & u_{n2} & \dots & u_{nn} \end{matrix})$

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

(Compute inverses once, store)

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Solve LU 2

Backward sweep:

Integers

Floating point numbers

Floating point math

$$\left(\begin{matrix} u_{11} & u_{12} & \dots & u_{1n} \\ u_{21} & u_{22} & \dots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ u_{n1} & u_{n2} & \dots & u_{nn} \end{matrix} \right) \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

$$x_n = y_n, \quad x_{n-1} = u_{n-1,n-1}^{-1} (y_{n-1} - u_{n-1,n} x_n), \dots$$

Multicore block algorithms

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SMP/SPU memory model

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

$$\begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks, complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

Hard problems and difficult formulations

Graph analytics, interpretation as sparse matrix problems

Sparsity patterns

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Computational aspects

Compare:

Matrix-vector product:

Solving LU system:

$$\begin{pmatrix} a_{11} & & \emptyset \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

(and similarly the *U* matrix)

Compare operation counts. Can you think of other points of comparison? (Think modern computers.)

- Structure of a modern processor

The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model

- Memory hierarchy: caches, register, TLB.

Interconnects and topologies; theoretical concepts
Programming models

Load balancing, locality, space-filling curves

- Multicore issues

First we dig into bits
Integers

- Programming strategies for performance

Floating point numbers
Floating point math
Examples

- The power question

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

- Parallelism85
- Iterative methods: basic concepts and available methods
Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

- Basic concepts

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

- Theoretical concepts

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

- The SIMD/MIMD/SPMD/SIMT model for parallelism

N-body problems: naive and equivalent formulations

- Characterization of parallelism by memory model

Derived datatypes

Communicator manipulation

- Interconnects and topologies; theoretical concepts

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Table of Contents

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Sparse matrices: storage and algorithms

more

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

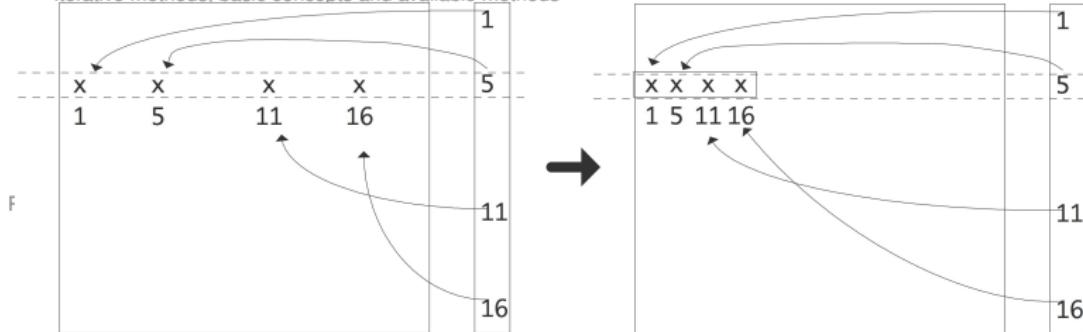
Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SIMD+MIMD model of parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts

Matrix above has many zeros: n^2 elements but only $O(n)$ nonzeros.
Big waste of space to store this as square array.

Matrix is called 'sparse' if there are enough zeros to make specialized storage feasible.

Computational aspects of LU factorization
Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods



One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Compressed Row Storage

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

$$A = \begin{pmatrix} 10 & 0 & 0 & 0 & -2 & 0 \\ 3 & 9 & 0 & 0 & 0 & 3 \\ 0 & 7 & 8 & 7 & 0 & 0 \\ 3 & 0 & 8 & 7 & 5 & 0 \\ 0 & 8 & 0 & 9 & 9 & 13 \\ 0 & 4 & 0 & 0 & 2 & -1 \end{pmatrix}. \quad (3)$$

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Compressed Row Storage (CRS): store all nonzeros by row, their

column indices, pointers to where the columns start (1-based

indexing): Computational aspects of iterative methods

Latency hiding / communication minimizing

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

val	10	-2	3	9	3	7	8	7	3	9	13	4	2	-1
col_ind	1	5	1	2	6	2	3	4	1	5	6	2	5	6
row_ptr	1	3	6	9	13	17	20							

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD Model / SIMD and MIMD modeling parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More

- Simplest, and important in many contexts: matrix-vector product.
- Matrix-matrix product rare in engineering science
very important in Deep Learning
 - Essential aspects of LU factorization
 - Sparse matrices: direct and iterative methods
 - Iterative methods, basic concepts and available methods
 - Conjugate gradient building blocks: complexity
 - Scalability analysis of dense matrix-vector product
 - LU factorization: direct vs iterative
 - Latency hiding / communication minimizing
- Gaussian elimination is a complicated story.
- In general: changes to sparse structure are hard!

Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/Multicore model of parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Parallel algorithms

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

Now

```
aptr = 0;  
for (row=0; row<nrows; row++) {  
    s = 0;  
    for (col=0; col<ncols; col++) {  
        s += a[aptr] * x[col];  
        aptr++;  
    }  
    y[row] = s;  
}
```

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

Communication

Reuse? Locality? Cachelines?

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SIMD hybrid model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Load balancing, locality, space-filling curves

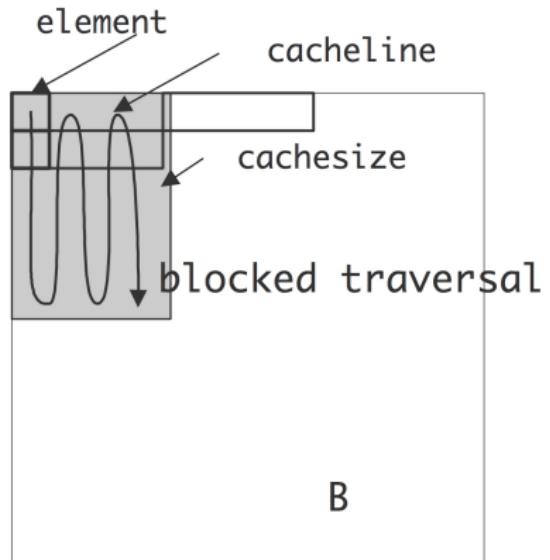
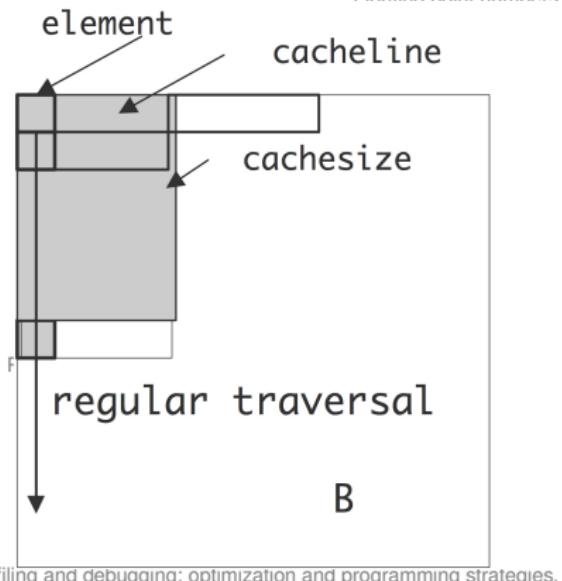
Conversion of integers into bits

Integers

Floating point numbers

Three loops: block, columns inside block, row;
permute blocks to outermost

Better implementation



Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
 Memory hierarchy: caches, register, TLB.
 Multicore issues
 Programming strategies for performance
 The power question
 Basic concepts
 Theoretical concepts
 The SIMD/MIMD divide and its relevance
 Characterization of parallelism by memory model
 Interconnects and topologies, theoretical concepts
 Programming models
`aptr = 0;` Load balancing, locality, space-filling curves
`for (row=0; row<nrows; row++) {`
 `s = 0;` First we dig into bits
 `for (icol=ptr[row]; icol<ptr[row+1]; icol++) {`
 `int col = ind[icol];` Examples
 `s += a[aptr] * x[col];` More
 `}` Essential aspects of LU factorization
 `Sparse matrices: storage and algorithms`
 Iterative methods: basic concepts and available methods
 `Collectives as building blocks; complexity`
 `Scalability analysis of dense matrix-vector product`
 `aptr++;` Sparse matrix-vector product
`}` Latency hiding / communication minimizing
`Computational aspects of iterative methods`
`y[row] = s;` Parallel LU through nested dissection
`Incomplete approaches to matrix factorization`
`}` Parallelism and implicit operations: wavefronts, approximation
`Multicore block algorithms`
`N-body problems: naive and equivalent formulations`
`Graph analytics, interpretation as sparse matrix problems`
`Derived datatypes`
`CORBA, Cato, manipulation`
`Non-blocking collectives`
`One-sided communication`

Again: Reuse? Locality? Cachelines?

Indirect addressing of x gives low spatial and temporal locality.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/APM/SIMT model of parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks: complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Exercise: sparse coding

What if you need access to both rows and columns at the same time?
Implement an algorithm that tests whether a matrix stored in CRS
format is symmetric. Hint: keep an array of pointers, one for each row,
that keeps track of how far you have progressed in that row.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMV/MATMUL/BLAS Level 3

Characterization of parallelism by memory model

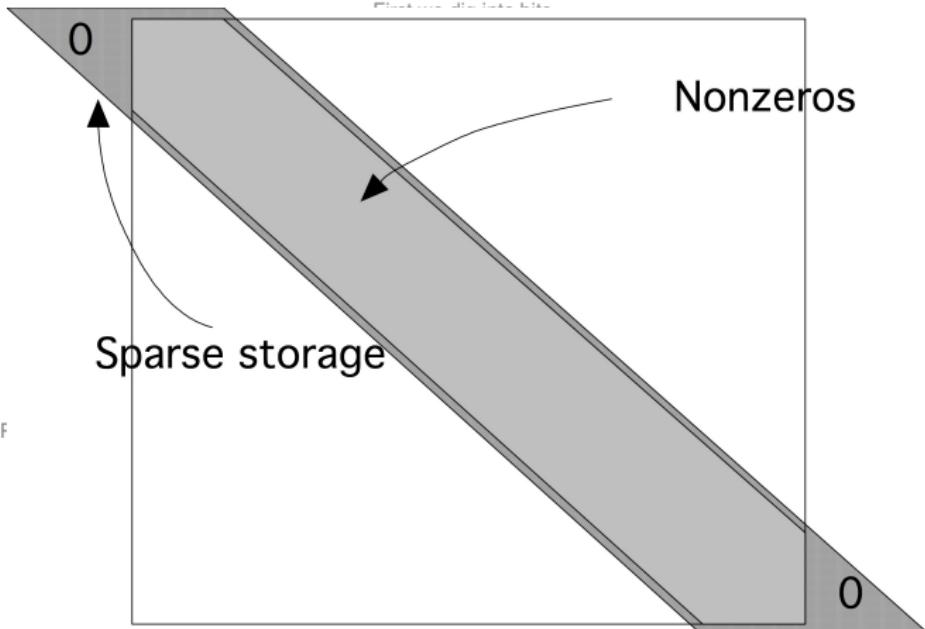
Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

Storage by diagonals

Use the banded format:



Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

Diagonal matrix-vector product

The SIMD model and SIMD-like model for parallelism
Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

$$y_i \leftarrow y_i + A_{ii}x_i,$$

Floating point numbers

Floating point math

$$y_i \leftarrow y_i + A_{ii+1}x_{i+1} \quad \text{for } i < n,$$

More Examples

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

latency+bandwidth minimization

Computational aspects of iterative methods

for diag = diag_left, diag_right
for loc = max(1,1-diag), min(n,n-diag)

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavyfronts, approximation

Multicore block algorithms

end
Naive problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

Pro: long vectors (D'Azevedo: 20th speedup on Cray X-1)

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks, complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Communication minimization

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Pro/con

Con: limited, little cache reuse

Variants: jagged diagonal

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

Jagged diagonal storage

The SIMD/MIMD/PIMD paradigm for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Align irregular sparse matrices along ‘jagged’ diagonals

Integers
floating point numbers

Floating point math

Examples

More

$$\left(\begin{array}{cccccc} 10 & -3 & 0 & 1 & 0 & 0 \\ 0 & 9 & 6 & 0 & 2 & 0 \\ 3 & 0 & 8 & 7 & 0 & 0 \\ 0 & 6 & 0 & 7 & 5 & 4 \\ 0 & 0 & 0 & 0 & 9 & 13 \\ 0 & 0 & 0 & 0 & 5 & -1 \end{array} \right) \rightarrow \left(\begin{array}{ccc} 10 & -3 & 1 \\ 9 & 6 & -2 \\ 3 & 8 & 7 \\ 6 & 7 & 5 \\ 9 & 13 & 4 \\ 5 & -1 & \end{array} \right)$$

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability: analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

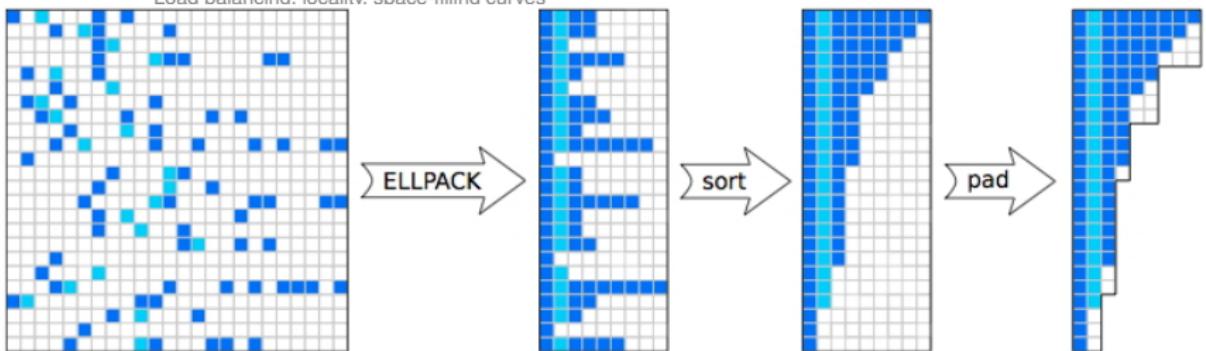
Characterization of parallelism by memory model

Memory access patterns, cache management

Programming models

Load balancing, locality, space-filling curves

Long vectors make it suitable for GPU



Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

Remember Gaussian elimination algorithm:

Integers

Floating point numbers

Floating point math

Examples

More

for $k = 1, n - 1$:

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods: basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Complete approximation theory for factorization

Fill-in: index (i, j) where $a_{ij} = 0$ originally, but gets updated to non-zero.

Parallelism and implicit operations: wavefronts, approximation

(and so $\ell_{ij} \neq 0$ or $u_{ij} \neq 0$)

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Parallel collectives

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Fill-in

Change in the sparsity structure! How do you deal with that?

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMV SIMD model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

How does this continue by induction?

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

Observations? Parallelizing, naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

$$\left(\begin{array}{c|cccc} 2 & -1 & 0 & \dots & \\ \hline -1 & 2 & -1 & & \\ 0 & -1 & 2 & -1 & \\ \vdots & \vdots & \vdots & \ddots & \vdots \end{array} \right)$$

$$\left(\begin{array}{c|cccc} 2 & -1 & 0 & \dots & \\ \hline 0 & 2 - \frac{1}{2} & -1 & & \\ 0 & -1 & 2 & -1 & \\ \vdots & \vdots & \vdots & \ddots & \vdots \end{array} \right)$$

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPM/VSIMD model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

$$\left(\begin{array}{cccc|cc} 4 & -1 & 0 & \text{Integers} & -1 & \\ -1 & 4 & -1 & 0 & 0 & -1 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 1 & 0 & \dots & \text{Floating point numbers} & 4 & -1 \\ 0 & -1 & 0 & \dots & -1 & 4 & -1 \end{array} \right)$$

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks, complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

→ Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

LU of a sparse matrix

$$\left(\begin{array}{cccc|cc} 4 & -1 & 0 & \dots & -1 & \\ -1 & 4 & -1 & 0 & 0 & -1 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 1 & 0 & \dots & \text{Computational aspects of iterative methods} & -1/4 & -1 \\ 0 & -1 & 0 & \dots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -1/4 & 4 & -1 & 0 & 4 & -1 \\ 1/4 & -1 & 4 & -1 & -1 & 4 & -1 \end{array} \right)$$

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

→ Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Topical concepts

The SIMD/MIMD/SPMD/MT model for parallelism

Characterization of parallelism by memory model

Interconnection topologies, theoretical models

Programming models

Load balancing, locality, performance curves

First we dig into bits

Integers

pers

math

ples

flore

ition

hms

iods

exity

duct

duct

zing

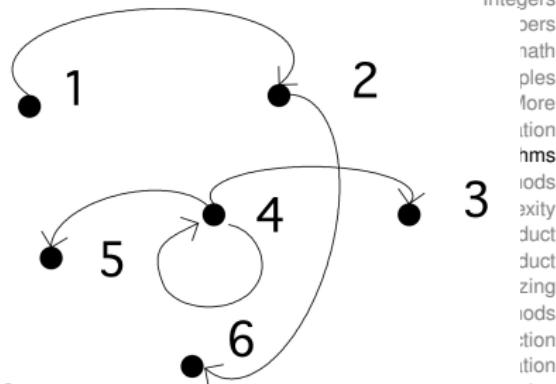
iods

tion

ition

A little graph theory

Graph is a tuple $G = \langle V, E \rangle$ where $V = \{v_1, \dots, v_n\}$ for some n , and $E \subset \{(i, j) : 1 \leq i, j \leq n, i \neq j\}$.



Parallelism and implicit operations, wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicate, Manipulation

Non-blocking collectives

One-sided communication

$$\left\{ \begin{array}{l} V = \{1, 2, 3, 4, 5, 6\} \\ E = \{(1, 2), (2, 6), (4, 3), (4, 4), (4, 5)\} \end{array} \right.$$

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMV/BLAS parallel problem

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Example

More

Graphs and matrices

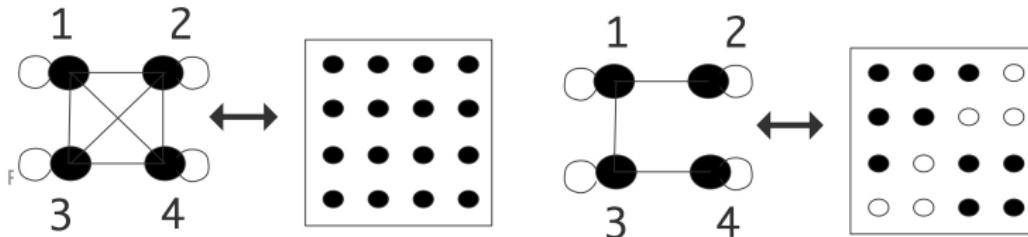
For a graph $G = \langle V, E \rangle$, the adjacency matrix M is defined by

$$M_{ij} = \begin{cases} 1 & (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$$

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods



Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

A dense and a sparse matrix, both with their adjacency graph

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

$$a_{ij} \leftarrow a_{ij} - a_{ik} * a_{kj} / a_{kk}$$

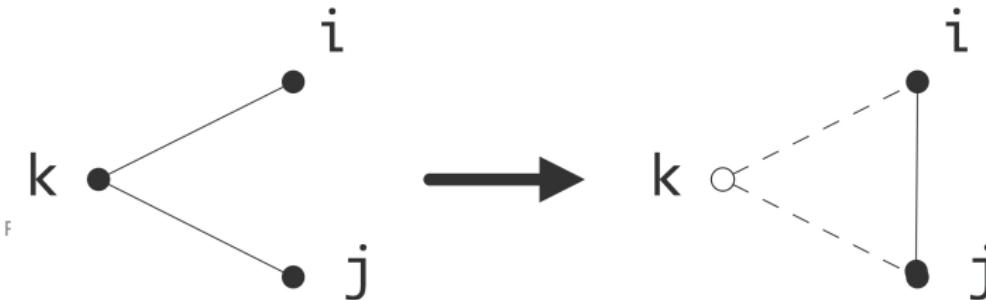
More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Fill-in

Fill-in: index (i, j) where $a_{ij} = 0$ originally, but gets updated to non-zero.



Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Eliminating a vertex introduces a new edge in the quotient graph

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD Model, SIMD vs. SIMD-like, SIMD-like issues

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Original matrix.



Essential aspects of LU factorization
Sparse matrices storage and algorithms

3 Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

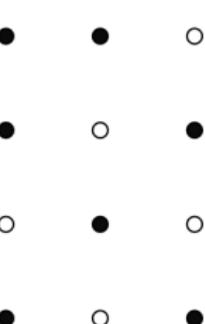
Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.



Structure of a modern processor

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SVD of sparse matrices

Characterization of parallelism by memory model

theoretical concepts

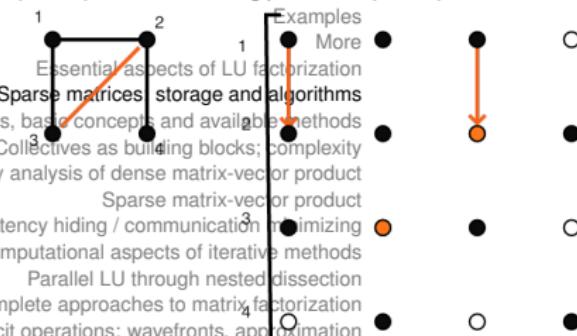
Programming models

First we dig into bits

Integers

Floating point numbers

Eliminating (2, 1) causes fill-in at (2, 3).



Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

use matrix problems

Derived datatypes

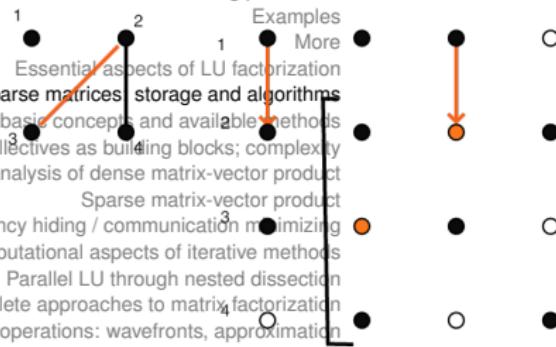
Communicator manipulation

Non-blocking collectives

Profiling and debugging: optimization and programming strategies

- Structure of a modern processor
- Memory hierarchy: caches, register, TLB.
- Multicore issues
- Programming strategies for performance
- The power question
- Basic concepts
- Theoretical concepts
- The SIMD Model: SIMD vs. SIMD-like parallelism
- Characterization of parallelism by memory model
- Interconnects and topologies, theoretical concepts
- Programming models
- Load balancing, locality, space-filling curves
- First we dig into bits
- Integers
- Floating point numbers
- Floating point math
- Examples
- More
- Essential aspects of LU factorization
- 3 Sparse matrices storage and algorithms**
- Iterative methods, basic concepts and available methods
- Collectives as building blocks; complexity
- Scalability analysis of dense matrix-vector product
- Sparse matrix-vector product
- Latency hiding / communication minimizing
- Computational aspects of iterative methods
- Parallel LU through nested dissection
- Incomplete approaches to matrix factorization
- Parallelism and implicit operations: wavefronts, approximation
- Multicore block algorithms
- N-body problems: naive and equivalent formulations
- Graph analytics, interpretation as sparse matrix problems
- Derived datatypes
- Communicator manipulation
- Non-blocking collectives
- One-sided communication
- Profiling and debugging; optimization and programming strategies.

Remaining matrix when step 1 finished.



Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD Model, SIMD vs. SIMD-like, SIMD-like issues

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Eliminating (3,2) fills (3,4)

1

•

2

•

1

•

•

•

Examples

More

Essential aspects of LU factorization

Sparse matrices storage and algorithms

3 Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

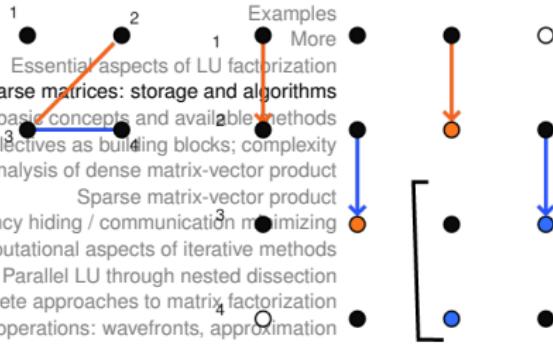
One-sided communication

Profiling and debugging; optimization and programming strategies.



- Structure of a modern processor
- Memory hierarchy: caches, register, TLB.
- Multicore issues
- Programming strategies for performance
 - The power question
 - Basic concepts
 - Theoretical concepts
 - The SIMD Model: SIMD vs. SIMD-like
 - Characterization of parallelism by memory model
 - Interconnects and topologies, theoretical concepts
 - Programming models
 - Load balancing, locality, space-filling curves
 - First we dig into bits
 - Integers
 - Floating point numbers
 - Floating point math
 - Examples
 - More
- Essential aspects of LU factorization
- 3 Sparse matrices: storage and algorithms**
- Iterative methods, basic concepts and available methods
- Collectives as building blocks; complexity
- Scalability analysis of dense matrix-vector product
- Sparse matrix-vector product
- Latency hiding / communication minimizing
- Computational aspects of iterative methods
- Parallel LU through nested dissection
- Incomplete approaches to matrix factorization
- Parallelism and implicit operations: wavefronts, approximation
- Multicore block algorithms
- N-body problems: naive and equivalent formulations
- Graph analytics, interpretation as sparse matrix problems
 - Derived datatypes
 - Communicator manipulation
 - Non-blocking collectives
 - One-sided communication
- Profiling and debugging; optimization and programming strategies.

After step 2



Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

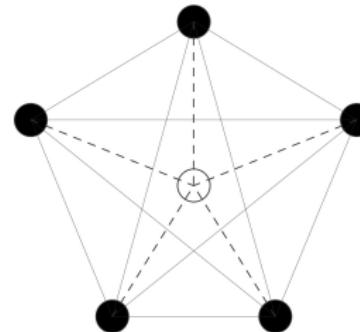
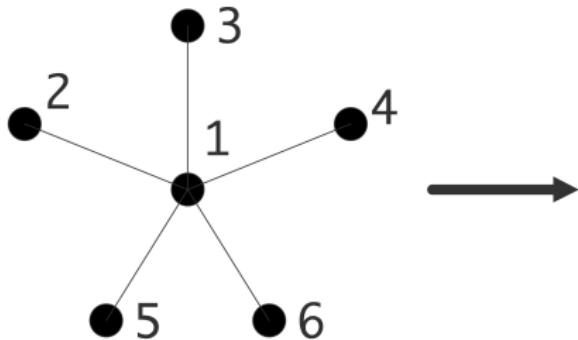
The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SIMD+MIMD model for parallelism

Fill-in is a function of ordering



Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing
Computational aspects of iterative methods

Parallel LU through nested dissection
Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problem

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging: optimization and programming strategies.

After factorization the matrix is dense.
Can this be permuted?

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

Exercise: LU of a penta-diagonal matrix

Consider the matrix

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

-1 Floating point numbers

Floating point math

-1 Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

• Convince yourself that there will be no fill-in. Give an inductive proof of this.

Incomplete approaches to matrix factorization

Parallelism in sparse operations: wavefronts, approximation

Multicore block algorithms

• What does the graph of this matrix look like? (Find a tutorial on

graph theory. What is a name for such a graph?)

Graph analytics, interpretation as sparse matrix problems

• Communicator manipulation

Hotblocking collectives

One-sided communication

• Can you relate this graph to the answer on the question of the fill-in?

Profiling and debugging optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/IMD/SPMD model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

$$a_{ij} = 0 \text{ if } |i - j| > p$$

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collective vs building block complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Can you also derive how much space the inverse will take? (Hint: if $A = LU$, does that give you an easy formula for the inverse?)

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Exercise: LU of a band matrix

Suppose a matrix A is banded with halfbandwidth p :

Floating point numbers

Floating point math

Examples

More

$$a_{ij} = 0 \text{ if } |i - j| > p$$

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collective vs building block complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Can you also derive how much space the inverse will take? (Hint: if $A = LU$, does that give you an easy formula for the inverse?)

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

- Structure of a modern processor

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

- Memory hierarchy: caches, register, TLB.

Load balancing, locality, space-filling curves

- Multicore issues

First we dig into bits

Integers

- Programming strategies for performance

Floating point numbers

Floating point math

Examples

More

- The power question

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Parallelism 85

- Iterative methods: basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

- Basic concepts

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

- Theoretical concepts

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

- The SIMD/MIMD/SPMD/SIMT model for parallelism

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

- Interconnects and topologies; theoretical concepts

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves

Final notes: discrete bits

Iterative methods, basic concepts and available methods

Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
The critical concepts
The SIMD/MIMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Solve $Ax = b$
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More

- **Deterministic**
Essential aspects of LU factorization
- **Exact up to machine precision**
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
- **Expensive (in time and space)**
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods

Iterative methods:
LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
• **Only approximate**
N-body problems, naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
• **Cheaper in space and (possibly) time**
Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

Two different approaches

Stationary iteration

Structure of a modern processor

Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Blas

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
until $\|x^{k+1} - x^k\|_2 < \epsilon$ or until $\frac{\|x^{k+1} - x^k\|_2}{\|x^k\|} < \epsilon$
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Iterative methods

Choose any x_0 and repeat

Sparse matrix-vector product

$$x^{k+1} = Bx^k + c$$

Sparse matrix-vector product
 $\frac{\|x^{k+1} - x^k\|_2}{\|x^k\|} < \epsilon$

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/IMM/SDMM/IMM2D/IMM3D

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

$$\begin{pmatrix} 10 & 0 & 1 \\ 1/2 & 7 & 1 \\ 1 & 0 & 6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 21 \\ 9 \\ 8 \end{pmatrix}$$

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

with solution (2,1,1) (and available methods)

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Communication-aware iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Matrix-block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problem

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and optimization for iterative solvers

$$\begin{pmatrix} 10 & 0 & 1 \\ 7 & 7 & 1 \\ 6 & 0 & 6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 21 \\ 9 \\ 8 \end{pmatrix}$$

might be a good approximation: solution (2,1,9/7,8/6).

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model (four pillars)

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Example system

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

10	0	1
Floating point numbers		
1/2	7	1

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods
with solution (2,1,1)

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Long distance communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

Ordered communication

with solution (2,1,7.95/7,5.9/6).

Profiling and debugging; optimization and programming strategies.

Iterative example'

$$\begin{pmatrix} 10 & 0 & 1 \\ 1/2 & 7 & 1 \\ 1 & 0 & 6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 21 \\ 9 \\ 8 \end{pmatrix}$$

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 21 \\ 9 \\ 8 \end{pmatrix}$$

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 21 \\ 9 \\ 8 \end{pmatrix}$$

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies; theoretical concepts

Programming models

Load balancing; locality; space-filling curves

First we dig into bits

- To solve $Ax = b$, too expensive; suppose $K \approx A$ and solving

$Kx = b$ is possible

Integers

Floating point numbers

Floating point math

Examples

More

- so $Ae_0 = Ax_0 - b = r_0$; this is again unsolvable, so

Sparse matrices: storage and algorithms

- Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

- In one formula:

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Group analysis; interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

$x_{i+1} = x_i - K^{-1}r_i$

One-sided communication

Profiling and debugging; optimization and programming strategies.

Iterative scheme:

where $r_i = Ax_i - b$

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples

More

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods

- **multiplying by A ,**
Scalability analysis of dense matrix-vector product
- **solving with K**
Sparse matrix-vector product
Locality hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems

Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

Takeaway

Each iteration involves:

More

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods

- **multiplying by A ,**
Scalability analysis of dense matrix-vector product
- **solving with K**
Sparse matrix-vector product
Locality hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems

Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

$$r_1 = \underbrace{Ax_1 - b}_{\text{More}} = A(x_0 - \tilde{e}_0) - b \quad (4)$$

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks: complexity

$$(I - AK^{-1})r_0 \quad (5)$$

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

- Inductively: $r_n = (I - AK^{-1})^n r_0$ so $r_n \downarrow 0$ if $|\lambda(I - AK^{-1})| < 1$

Geometric reduction (or amplification!)

Parallelism and implicit operations: wavefronts, approximation

- This is 'stationary iteration': no dependence on the iteration number. Simple analysis, limited applicability.

N-body problems: naive and equivalent formulations

Graphs and hypergraphs: sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Error analysis

- One step

$$r_1 = Ax_1 - b = A(x_0 - \tilde{e}_0) - b \quad (4)$$

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks: complexity

$$(I - AK^{-1})r_0 \quad (5)$$

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

- Inductively: $r_n = (I - AK^{-1})^n r_0$ so $r_n \downarrow 0$ if $|\lambda(I - AK^{-1})| < 1$

Geometric reduction (or amplification!)

Parallelism and implicit operations: wavefronts, approximation

- This is 'stationary iteration': no dependence on the iteration number. Simple analysis, limited applicability.

N-body problems: naive and equivalent formulations

Graphs and hypergraphs: sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Takeaway

The iteration process does not have a pre-determined number of operations.
depends *spectral properties* of the matrix.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD SIMD model of computation
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

- Direct solution is $O(N^3)$
Integers
floating point numbers
Floating point mesh
Examples
More
- Iterative per iteration cost $O(N)$ assuming sparsity.
- Number of iterations is complicated function of spectral properties:
 - Collective as building blocks, complexity
 - Costs of dense matrix-vector product
 - Sparse matrix-vector product
 - Iterative solvers for linear systems
 - Computational aspects of iterative methods
 - Parallelizing iterative methods: domain decomposition
 - Incomplete approaches to matrix factorization
(2nd order only, more for higher order)
- Parallelism and implicit operations: wavefronts, approximation
Multiple block algorithms
- Multigrid and fast solvers: #it = $O(\log N)$ or even $O(1)$
N-body problems: naive and equivalent formulations
- Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
- Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

Fir tree, bin packing

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks: complexity, scalability, analysis of dense matrix-vector product

then Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallel LU with block matrices: naive and equivalent formulations

Multicore block algorithms

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Choice of K

- The closer K is to A , the faster convergence.
- Diagonal and lower triangular choice mentioned above: let

$$A = D_A + L_A + U_A$$

be a splitting into diagonal, lower triangular, upper triangular part,

then Sparse matrix-vector product

Latency hiding / communication minimizing

- Jacobi method: $K = D_A$ (diagonal part),

Parallel LU with block matrices: naive and equivalent formulations

Multicore block algorithms

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

- SOR method: $K = \omega D_A + L_A$

Profiling and debugging; optimization and programming strategies.

If

then

Computationally

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model of parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples

$$A = K - N$$

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix factor product

Sparse matrix-vector product

Latency hiding / communication minimizing

Conjugate gradient, iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

$$Ax = b \Rightarrow Kx = Nx + b \Rightarrow Kx_{i+1} = Nx_i + b$$

Equivalent to the above, and you don't actually need to form the residual.

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
 Memory hierarchy: caches, register, TLB.
 Multicore issues
 Programming strategies for performance
 The power question
 Basic concepts
 Theoretical concepts
 The SIMD/MIMD/SPMD/SIMT model for parallel computation
 Characterization of parallelism by memory model
 Interconnects and topologies, theoretical concepts
 Programming models
 Load balancing, locality, space-filling curves
 First we dig into bits
 Integers
 Floating point numbers
 Floating point math
 $K = D_A$

Algorithm:

for $k = 1, \dots$ until convergence, do:

for $j = 1 \dots n$:

$$a_{ii}x_i^{(k+1)} = \sum_{j \neq i} a_{ij}x_j^{(k)} + b_i \Rightarrow x_i^{(k+1)} = a_{ii}^{-1} \left(\sum_{j \neq i} a_{ij}x_j^{(k)} + b_i \right)$$

Collective building block / implicitly
 Sparse matrix-vector product
 Scalability analysis / dense matrix-vector product
 Latency hiding / communication minimizing
 Computational aspects of iterative methods

Implementation:

for $k = 1, \dots$ until convergence, do:

for $i = 1 \dots n$:

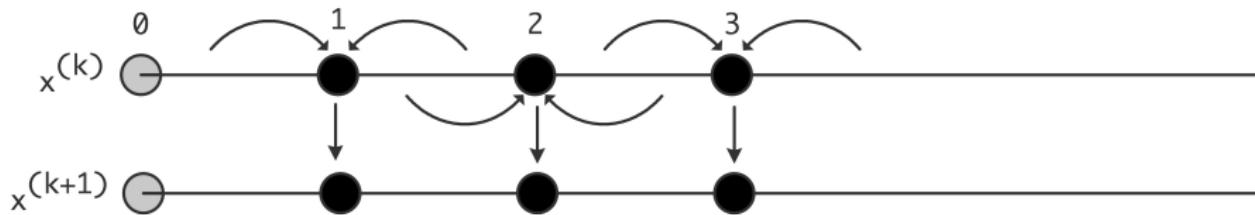
$$t_i = a_{ii}^{-1} \left(\sum_{j \neq i} a_{ij}x_j^{(k)} + b_i \right)$$

Multicore block algorithms
 N-body problems, naive and equivalent formulations
 Graph analytics, interpretation as sparse matrix problems
 Derived datatypes
 Combinator manipulation
 Non-localing collectives

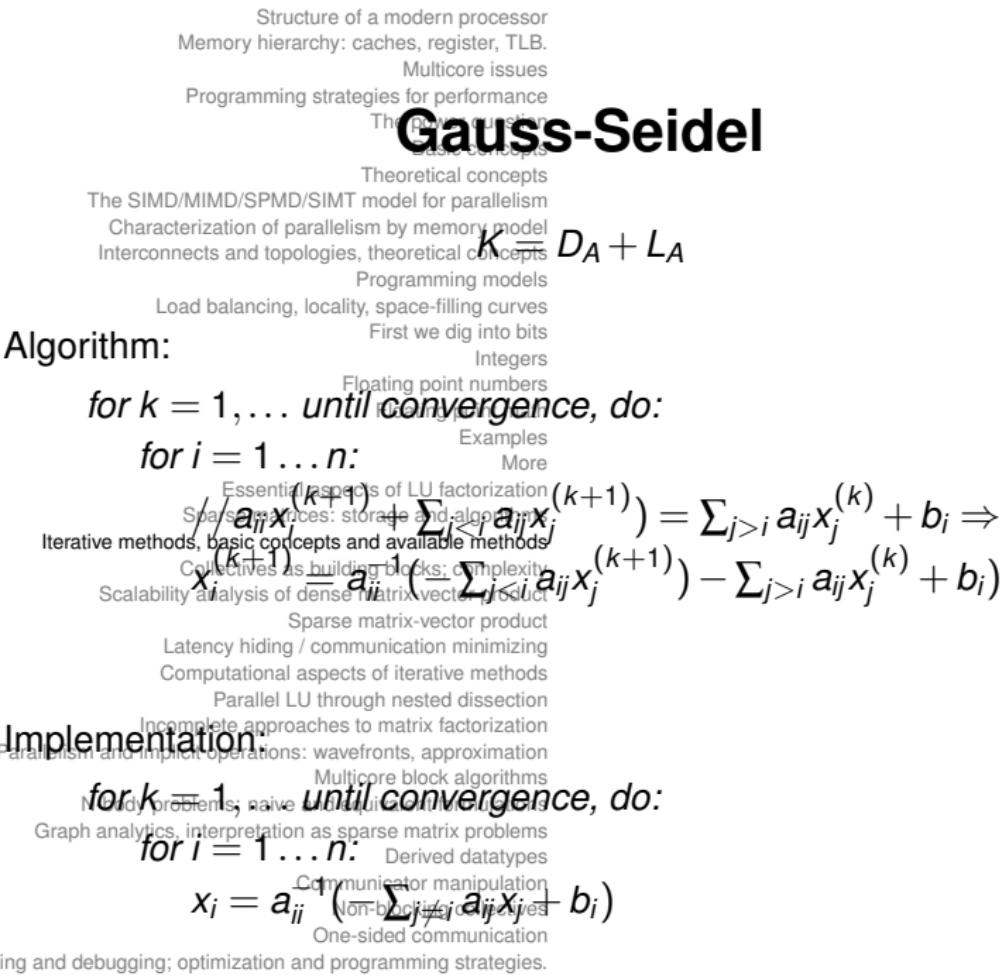
copy $x \leftarrow t$

One-sided communication
 Profiling and debugging, optimization and programming strategies.

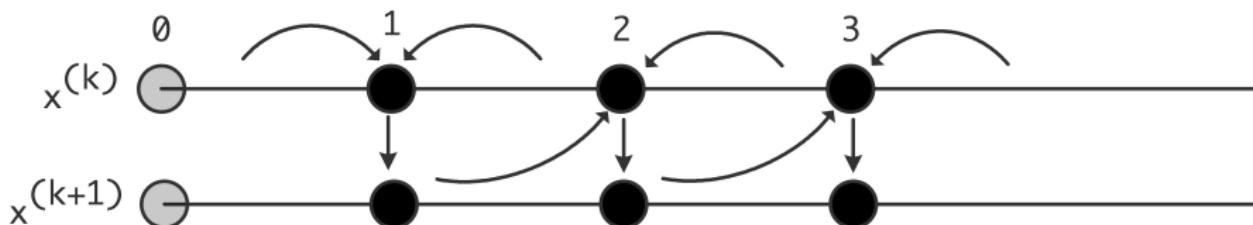
Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT paradigm
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math



Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.



Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math



Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical models

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Characterization of parallelism by communication

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

for k, i, j:

Essential aspects of LU factorization

$a[i, j] = a[i, j] - a[i, k] * a[k, j] / a[k, k]$

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and multi operations: wavefronts, approximation

Multicore block algorithms

IN-local problems: node and row uniform formulations

Graph analytics, interpretation as sparse matrix problems

$a[i, j] = a[i, j] - a[i, k] * a[k, j] / a[k, k]$

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging, optimization and performance strategies

→ sparsity of $L + U$ the same as of A

Choice of K through incomplete LU

- Inspiration from direct methods: let $K = LU \approx A$

Gauss elimination:

for k, i, j:

Essential aspects of LU factorization

$a[i, j] = a[i, j] - a[i, k] * a[k, j] / a[k, k]$

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and multi operations: wavefronts, approximation

Multicore block algorithms

IN-local problems: node and row uniform formulations

Graph analytics, interpretation as sparse matrix problems

$a[i, j] = a[i, j] - a[i, k] * a[k, j] / a[k, k]$

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging, optimization and performance strategies

→ sparsity of $L + U$ the same as of A

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More

Incomplete factorizations mostly work for M-matrices:
2nd order FDM and FEM

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Can be severe headache for higher order

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Applicability

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model of parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

We can't be sure

Integers

Floating point numbers

Floating point math

Examples

- Direct tests on error $e_n = \bar{x} - x_n$ impossible; two choices

- Relative change in the computed solution small:

Based on condition number

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks: complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived databases

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Stopping tests

When to stop converging? Can size of the error be guaranteed?

$$\|x_{n+1} - x_n\| / \|x_n\| < \epsilon$$

- Residual small enough:

$$\|r_n\| = \|Ax_n - b\| < \epsilon$$

Without proof: both imply that the error is less than some other ϵ' .

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Rounding point faults
Examples
More

The serial aspects of LU factorization
Parallelization, storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Polynomial iterative methods

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

Parallelism (SMP/MPI/OpenMP) for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

Processor management

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

then Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

$x = A^{-1}b = \tilde{x} - A^{-1}K\tilde{r}$

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

General form of iterative methods 1.

System $Ax = b$ has the same solution as $K^{-1}Ax = K^{-1}b$.

Let \tilde{x} be a guess and define

$$r = Ax - b, \quad \tilde{r} = K^{-1}r = K^{-1}A\tilde{x} - K^{-1}b$$

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

then Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

$x = A^{-1}b = \tilde{x} - A^{-1}K\tilde{r}$

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

$$x = A^{-1}b = \tilde{x} - A^{-1}K\tilde{r} = \tilde{x} - (K^{-1}A)^{-1}\tilde{r} = \tilde{x} - K^{-1}(AK^{-1})^{-1}\tilde{r}.$$

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/MT model of parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point

Floating point math

Examples

More

Cayley-Hamilton theorem:

$$A \text{ nonsingular} \Rightarrow \exists_{\phi}: \phi(A) = 0.$$

Write

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Apply this to $K^{-1}A$:

$$\phi(x) = 1 + x\pi(x),$$

Stability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Partial LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

so (with previous slide):

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

$$x_{\text{true}} = x_{\text{initial}} + K^{-1}\pi(AK^{-1})r_{\text{initial}}$$

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts

Programming models
Load balancing, locality, space-filling curves
First we dig into bits

$$x_{i+1} = x_0 + K^{-1} \pi^{(i)}(AK^{-1})r_0$$

Integers
floating point number
floating point math

Multiply by A and subtract b : Examples More
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product

So: Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods

$$r_i = \hat{\pi}^{(i)}(AK^{-1})r_0$$

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation
where $\hat{\pi}^{(i)}$ is a polynomial of degree i with $\hat{\pi}^{(i)}(0) = 1$.

N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems

⇒ convergence theory Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

Residuals

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

Implementation of SIMD/SIMT code for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Computational analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

General form of iterative methods 3.

$$x_{i+1} = x_0 + \sum_{j \leq i} K^{-1} r_j \alpha_{ji}.$$

or equivalently:

$$x_{i+1} = x_i + \sum_{j \leq i} K^{-1} r_j \alpha_{ji}.$$

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies; theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

and $\gamma_{i+1,i} = \sum_{j \leq i} \gamma_{ji}$.

Integers

Floating point numbers

Floating point math

Write this as $AK^{-1}R = RH$ where

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

$H = \begin{pmatrix} \gamma_{11} & \gamma_{12} & & \\ \gamma_{21} & \gamma_{22} & & \\ 0 & \gamma_{32} & & \\ 0 & & \ddots & \end{pmatrix}$

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Matrix-free algorithms

H is a Hessenberg matrix, and note zero column sums.

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

No block collective

One-sided communication

$$x_{i+1} \gamma_{i+1,i} = K^{-1} r_i + \sum_{j \leq i} x_j \gamma_{ji}$$

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

Thread-level parallelism (SMT, hyper-threading)

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Efficiency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

General form of iterative methods 5.

$$\begin{cases} r_i = Ax_i - b \\ x_{i+1}, \gamma_{i+1,i} \\ x_{i+1} = K^{-1}r_i + \sum_{j < i} x_j \gamma_{ji} \\ r_{i+1} = A x_{i+1} - b \\ \gamma_{i+1,i+1} = AK^{-1}r_{i+1} + \sum_{j < i+1} r_j \gamma_{ji} \end{cases}$$

where $\gamma_{i+1,i} = \sum_{j \leq i} \gamma_{ji}$.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples

More

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods

- **multiplying by A ,**
Scalability analysis of dense matrix-vector product
- **solving with K**
Sparse matrix-vector product
Locality hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems

Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

Takeaway

Each iteration involves:

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

- **multiplying by A ,**
Scalability analysis of dense matrix-vector product
- **solving with K**
Sparse matrix-vector product
Locality hiding / communication minimizing
Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
The practical concepts
The SIMD/MIMD/SPMD/SIMT model of parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves

Idea one:

If you can make all your residuals orthogonal to each other, and the matrix is of dimension n, then after n iterations you have to have converged: it is not possible to have an n+1-st residual that is orthogonal and nonzero.

Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product

Idea two:

The sequence of residuals spans a series of subspaces of increasing dimension, and by orthogonalizing the initial residual is projected on these spaces. This means that the errors will have decreasing sizes.

N-body problems: naive and equivalent formulations
Graphs and their properties: connectedness, items
Derived datatypes

Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

Orthogonality

Integers
floating point numbers
Floating point math,
arithmetic
More

Computational aspects: LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multidirectional algorithms

N-body problems: naive and equivalent formulations

Graphs and their properties: connectedness, items

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Related concepts:
Integers
Floating point numbers
Floating point math
Examples

- **Positive definite operator** More
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
- **Inner product**
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
- **Projection**
Numerical approaches to matrix factorization
- **Minimization** Multicore block algorithms
 N -body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication

Minimization

$$\forall \mathbf{x} : \mathbf{x}^t \mathbf{A} \mathbf{x} > 0$$

- Structure of a modern processor
 - Memory hierarchy: caches, register, TLB.
 - Multicore issues
 - Programming strategies for performance
 - The power question
 - Basic concepts
 - Theoretical concepts
 - The SIMD/MIMD/SPMD/SIMT model for parallelism
 - Characterization of parallelism by memory model
 - Interconnects and topologies, theoretical concepts
-
- Sparse matrix-vector product
- Latency hiding / communication minimizing
- Computational aspects of iterative methods
- Parallel LU through nested dissection
- Incomplete approaches to matrix factorization
- Parallelism and implicit operations: wavefronts, approximation
- Multicore block algorithms
- N-body problems: naive and equivalent formulations
- Graph analytics, interpretation as sparse matrix problems
 - Derived datatypes
 - Communicator manipulation
 - Non-blocking collectives
 - One-sided communication
- Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SIMD/MIMD model
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models

Let r_0 be given
Space filling curves
First we dig into bits

For $i \geq 0$: Integers

let $s \leftarrow K^{-1}r_i$ Floating point numbers

let $t \leftarrow AK^{-1}r_i$ Floating point math Examples More

for $j \leq i$: Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collective and building block operations

Scalability analysis of dense matrix-vector product

for $j \leq i$: Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

let $x_{i+1} = (\sum_j \gamma_j)$ $s_i, r_{i+1} = (\sum_j \gamma_j)^{-1} t$.

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Full Orthogonalization Method

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD SIMD model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

- Given x, y , can you

First we dig into bits

Integers

Floating point numbers

Floating point math

$x \leftarrow$ something with x, y

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

(What was that called again in your linear algebra class?

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD SIMD model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

- Given x, y , can you

First we dig into bits

Integers

Floating point numbers

Floating point math

$x \leftarrow$ something with x, y

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

(What was that called again in your linear algebra class?

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

- Gramm-Schmid method

Sparse matrix-vector product

Efficiency, scaling, minimization, optimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

How do you orthogonalize?

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD SIMD model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

- Given x, y , can you

First we dig into bits

Integers

Floating point numbers

Floating point math

$x \leftarrow$ something with x, y

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

(What was that called again in your linear algebra class?

Scalability analysis of dense matrix-vector product

- Gramm-Schmid method

Collectives as building blocks; complexity

Sparse matrix-vector product

- Update

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithm

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

$$x \leftarrow x - \frac{x^t y}{y^t y} y$$

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More

Essential aspects of LU factorization
Sparse matrices: storage and algorithms

- **multiplying by A_{i,j}** and available methods
Collectives as building blocks; complexity
- **solving with K**
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
- **inner products**
Other important aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms

N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

Takeaway

Each iteration involves:

- **multiplying by A_{i,j}** and available methods
Collectives as building blocks; complexity
- **solving with K**
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
- **inner products**
Other important aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms

N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Algorithmic tools

Coupled recurrences form

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

- Update iterate with search direction: direction:

Case studies of parallelization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimization

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collective

One-sided communication

Inductively:

$$p_i = K^{-1} r_i + \sum_{j < i} \beta_{ij} K^{-1} r_j,$$

Profiling and debugging; optimization and programming strategies.

(7)

This equation is often split as

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD SIMD, OpenMP, OpenCL

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Basic idea:

$$r_i^T K^{-1} r_j = 0 \quad \text{if } i \neq j.$$

Split recurrences:

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

Residuals and search directions

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

$$\begin{cases} x_{i+1} = x_i - \delta_i p_i \\ r_{i+1} = r_i - \delta_i A p_i \\ p_i = K^{-1} r_i + \sum_{j < i} \gamma_{ij} p_j, \end{cases}$$

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD LVN and SPV model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
More

Three term recurrence is enough:

Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimization
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

$$\begin{cases} x_{i+1} = x_i - \delta_i p_i \\ r_{i+1} = r_i - \delta_i A p_i \\ p_{i+1} = K^{-1} r_{i+1} + \gamma_i p_i \end{cases}$$

Preconditioned Conjugate Gradients

Structure of a modern processor

Memory hierarchy: caches, register, TLB

Processor cores, threads, processes

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies; theoretical concepts

Programming models

Load balancing: locality, space-filling curves

First we digitize bits

solve $Mz^{(i-1)} = r^{(i-1)}$

Integers

Floating point numbers

$\rho_{i-1} = f^{(i-1)^\top} z^{(i-1)}$

Floating point math

if $i = 1$

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector multiplication

$\beta_{i-1} = \rho_{i-1}/\rho_{i-2}$

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational cost of iterative methods

Parallel LU through nested dissection

Incomplete approximations, factorization

Parallelism and implicit operations: wavefronts, approximation

$\alpha_i = p_{i-1}/p_i$ block algorithm

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problem

$x^{(i)} = x^{(i-1)} + \alpha_i p^{(i)}$

Derived datatypes

Communicator manipulation

New blocking collectives

One-sided communication

check convergence; continue if necessary

Profiling and debugging; optimization and programming strategies.

end

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More

Each iteration involves:

- **Matrix-vector product**
Sparse matrix-vector product and algorithms
Iterative methods, basic concepts and available methods
- **Preconditioner solve**
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
- **Two inner products**
Sparse matrix-vector product
Computational aspects of iterative methods
- **Other vector operations.**
Parallel LU through nested dissection
Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

takeaway

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

Three popular iterative methods

The SIMD Model: vector/SIMD multithreading

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

- Conjugate gradients: constant storage and inner products; works

only for symmetric systems

More
Essential aspects of LU factorization
Sparse matrices, storage and algorithms

Iterative methods, basic concepts and available methods

- GMRES (like FOM): growing storage and inner products:
restarting and numerical cleverness

Latency hiding / communication minimizing

Implementation aspects: parallelization

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MATMSP(0) model and generalizations
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves

Special case of SPD:

First we dig into bits
Integers
Floating point numbers
Floating point math
More

For which vector x with $\|x\| = 1$ is $f(x) = 1/2x^t Ax - b^t x$ minimal?

(8)

Taking derivative:

$f'(x) = Ax - b.$

Optimal solution:

$f'(x) = 0 \Rightarrow Ax = b.$

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms
N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems
Derived datatypes

Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

CG derived from minimization

The SIMPLER(SIMD) model and generalizations
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Matrix-vector product: concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms
N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems
Derived datatypes

Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMV/MATMUL parallel algorithm

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

Assume full minimization $\min_{\mathbf{x}} f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^t \mathbf{A} \mathbf{x} - \mathbf{b}^t \mathbf{x}$ too expensive.

with auto-tuning

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

where p_i is search direction

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations (averaging, summation)

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Profile guided optimization

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Minimization by line search

Iterative update

$$\mathbf{x}_{i+1} = \mathbf{x}_i + p_i \delta_i$$

where p_i is search direction

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations (averaging, summation)

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Profile guided optimization

Communicator manipulation

Non-blocking collectives

One-sided communication

$$\delta_i = \operatorname{argmin}_8 \|f(\mathbf{x}_i + p_i \delta_i)\| = \frac{r_i^t p_i}{p_i^t A p_i}$$

Other constants follow from orthogonality.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

More math

Examples

More

Also popular in other contexts:

- **General non-linear systems**

Scalability analysis of iterative methods

Iterative methods, basic concepts and available methods

Parallel LU through nested dissection

Scalability analysis of dense matrix-vector product

- **Machine learning: stochastic gradient descent**

p_i is 'block vector' of training set

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Line search

$p_1^t A p_i$ is a matrix $\Rightarrow (p_1^t A p_i)^{-1} r_i^t p_i$ system solving

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More
Basic I aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

High performance linear algebra

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods: basic concepts and available methods
Collectives as building blocks, complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Justification

Bringing architecture-awareness to linear algebra, we discuss how high performance results from using the right formulation and implementation of algorithms.

- Structure of a modern processor

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

- Memory hierarchy: caches, register, TLB.

Load balancing, locality, space-filling curves

- Multicore issues

First we dig into bits

Integers

- Programming strategies for performance

Floating point numbers

Floating point math

Examples

More

- The power question

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Parallelism 85

- Parallelism

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

- Basic concepts

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

- Theoretical concepts

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

- The SIMD/MIMD/SPMD/SIMT model for parallelism

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Programming models

Processor Architecture 4

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies; theoretical concepts

Programming models

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Collectives as building blocks; complexity

more

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies: theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Basic cases: Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

• **Collect data: gather.**

Parallel LU through nested dissection

Incomplete approximations to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

• **Collect data and compute some overall value (sum, max): reduction.**

N-body problems: naive and equivalent formulations

Graph coloring: interpretation as sparse matrix problems

Derived datatypes

Communication optimization

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Collectives

Gathering and spreading information:

- Every process has data, you want to bring it together;
- One process has data, you want to spread it around.

Root process: the one doing the collecting or disseminating.

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Basic cases: Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

• **Collect data: gather.**

Parallel LU through nested dissection

Incomplete approximations to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

• **Collect data and compute some overall value (sum, max): reduction.**

N-body problems: naive and equivalent formulations

Graph coloring: interpretation as sparse matrix problems

Derived datatypes

Communication optimization

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

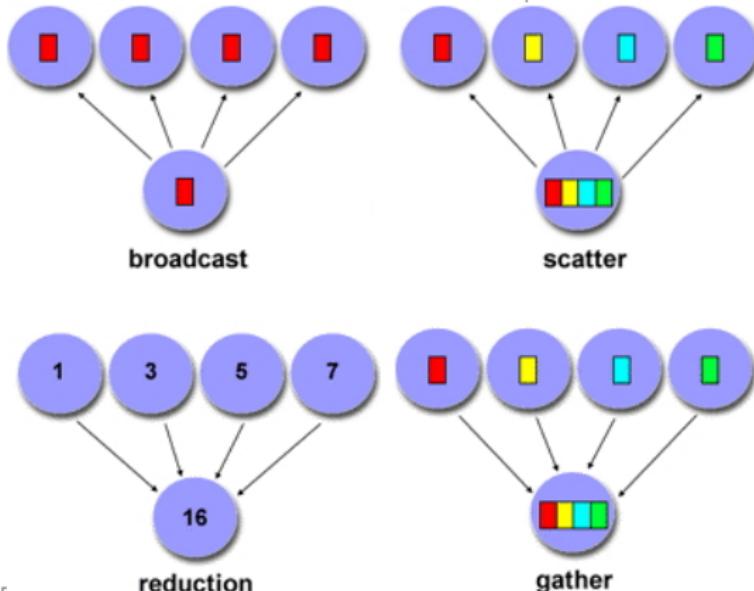
Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts



Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD SIMD model
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Integers
Floating point math
Examples
More

- Let each process compute a random number. You want to print the maximum of these numbers to your screen.
- Each process computes a random number again. Now you want to scale these numbers by their maximum.
- Let each process compute a random number. You want to print on what processor the maximum value is computed.

Parallelism and implicit operations: wavefront approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Collective scenarios

How would you realize the following scenarios with collectives?

Structure of a modern processor

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

Simple model of parallel computation

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

floating point numbers

- α : message latency
First we dig into bits
Integers
 - β : time per word (inverse of bandwidth)
Floating point numbers
Matrix math
Examples
 - γ : time per floating point operation
More

Send n items and do m operations: Collectives as building blocks of complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of **cost** = α

Parallel I.II through nested dissection

Incomplete approaches to matrix factorization

note: no **new** term^s approximation

Pure sends: no yterm.

Multicore block algorithms

pure computation: no α , β terms, Non-parallelisable and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

sometimes mixed: reduction

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD model and parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Load balancing, locality, space-filling curves

- One simultaneous send and receive:

Load balancing, locality, space-filling curves

- doubling of active processors

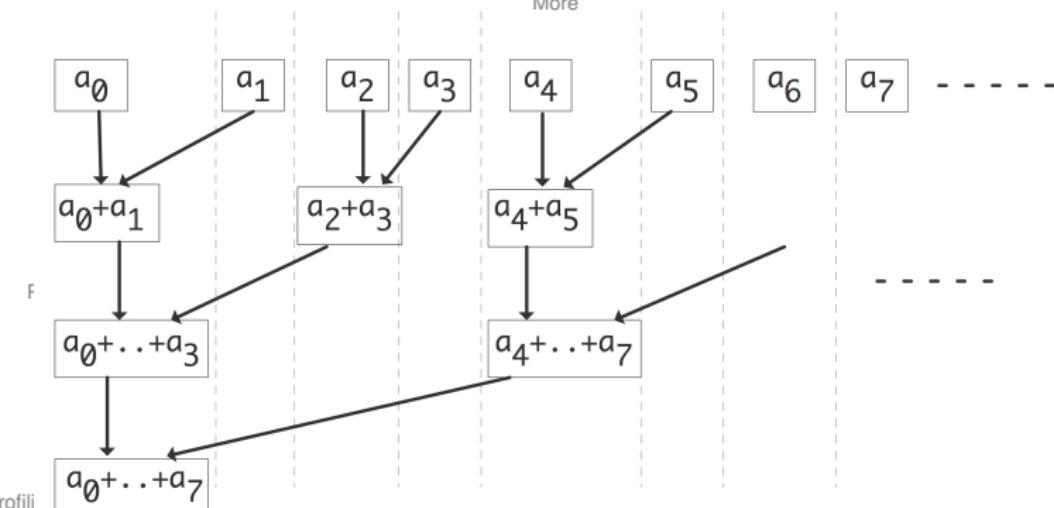
Floating point numbers

Floating point math

Examples

More

- collectives have a $\alpha \log_2 p$ cost component



Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Broadcast

	Load balancing, locality, space-filling curves First we dig into bits	$t = 0$	$t = 1$	$t = 2$
p_0	$x_0 \downarrow, x_1 \downarrow, x_2 \downarrow, x_3 \downarrow$ Integers Floating point numbers	$x_0 \downarrow, x_1 \downarrow, x_2 \downarrow, x_3 \downarrow$	$x_0 \downarrow, x_1 \downarrow, x_2 \downarrow, x_3 \downarrow$	x_0, x_1, x_2, x_3
p_1	Floating point math Example	$x_0 \downarrow, x_1 \downarrow, x_2 \downarrow, x_3 \downarrow$	$x_0 \downarrow, x_1 \downarrow, x_2 \downarrow, x_3 \downarrow$	x_0, x_1, x_2, x_3
p_2	More			x_0, x_1, x_2, x_3
p_3	Essential aspects of LU factorization Sparse matrices: storage and algorithms <u>Iterative methods, basic concepts and available methods</u>			x_0, x_1, x_2, x_3

On $t = 0$, p_0 sends to p_1 ; on $t = 1$ p_0, p_1 send to p_2, p_3 .

Scalability and performance: multi-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximations

Multicore block algorithms

N-body problems: active and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatype

Communicator manipulation

Non-blocking collectives

Send/receive operation

$$\lceil \log_2 p \rceil \alpha + n\beta.$$

$$\lceil \log_2 p \rceil (\alpha + n\beta).$$

Good enough for short vectors.

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SPS model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

	$t = 0$	Floating point numbers Floating point math	$t = 1$	Examples $x_0, x_1 \downarrow, x_2, x_3$	$t = 2$	$t = 3$
p_0	$x_0 \downarrow, x_1, x_2, x_3$				$x_0, x_1, x_2 \downarrow, x_3$	$x_0, x_1, x_2, x_3 \downarrow$
p_1		Essential aspects of LU factorization	x_1			
p_2		Sparse matrices: storage and algorithms			x_2	
p_3		Iterative methods, basic concepts and available methods				x_3
		Collectives as building blocks; complexity				
		Scalability analysis of dense matrix-vector product				
		Sparse matrix-vector product				

Latency hiding / communication minimizing

Computational aspects of iterative methods

takes $p - 1$ messages of size N/p , for a total time of

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: Naive and block-based formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

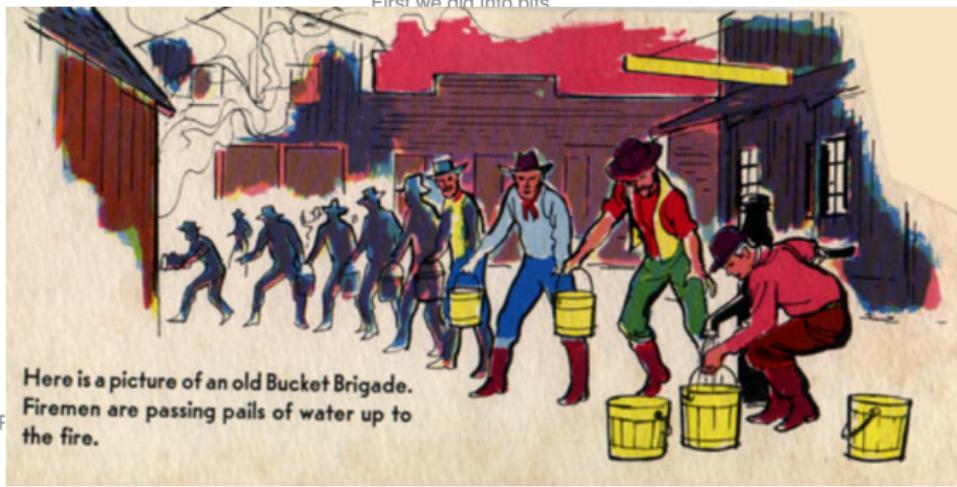
One-sided communication

Profiling and debugging; optimization and programming strategies.

$$T_{\text{scatter}}(N, P) = (p - 1)\alpha + (p - 1) \cdot \frac{N}{p} \cdot \beta.$$

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits.

Bucket brigade



Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SIMD model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

After the scatter do a bucket-allgather:

Load balancing, locality, space-filling curves

First we dig into bits

	$t = 0$	Integers Floating point numbers	$t = 1$		<i>etcetera</i>
p_0	$x_0 \downarrow$	Floating point math Examples	x_0	$x_3 \downarrow$	x_0, x_2, x_3
p_1	$x_1 \downarrow$	More Essential aspects of LU factorization	$x_0 \downarrow, x_1$		x_0, x_1, x_3
p_2	$x_2 \downarrow$	Sparse matrices: storage and algorithms Iterative methods, basic concepts and available methods	$x_1 \downarrow, x_2$		x_0, x_1, x_2
p_3	$x_3 \downarrow$	Collectives as building blocks: complexity Scalability analysis of dense matrix-vector product	$x_2 \downarrow, x_3$		x_1, x_2, x_3

Sparse matrix-vector product

Latency hiding / communication minimizing

Complexity analysis: iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation, sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Better if N large.

Profiling and debugging; optimization and programming strategies.

$$T_{\text{bucket}}(N, P) = (P - 1)\alpha + (P - 1) \cdot \frac{N}{P} \cdot \beta.$$

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Optimal complexity: Programming models

Load balancing, locality, space-filling curves

$$\lceil \log_2 p \rceil \alpha + n\beta + \frac{p-1}{p} \gamma n.$$

First digit omitted
Integers

Floating point numbers

Floating point math

Examples

More

	$t = 1$ Essential aspects of LU factorization Sparse matrices: storage and algorithms	$t = 2$ $x_0^{(0:1)}, x_1^{(0:1)}, x_2^{(0:1)}, x_3^{(0:1)}$	$t = 3$ $x_0^{(0:3)}, x_1^{(0:3)}, x_2^{(0:3)}, x_3^{(0:3)}$
p_0	$x_0^{(0)}, x_1^{(0)}, x_2^{(0)}, x_3^{(0)}$ Collectives as building blocks; complexity	$x_0^{(0:1)}, x_1^{(0:1)}, x_2^{(0:1)}, x_3^{(0:1)}$	
p_1	$x_0^{(1)}, x_1^{(1)}, x_2^{(1)}, x_3^{(1)}$ Scalability analysis / dense matrix vector product	$x_0^{(1)}, x_1^{(1)}, x_2^{(1)}, x_3^{(1)}$ Sparse matrix vector product	
p_2	$x_0^{(2)}, x_1^{(2)}, x_2^{(2)}, x_3^{(2)}$ (2) Latency hiding / communication minimization	$x_0^{(2:3)}, x_1^{(2:3)}, x_2^{(2:3)}, x_3^{(2:3)}$ Computational aspects of iterative methods	
p_3	$x_0^{(3)}, x_1^{(3)}, x_2^{(3)}, x_3^{(3)}$ (3) Parallel (3) through (3) nested dissection	$x_0^{(3)}, x_1^{(3)}, x_2^{(3)}, x_3^{(3)}$ Incomplete approaches to matrix factorization	

Parallelism and implicit operations: wavefronts, approximation

Running time Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

$$\lceil \log_2 p \rceil (\alpha + n\beta + \frac{p-1}{p} \gamma n).$$

Communicator manipulation

Non-blocking collectives

One-sided communication

Good enough for short vectors!

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
 Memory hierarchy: caches, register, TLB.
 Multicore issues
 Programming strategies for performance
 The power question
 Basic concepts
 Theoretical concepts
 The SIMD/MIMD/SPMD/SIMT model for parallelism
 Characterization of parallelism by memory model
 Interconnects and topologies, theoretical concepts
 Programming models
 Load balancing, locality, space-filling curves
 First we dig into bits

Allreduce ≡ Reduce + Broadcast
 Reduces
 Floating point numbers
 Floating point math
 Examples
 More

	$t = 1$	$t = 2$	$t = 3$
Essential aspects of LU factorization	x_0	$x_0x_1 \downarrow$	$x_0x_1x_2x_3$
Sparse matrices: storage and algorithms	p_0	$x_1 \uparrow$	$x_0x_1x_2x_3$
Iterative methods, basic concepts and available methods	p_1	$x_0x_1 \downarrow$	$x_0x_1x_2x_3$
Collectives as building blocks; complexity	p_2	$x_2x_3 \uparrow$	$x_0x_1x_2x_3$
Scalability analysis of dense matrix-vector product	p_3	$x_2x_3 \uparrow$	$x_0x_1x_2x_3$
Sparse matrix-vector product			
Latency hiding / communication minimizing			
Computational aspects of iterative methods			
Parallel LU through nested dissection			
Incomplete approaches to matrix factorization			
Parallelism and implicit operations: wavefronts, approximation			
Multicore block algorithms			
High-level: programming models			
Graph analytics, interpretation as sparse matrix problems			
Derived datatypes			
Communicator manipulation			
Non-blocking collectives			
One-sided communication			

Same running time as regular reduce!
 Graph analytics, interpretation as sparse matrix problems

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
 Memory hierarchy: caches, register, TLB.
 Multicore issues
 Programming strategies for performance
 The power question
 Basic concepts
 Theoretical concepts
 The SIMD/MIMD/SPMD/SIMT model for parallelism
 Characterization of parallelism by memory model
 Interconnects and topologies, theoretical concepts
 Programming models
 Load balancing, locality, space-filling curves
 First we dig into bits

Gather n elements: each processor owns n/p ; optimal running time

Integers
 Floating point numbers
 Floating point math
 Examples
 More
 Essential aspects of LU factorization
 Sparse matrices: storage and algorithms
 Iterative methods, basic concepts and available methods
 Collectives as building blocks; complexity

$$\lceil \log_2 p \rceil \alpha + \frac{p-1}{p} n\beta.$$

	$t=1$	$t=2$	$t=3$
Scalability analysis of dense matrix-vector product	x_0	$x_0 x_1$	$x_0 x_1 x_2 x_3$
Sparse matrix-vector product	p_0	x_0	$x_0 x_1 x_2 x_3$
Latency hiding / communication minimizing	p_1	x_1	$x_0 x_1 x_2 x_3$
Computational aspects of iterative methods	p_2	x_2	$x_0 x_1 x_2 x_3$
Parallel LU through nested dissection	p_3	x_3	$x_0 x_1 x_2 x_3$
Incomplete approaches to matrix factorization			
Parallelism and implicit operations: waypoints, approximation			
Multicore block algorithms			
N-body problems: naive and hierarchical formulation			
Graph analytics, interpretation as sparse matrix problems			

Same time as gather, half of gather-and-broadcast.

Derived datatypes
 Communicator management
 Non-blocking collectives
 One-sided communication
 Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model and parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

$t = 2$

$t = 3$

	$t = 1$	$t = 2$	$t = 3$
p_0	$x_0^{(0)}, x_1^{(0)}, x_2^{(0)} \downarrow, x_3^{(0)} \downarrow$ Essential aspects of LU factorization	$x_0^{(0:2:2)}, x_1^{(0:2:2)} \downarrow$ Sparse matrices: storage and algorithms	$x_0^{(0:3)}$
p_1	$x_0^{(1)}, x_1^{(1)}, x_2^{(1)} \downarrow, x_3^{(1)} \downarrow$ Iterative methods: basic concepts and available methods	$x_0^{(1:3:2)} \uparrow, x_1^{(1:3:2)}$ Collectives as building blocks; complexity	$x_1^{(0:3)}$
p_2	$x_0^{(2)}, x_1^{(2)}, x_2^{(2)} \uparrow, x_3^{(2)}$ Scalability analysis of dense matrix-vector product	$x_2^{(0:2:2)}, x_3^{(0:2:2)} \downarrow$ $x_0^{(1:3:2)} \uparrow, x_1^{(1:3:2)}$ Sparse matrix-vector product	$x_2^{(0:3)}$
p_3	$x_0^{(3)}, x_1^{(3)}, x_2^{(3)}, x_3^{(3)}$ Latency hiding / communication minimizing	Computational aspects of iterative methods	$x_3^{(0:3)}$

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefront computation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

$$[\log_2 p] \alpha + \frac{p-1}{p} n(\beta + \gamma).$$

Table of Contents

Processor Architecture⁴

- Structure of a modern processor
 - The SIMD/MIMD/SPMD/SIMT model for parallelism
 - Characterization of parallelism by memory model
- Memory hierarchy: caches, register, TLB.
- Multicore issues
 - Load balancing, locality, space-filling curves
 - First we dig into bits
 - Integers
- Programming strategies for performance
- The power question
 - Essential aspects of LU factorization
 - Sparse matrices: storage and algorithms
 - Iterative methods: basic concepts and available methods
 - Collectives as building blocks; complexity
 - Scalability analysis of dense matrix-vector product

Parallelism⁸⁵

- Basic concepts
 - Sparse matrix-vector product
 - Latency hiding / communication minimizing
 - Computational aspects of iterative methods
 - Parallel LU through nested dissection
 - Incomplete approaches to matrix factorization
- Theoretical concepts

Parallelism and implicit operations: wavefronts, approximation

- The SIMD/MIMD/SPMD/SIMT model for parallelism
 - Multicore block algorithms
 - N-body problems: naive and equivalent formulations
 - Graph analytics, interpretation as sparse matrix problems
- Characterization of parallelism by memory model
 - Derived datatypes
 - Communicator manipulation
 - Non-blocking collectives
 - One-sided communication
- Interconnects and topologies: theoretical concepts

Profiling and debugging; optimization and programming strategies.

Programming models

- Structure of a modern processor
- Memory hierarchy: caches, register, TLB.
- Multicore issues
- Programming strategies for performance
 - The power question
 - Basic concepts
 - Theoretical concepts
- The SIMD/MIMD/SPMD/SIMT model for parallelism
 - Characterization of parallelism by memory model
 - Interconnects and topologies, theoretical concepts
 - Programming models
 - Load balancing, locality, space-filling curves

Scalability analysis of dense matrix-vector product

- Sparse matrices: storage and algorithms
- Iterative methods, basic concepts and available methods
 - Collectives as building blocks; complexity
- Scalability analysis of dense matrix-vector product**
 - Sparse matrix-vector product
 - Latency hiding / communication minimizing
 - Computational aspects of iterative methods
 - Parallel LU through nested dissection
 - Incomplete approaches to matrix factorization
- Parallelism and implicit operations: wavefronts, approximation
 - Multicore block algorithms
- N-body problems: naive and equivalent formulations
- Graph analytics, interpretation as sparse matrix problems
 - Derived datatypes
 - Communicator manipulation
 - Non-blocking collectives
 - One-sided communication
- Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

ESAC, NLP, SPMV, model of parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

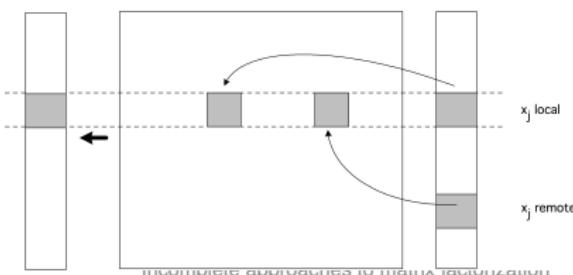
Floating point numbers

Floating point math

Examples

- Assume a division by block rows

- Every processor p has a set of row indices I_p



Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Mvp on processor p :

$$\forall i \in I_p : y_i = \sum_j a_{ij} x_j = \sum_q \sum_{j \in I_q} a_{ij} x_j$$

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/IMM/EPIC/SIMD model of parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

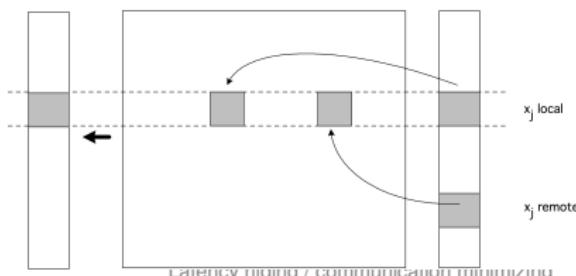
Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Local and remote parts



Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multiple block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

$$\nabla_{\forall i \in I_p} \quad \forall i \quad \sum_{j \in I_p} a_{ij} x_j + \sum_{q \neq p} \sum_{j \in I_q} a_{qj} x_j$$

Local part I_p can be executed right away, I_q requires communication.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/MDSM memory model

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point numbers

Examples

More

Essential aspects of LU factorization

Parallel LU: storage and algorithms

Iterative methods, basic concepts and available methods

1. each process gets a full copy of the input vector (how?)

Scalability analysis of dense matrix-vector product

2. then operates on the whole input

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

How to deal with remote parts

- Very flexible: mix of working on local parts, and receiving remote parts.

- More orchestrated:
Iterative methods, basic concepts and available methods

1. each process gets a full copy of the input vector (how?)

Scalability analysis of dense matrix-vector product

2. then operates on the whole input

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Compare?

(Are we making a big assumption here?)

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model and its variants

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

- **Separate communication and computation:**

Sparse matrices: storage and algorithms

Iterative methods: basic concepts and available methods

Collectives as building blocks; complexity

- **first allgather**

Scalability analysis of dense matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Dense MVP

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD SIMD model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More
Essential aspects of LU factorization

Algorithm:

Step
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product
Allgather x_i so that x is available
on all nodes
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection

Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms

N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Cost (lower bound)

$$\approx 2 \frac{n^2}{P} \gamma$$

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts

Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More
Essential aspects of LU factorization

Assume that data arrives over a binary tree:

- latency $\alpha \log_2 P$
Scalability analysis of dense matrix-vector product
- transmission time, receiving n/P elements from $P - 1$ processors

Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Allgatherers building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Allgatherer

	Structure of a modern processor
	Memory hierarchy: caches, register, TLB.
	Multicore issues
	Programming strategies for performance
	The power question
	Basic concepts
	Theoretical concepts
	The SIMD/MIMD/SPMD/SIMT model for parallelism
	Characterization of parallelism by memory model
	Memory access patterns, programming strategies, theoretical concepts
	Programming models
	Load balancing, locality, space-filling curves
	First we dig into bits
	Integers
	Floating point numbers
	Floating point math
	Examples
	More
Algorithm with cost:	Essential aspects of LU factorization
	Sparse matrices: storage and algorithms
Allgather x_i so that x is available on all nodes	Iteration methods: basic concepts and available methods
	Collectives as building blocks; complexity
Locally compute $y = Ax$	Scalability analysis of dense matrix-vector product
	Sparse matrix-vector product
	Latency hiding / communication minimizing
	Computational aspects of iterative methods
	Parallel LU through nested dissection
	Incomplete approaches to matrix factorization
	Parallelism and implicit operations: wavefronts, approximation
	Multicore block algorithms
	N-body problems: naive and equivalent formulations
	Graph analytics, interpretation as sparse matrix problems
	Derived datatypes
	Communicator manipulation
	Non-blocking collectives
	One-sided communication
	Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

$$S_p^{\text{1D-row}}(n) = \frac{T_1(n)}{T_p^{\text{1D-row}}(n)}$$

Floating point numbers
Floating point math
Examples
More

$$= \frac{2n^2\gamma}{2\frac{n^2}{p}\gamma + \log_2(p)\alpha + n\beta}$$
$$= \frac{p}{1 + \frac{p\log_2(p)}{2n^2} \frac{\alpha}{\gamma} + \frac{p}{2n} \frac{\beta}{\gamma}}$$

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Efficiency: latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU thresholding, column dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

One-sided communication

Strong scaling, weak scaling?

Profiling and debugging; optimization and programming strategies.

Parallel efficiency

Speedup:

$$E_p^{\text{1D-row}}(n) = \frac{s_p^{\text{1D-row}}(n)}{p}$$

Computational aspects of iterative methods
Parallel LU thresholding, column dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
One-sided communication

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Processors fixed, problem grows:

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

$E_p^{\text{1D-row}}(n)$

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

$$1 + \frac{p \log_2(p)}{2n^2} \frac{\alpha}{\gamma} + \frac{p}{2n} \frac{\beta}{\gamma}.$$

Roughly $E_p \approx 1 - n^{-1}$

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

Strong scaling

The SIMD/MIMD/SPMD/SIMT model of parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Problem fixed, $p \rightarrow \infty$

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

$E_p^{1D\text{-row}}(n)$

1

$$1 + \frac{p \log_2(p)}{2n^2} \frac{\alpha}{\gamma} + \frac{p}{2n} \frac{\beta}{\gamma}.$$

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model of parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More

Problem fixed, $p \rightarrow \infty$

$$\frac{1}{E_p(n)} = \frac{1}{1 + \frac{p \log_2(p) \alpha}{2n^2} \frac{\gamma}{\beta} + \frac{p \beta}{2n \gamma}}.$$

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
 $E_p(n)$
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Strong scaling

Roughly $E_p \approx p^1$

Structure of a modern processor
 Memory hierarchy: caches, register, TLB.
 Multicore issues
 Programming strategies for performance
 The power question
 Basic concepts
 The crucial concepts
 The SIMD/MIMD/SPMD/SIMT model
 Characterization of parallelism by memory model
 Interconnects and topologies, theoretical concepts
 Programming models
 Load balancing, locality, space-filling curves
 First we dig into bits
 Integers
 Floating point numbers
 Floating point math
 Examples

Weak scaling
 The SIMD/MIMD/SPMD/SIMT model
 Characterization of parallelism by memory model
 Interconnects and topologies, theoretical concepts
 Programming models
 Load balancing, locality, space-filling curves
 First we dig into bits
 Integers
 Floating point numbers
 Floating point math
 Examples

Memory fixed:
 Essential aspects of LU factorization
 Sparse matrices: storage and algorithms
 Iterative methods: basic concepts and available methods
 Collectives as building blocks: complexity
 $E_p = \frac{1}{1 + \frac{\log_2(p) \alpha}{2n \beta} + \frac{p \beta}{2n \gamma}} = \frac{1}{1 + \frac{\log_2(p) \alpha}{2M \gamma} + \frac{\sqrt{p} \beta}{2\sqrt{M} \gamma}}$
 Scalability analysis of dense matrix-vector product
 Sparse matrix-vector product
 Latency hiding / communication minimizing
 Computational aspects of iterative methods
 Parallel LU through nested dissection
 Incomplete approaches to matrix factorization
 Parallelism and implicit operations: wavefronts, approximation
 Multicore block algorithms
 N-body problems: naive and equivalent formulations
 Graph analytics, interpretation as sparse matrix problems
 Derived datatypes
 Communicator manipulation
 Non-blocking collectives
 One-sided communication
 Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
The crucial concepts

The SIMD/MIMD/SPMD/SIMT model
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples

$$M_{\text{re}} = n^2/p$$

Memory fixed:

$$E_p = \frac{1}{1 + \frac{\log_2(p) \alpha}{2n \beta} + \frac{p \beta}{2n \gamma}} = \frac{1}{1 + \frac{\log_2(p) \alpha}{2M \gamma} + \frac{\sqrt{p} \beta}{2\sqrt{M} \gamma}}$$

1D row (n)
Collectives as building blocks: complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product

Latency hiding / communication minimizing

Communication aspects of iterative methods

Parallel LU through nested dissection

Implementation aspects: cache locality

Does not scale: $E_p \sim 1/\sqrt{p}$
problem in β term: too much communication

Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Weak scaling

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

Two-dimensional partitioning

The SIMD/MIMD/SPOB paradigm and its applications

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

x_0 a_{00} a_{01} a_{02} y_0	x_3 Floating point math a_{03} a_{04} Examples a_{10} a_{11} a_{12} a_{20} a_{21} a_{22} Essential aspects of LU factorization a_{30} a_{31} a_{32} Sparse matrices: storage, algorithms	x_6 a_{06} a_{07} a_{08} a_{16} a_{17} a_{18} a_{26} a_{27} a_{28} y_2 a_{37} a_{37} a_{38}	x_9 a_{09} a_{10} a_{11} a_{19} $a_{1,10}$ $a_{1,11}$ a_{29} $a_{2,10}$ $a_{2,11}$ a_{39} $a_{3,10}$ $a_{3,11}$
Iterative methods, basic concepts and available methods a_{40} a_{41} a_{42} Collectives as building blocks a_{50} a_{51} Scalability analysis of dense matrix-vector product a_{60} a_{61} a_{62} Sparse matrix-vector product a_{70} a_{71} a_{72} Latency hiding / communication minimization	x_4 Iterative methods, basic concepts and available methods a_{43} collectives as building blocks a_{45} communication complexity a_{53} scalability analysis of dense matrix-vector product a_{63} sparse matrix-vector product a_{73} latency hiding / communication minimization	x_7 a_{46} a_{47} a_{48} a_{56} a_{57} a_{58} a_{66} a_{67} a_{68} y_6 a_{77} a_{77} a_{78}	x_{10} a_{49} $a_{4,10}$ $a_{4,11}$ a_{59} $a_{5,10}$ $a_{5,11}$ a_{69} $a_{6,10}$ $a_{6,11}$ a_{79} $a_{7,10}$ $a_{7,11}$
x_5 Computational aspects of iterative methods a_{80} a_{81} a_{82} Parallel LU thresholding near disjoint domain a_{90} a_{91} Incomplete approaches: matrix factorization $a_{10,0}$ $a_{10,1}$ $a_{10,2}$ Parallel and bit operations: weight fronts, approximation $a_{11,0}$ $a_{11,1}$ $a_{11,2}$ Multicore based algorithms	x_6 a_{83} high memory contention a_{93} matrix factorization $a_{10,3}$ weight fronts $a_{11,3}$ multicore based algorithms	x_8 a_{86} a_{87} a_{88} a_{96} a_{97} a_{98} $a_{10,6}$ $a_{10,7}$ $a_{10,8}$ y_{10} $a_{11,7}$ $a_{11,7}$ $a_{11,8}$	x_{11} a_{89} $a_{8,10}$ $a_{8,11}$ a_{99} $a_{9,10}$ $a_{9,11}$ $a_{10,9}$ $a_{10,10}$ $a_{10,11}$ $a_{11,9}$ $a_{11,10}$ $a_{11,11}$

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MMW/SIMD memory partitioning

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Floating point numbers

Floating point math

Processor grid $p = r \times c$, assume $r, c \approx \sqrt{p}$.

Integers

x_0 $a_{00} \quad a_{01} \quad a_{02} \quad y_0$ $a_{10} \quad a_{11} \quad a_{12}$ $a_{20} \quad a_{21} \quad a_{22}$ $a_{30} \quad a_{31} \quad a_{32}$ Iterative methods, basic concepts and available methods	x_3 Examples More Essential aspects of LU factorization Sparse matrices: storage and algorithms Sparse matrix-vector product Latency hiding / communication minimizing Computational aspects of iterative methods	x_6 y_2	x_9 y_3
$x_1 \uparrow$ Collectives as building blocks Scalability analysis of dense matrix-vector product y_4 Sparse matrix-vector product y_5 Latency hiding / communication minimizing Computational aspects of iterative methods	x_4 ; complexity y_5	x_7 y_6	x_{10} y_7
$x_2 \uparrow$ Parallel LU through nested dissection Incomplete approaches to matrix factorization y_8 Parallelism and implicit operations: wavefronts, approximation y_9 N-body problems: naive and equivalent formulations	x_5 y_9	x_8 y_{10}	x_{11} y_{11}

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/VM model of parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More

- Consider block (i,j)
Estimate steps of LU factorization
Sparse matrices: storage and algorithms
Iterative methods: basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Scalability analysis of iterative methods
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Key to the algorithm

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Examples
More

- Collecting x_j on each processor p_{ij} by an *allgather* inside the processor columns.
Essential aspects of LU factorization
Sparse matrix storage and algorithms
Iterative methods, basic concepts and available methods
- Each processor p_{ij} then computes $y_{ij} = A_{ij}x_j$.
- Gathering together the pieces y_{ij} in each processor row to form y_i , distribute this over the processor row: combine to form a *reduce-scatter*.
Collectives as building blocks: complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding: communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
- Setup for the next A or A^T product
Parallelism in collective operations: Multicore algorithms
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Algorithm

Structure of a modern processor
 Memory hierarchy: caches, register, TLB.
 Multicore issues
 Programming strategies for performance
 The power question
 Basic concepts
 Theoretical concepts
 The SIMD/MIMD/SPMD/SIMT model for parallelism
 Characterization of parallelism by memory model
 Interconnects and topologies, theoretical concepts
 Programming models
 Load balancing, locality, space-filling curves
 First we dig into bits
 Integers
 Floating point numbers
 Floating point math
 Examples
 More

Analysis 1.

Step	Cost (lower bound)
Allgather x_i 's within columns	$[\log_2(r)]\alpha + \frac{r-1}{p}n\beta$ $\approx \log_2(r)\alpha + \frac{n}{c}\beta$ $\approx 2\frac{n^2}{p}\gamma$
Perform local matrix-vector multiply	Iterative methods, basic concepts and available methods Collectives as building blocks; complexity Scalability analysis of dense matrix-vector product Sparse matrix-vector product Latency hiding / communication minimizing Computational aspects of iterative methods Parallel LU through blocked dissection Incomplete approaches to matrix factorization
Reduce-scatter y_i 's within rows	Parallelism and implicit operations: wavefronts, approximation Multicore block algorithms N-body problems: naive and equivalent formulations Graph analytics, interpretation as sparse matrix problems Derived datatypes Communicator manipulation Non-blocking collectives One-sided communication Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More

Time:

- Essential aspects of LU factorization
- Sparse matrices: storage and algorithms
- Iterative methods, basic concepts and available methods
- Collectives as building blocks, complexity
- Scalability analysis of dense matrix-vector product**
 - Sparse matrix-vector product
 - Latency hiding / communication minimizing
 - Computational aspects of iterative methods
 - Parallel LU through nested dissection
 - Incomplete approaches to matrix factorization
- Parallelism and implicit operations: wavefronts, approximation
 - Multicore block algorithms
- N-body problems: naive and equivalent formulations
- Graph analytics, interpretation as sparse matrix problems
 - Derived datatypes
 - Communicator manipulation
 - Non-blocking collectives
 - One-sided communication
- Profiling and debugging; optimization and programming strategies.

$$\lceil \log_2 p \rceil \alpha + \frac{p-1}{p} n(\beta + \gamma).$$

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

~~Characterization of parallelism by memory model~~

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space filling curves

Allgather x_i 's within columns

First we dig into bits
Integers

Floating point numbers

Floating point math

Examples

Perform local matrix-vector multiply

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Reduce-scatter y_i 's within rows

Collectives as building blocks, complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Cost (lower bound)

$$\lceil \log_2(r) \rceil \alpha + \frac{r-1}{p} n\beta$$

$$\approx \log_2(r)\alpha + \frac{n}{c}\beta$$

$$\approx 2\frac{n^2}{p}\gamma$$

$$\lceil \log_2(c) \rceil \alpha + \frac{c-1}{p} n\beta + \frac{c-1}{p} m\gamma$$

$$\approx \log_2(c)\alpha + \frac{n}{r}\beta + \frac{n}{r}\gamma$$

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical foundations

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Efficiency

Let $r = c = \sqrt{p}$, then

$E(p \times \sqrt{p})(n) =$

$\Theta\left(\frac{p \log_2(p)}{2n^2} \frac{\alpha}{\gamma} + \frac{\sqrt{p}}{2n} \frac{(2\beta+\gamma)}{\gamma}\right)$

1

$$\frac{1}{1 + \frac{p \log_2(p)}{2n^2} \frac{\alpha}{\gamma} + \frac{\sqrt{p}}{2n} \frac{(2\beta+\gamma)}{\gamma}}$$

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

Strong scaling

The SIMD/MIMD/SPMD/SIMT model of parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Same story as before for $p \rightarrow \infty$

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, direct solvers and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

$$\frac{1}{\frac{p \log_2(p) \alpha}{2n^2} + \frac{\sqrt{p}}{2n} \frac{(2\beta+\gamma)}{\gamma}} \sim p^{-1}$$

No strong scaling

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
The crucial concepts

The SIMD/MIMD/SPMD/SIMT model
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math

Examples

Constant memory $M = n^2/p$: More

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods: basic concepts and available methods
 $E_p^{V^p \times V^p}(n) = \frac{1}{\gamma} \left(\frac{\log(n)}{p} + \frac{\log(p)}{p} + \frac{(2\beta + \gamma)}{\gamma} \right)$

Scalability analysis of dense matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

Weak scaling

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

The crucial concepts

The SIMD/MIMD/SPMD/SIMT model; parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

Constant memory $M = n^2/p$: More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector products

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Weak scaling

$$E_p^{\sqrt{p} \times \sqrt{p}}(n) = \frac{1}{1 + \frac{\log_2(p)}{2M} \frac{\alpha}{\gamma} + \frac{1}{2\sqrt{M}} \frac{(2\beta + \gamma)}{\gamma}} = \frac{1}{1 + \frac{\log_2(p)}{2M} \frac{\alpha}{\gamma} + \frac{1}{2\sqrt{M}} \frac{(2\beta + \gamma)}{\gamma}}$$

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

The crucial concepts

The SIMD/MIMD/SPMD/SIMT model; parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating-point numbers

Floating-point math

Examples

More

Constant memory $M = n^2/p$:

$$E_p^{\sqrt{p} \times \sqrt{p}}(n) = \frac{1}{1 + \frac{p \log_2(p) \alpha + \sqrt{p}}{2n^2} \frac{(2\beta + \gamma)}{\gamma}}$$

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and variants: Jacobi methods
Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product
Weak scaling: hiding / communication minimizing

Computational aspects of iterative methods

for $p \rightarrow \infty$ this is $\approx 1/\log_2 p$:
parallel through tree-sectio
Incomplete approaches to matrix factorization
Parallel sparse direct solvers: pivots, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Weak scaling

$$= \frac{1}{1 + \frac{\log_2(p) \alpha}{2M} + \frac{1}{2\sqrt{M}} \frac{(2\beta + \gamma)}{\gamma}}$$

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

- **Needs a cyclic distribution**

Essential aspects of LU factorization

Optimal matrix-vector product

- **This is very hard to program, so:**

Iterative methods, basic concepts and available methods

- **Scalapack, 1990s product, not extendible, impossible interface**

Scalability analysis of dense matrix-vector product

- **Elemental: 2010s product, extendible, nice user interface (and it is way faster)**

Sparses matrix-vector product

Latency hiding + communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

LU factorizations

Table of Contents

Processor Architecture

- Structure of a modern processor

- The SIMD/MIMD/SPMD/SIMT model for parallelism

- Characterization of parallelism by memory model

- Memory hierarchy: caches, register, TLB.

- Interconnects and topologies; theoretical concepts

- Programming models

- Load balancing, locality, space-filling curves

- Multicore issues

- First we dig into bits

- Integers

- Programming strategies for performance

- Floating point numbers

- Floating point math

- Examples

- More

- The power question

- Essential aspects of LU factorization

- Sparse matrices: storage and algorithms

Parallelism

85

- Iterative methods: basic concepts and available methods

- Collectives as building blocks; complexity

- Scalability analysis of dense matrix-vector product

- Basic concepts

- Sparse matrix-vector product

- Latency hiding / communication minimizing

- Computational aspects of iterative methods

- Theoretical concepts

- Parallel LU through nested dissection

- Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

- The SIMD/MIMD/SPMD/SIMT model for parallelism

- Multicore block algorithms

- N-body problems: naive and equivalent formulations

- Graph analytics, interpretation as sparse matrix problems

- Characterization of parallelism by memory model

- Derived datatypes

- Communicator manipulation

- Non-blocking collectives

- One-sided communication

Profiling and debugging; optimization and programming strategies.

Programming models

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Sparse matrix-vector product

more

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/SIMD+SPMD/SIMD+GEMM paradigm

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

$$\forall i: y_i = \sum_{j \in \text{local}} a_{ij}x_j + \sum_{j \in \text{remote}} a_{ij}x_j$$

Essential aspects of LU factorization

Local part I_p can be executed right away, I_q requires communication.

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability

L

C

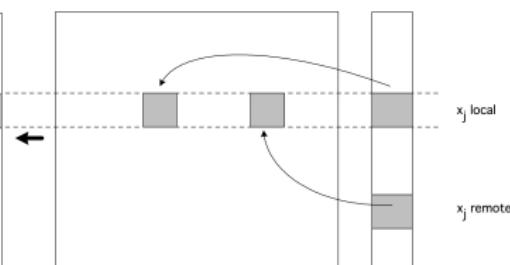
Incc

Parallelism and impl

N-body pr

Graph analytics

Combine:



Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Note possible overlap communication and computation;
only used in the sparse case

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More

- **Traditional: PDE, discussed next**
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
- **New: graph algorithms and big data, discussed later**

Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Sparse matrix operations

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMV/SIMD model for parallelism

Characterization of parallelism by memory model

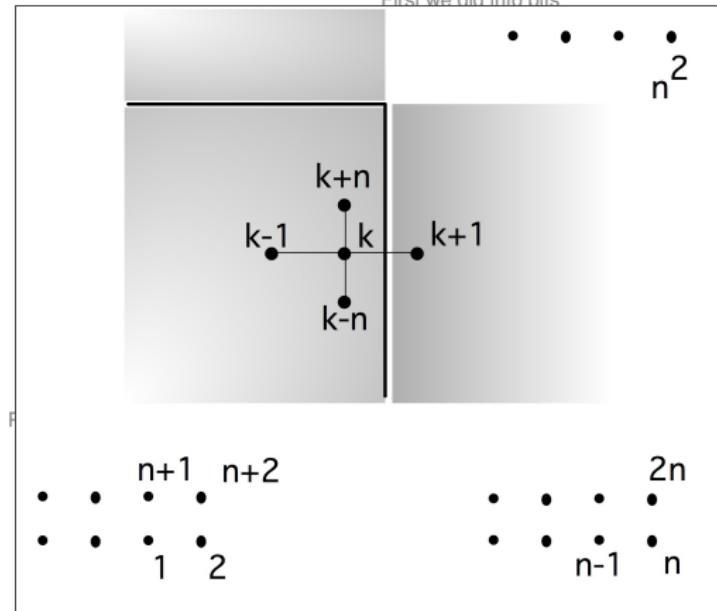
Interconnects and topologies, theoretical concepts

Difference stencil

Programming models

Load balancing, locality, space-filling curves

First we dig into bits



Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPM/Scan model of parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

pects of LU factorization

storage and algorithms

and available methods

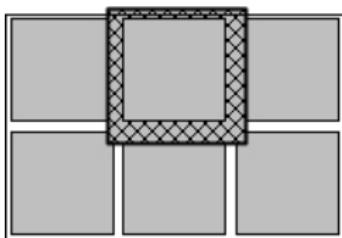
ing blocks; complexity

the matrix-vector product

the matrix-vector product

minimizing

ts of iterative methods



Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multi-tree block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Parallel operator view

induces ghost region:

Limited number of neighbours, limited buffer space

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

- Domain partitioning: processor ‘owns’ variable i

Essential aspects of LU factorization

- owns all connections from j to other js

Sparse matrices: storage and algorithms

Iterative methods; basic concepts and available methods

Collectives as building blocks; complexity

- \Rightarrow processor owns whole matrix row

Sparse matrix-vector product

- \Rightarrow 1D partitioning of the matrix, always

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Matrix vs operator view

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

Essential aspects of LU factorization

Parallel LU: direct methods

- Same phenomenon as with dense matrix:

n^2 variables, memory needed is cn^2/p

- 1D partitioning of domain does not weakly scale

Iterative methods, basic concepts and available methods

– Message size is one line: n

Scalability analysis of dense matrix-vector product

– is $\sqrt{p}\sqrt{M}$, goes up with processors

Latency hiding / communication minimizing

– message size constant in M

Parallel LU through nested dissection

– increasing message size, matrix factorization

– constant in M

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Scaling

- 2D partitioning of domain scales weakly.

Parallelism and implicit operations: wavefronts, approximation

– constant in M

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/MPMD model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

- Assume general communication structure:
neighbour processors can not statically be determined
 - Assume no structural symmetry
 - For matrix-vector product:
each processor issues send and receive requests
 - Problem: receives are easy, sends are hard
 - Inspector-executor: one-time discovery of structure,
followed by many executions
- Parallelism and implicit operations: wavefronts, approximation
N-body problems: Naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

MPI implementation

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD paradigm of parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

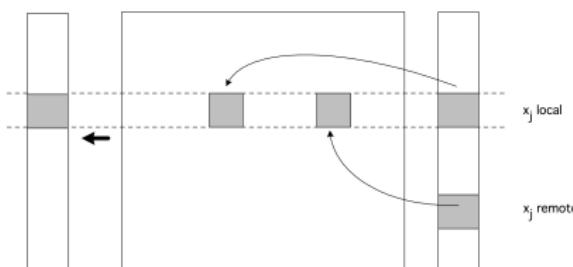
Floating point numbers

Floating point math

Examples

Say

- Processor owns row i , $a_{ij} \neq 0$, processor does not own j



Parallelism and implicit operations: wavefronts, approximation

- Needed: message from j to i

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

N-body problem primitives

One-sided communication

- Processor i can discover this

- Processor j not in general

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves

Make $p \times p$ matrix C :

First we dig into bits
Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Then Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimization

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Failure detection and resilience; error approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Hierarchical and fast multipole methods

Multicore block algorithms

Reduce-scatter, proc i has $C_{i,*}$

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Reduce-scatter

$$C_{ij} = \begin{cases} 1 & i \text{ receives from } j \\ 0 & \text{otherwise} \end{cases}$$

$$s_j = \sum_i C_{ij}$$

number of messages sent by /

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD model of parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More

- The above is collective, implies synchronization
- temp space $O(P)$
Essential aspects of LU factorization
Computations: storage and algorithms
Iterative methods, basic concepts and available methods
- can we get this down to $O(\# \text{neighbours})$?
Parallel nested dissection
Scalability analysis of dense matrix-vector product
- can we detect that we have received all requests
without knowing how many to expect?
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection

Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Reason for more cleverness

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/PIMD SIMD model of parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

- **Barrier: test that every process has reached this point blocking**

Essential aspects of LU factorization

Sparse matrices, storage and algorithms

Iterative methods, basic concepts and available methods

- **Ibarrier: non-blocking test**

Scalability analysis of dense matrix-vector product

Dense matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations via MPI_Ibarrier

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

MPI 3 non-blocking barrier

- **Barrier: test that every process has reached this point blocking**

- **Ibarrier: non-blocking test**

- **Ibarrier calls does not block, yields MPI_Request pointer**

- **Use Wait or Test on the request**

```
int MPI_Ibarrier(MPI_Comm comm, MPI_Request *request)
```

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
The SIMD concept
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
and beyond, including:
First we dig into bits
Integer
Floating point numbers
Floating point math
Examples
More
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods: direct vs iterative methods
Collectives as building blocks; complexity
Sensitivity analysis of dense matrix-vector product
Sparse matrix-vector product
– Loop unrolling, cache blocking, optimizing
Computational aspects of iterative methods
– Incomplete LU factorization
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation

- **Send all your own requests (Isend)**
- **Loop:**
 - **Test on send requests, if all done, enter non-blocking barrier**
 - **Probe for request messages, receive if there is something**
 - **If you're in the barrier, also test for the barrier to complete**
- ~~→ if the barrier completes, you have received all your requests~~

(For safety, use MPI_Issend)
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

DTD algorithms

'Distributed Termination Detection'

used to be (extremely) tricky, now easy with MPI 3

Establish sparse neighbours:

– Essential aspects of LU factorization
Sparse matrices: storage and algorithms

Iterative methods: direct vs iterative methods
Collectives as building blocks; complexity

Sensitivity analysis of dense matrix-vector product

Sparse matrix-vector product

– Loop unrolling, cache blocking, optimizing

Computational aspects of iterative methods

– Incomplete LU factorization
Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

• ~~→ if the barrier completes, you have received all your requests~~

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Table of Contents

Processor Architecture

- Structure of a modern processor

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

- Memory hierarchy: caches, register, TLB.

Interconnects and topologies; theoretical concepts

Programming models

Load balancing, locality, space-filling curves

- Multicore issues

First we dig into bits

Integers

- Programming strategies for performance

Floating point numbers

Floating point math

Examples

More

- The power question

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Parallelism

85

Iterative methods; basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

- Basic concepts

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

- Theoretical concepts

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

- The SIMD/MIMD/SPMD/SIMT model for parallelism

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

- Interconnects and topologies; theoretical concepts

Profiling and debugging; optimization and programming strategies.

Programming models

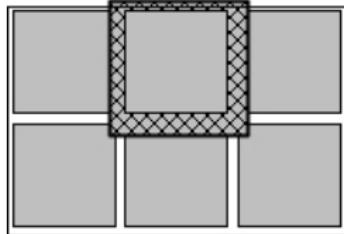
Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Latency hiding / communication minimizing

more
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts

Sparse matrix vector product induces ghost region;



The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
s, theoretical concepts
Programming models
ity, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More

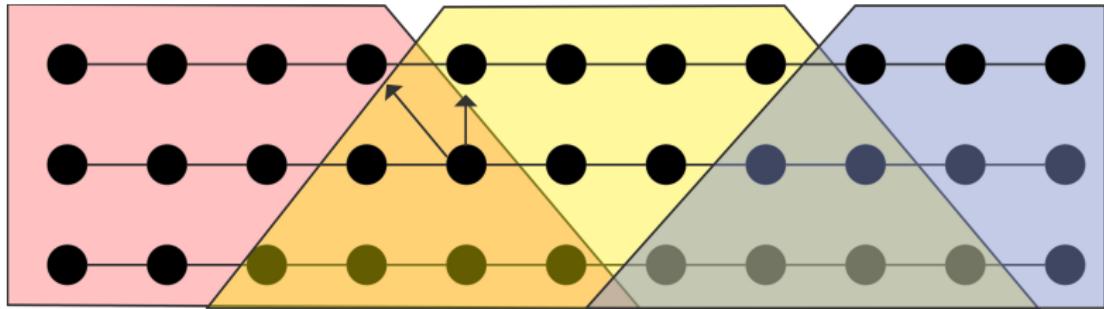
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Parallel matrix-vector product: concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product

- No: surface/volume argument
- Yes: communication is much slower than computation
- Case of multiple products is considerably more interesting

Parallelism and implicit operations: wavefronts, approximation
Multicore clock algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD vs. SPMD parallel model of parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models

Optimization of multiple products



Sparse matrix-vector product
Latency hiding / communication minimizing
Communication-aware matrix-free methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation

- Only one latency

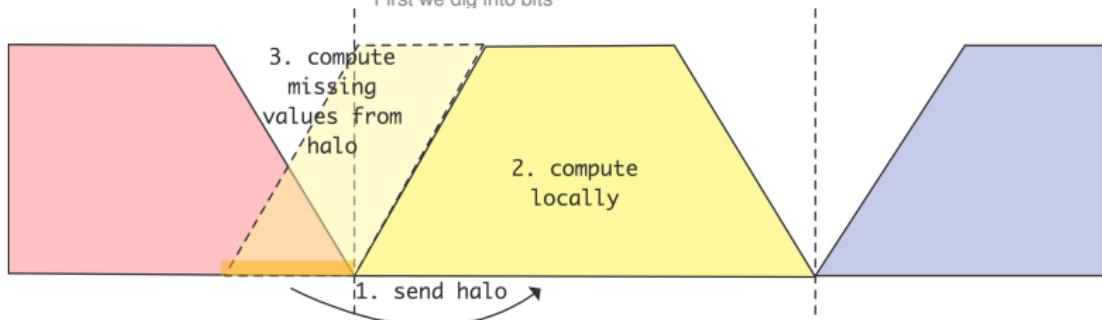
 Multicore block algorithms
 N-body problems: naive and equivalent formulations

- On-node code can be optimized (caching or cache-oblivious)

 Graph analytics interpretation as sparse matrix problems
 Derived datatypes
 Communicator manipulation
 Non-blocking collectives
 One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model and parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits



Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multi-frontal block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Overlap halo transfer with local computation: programming complication

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD approach to parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Parallelizing, communication avoiding

First we dig into bits

Integers

Floating point numbers

point math

Examples

More

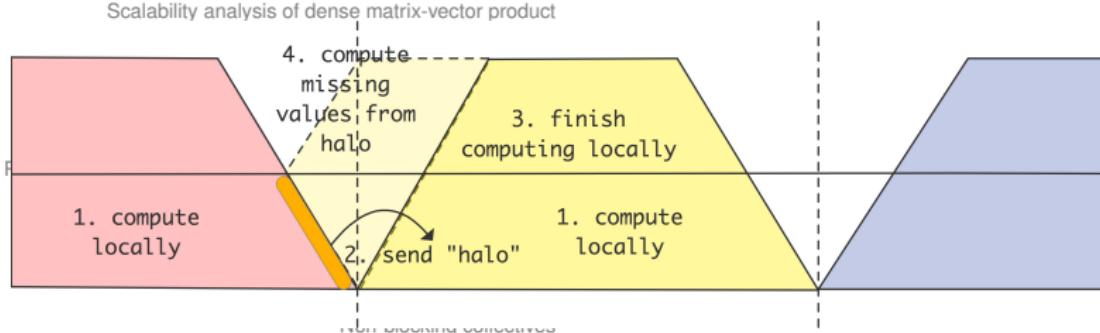
Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product



Profiling and debugging; optimization and programming strategies.

Table of Contents

Processor Architecture⁴

- Structure of a modern processor

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

- Memory hierarchy: caches, register, TLB.

Interconnects and topologies; theoretical concepts

Programming models

Load balancing, locality, space-filling curves

- Multicore issues

First we dig into bits

Integers

- Programming strategies for performance

Floating point numbers

Floating point math

Examples

- The power question

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Parallelism⁸⁵

Iterative methods: basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

- Basic concepts

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

- Theoretical concepts

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

- The SIMD/MIMD/SPMD/SIMT model for parallelism

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

- Interconnects and topologies; theoretical concepts

Profiling and debugging; optimization and programming strategies.

Programming models

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Computational aspects of iterative methods

more

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

matrix-vector product

Preconditioners

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/MD/SD/ST model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

- **Vector updates**

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

- **Inner product**

Dense matrix-vector product

Scalability analysis of dense matrix-vector product

Sparsified matrix-vector product

Latency hiding / communication minimizing

- **Matrix-vector product**

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations; waveform approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

What's in an iterative method?

From easy to hard

Matrix-vector product

Preconditioners

These are trivial

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations; waveform approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/IMMIX/MT model and parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

Collective operation: data from all processes is combined.

(Is a matrix-vector product a collective?)

Essential aspects of LU factorization

Matrix-vector products

Sparse matrix-vector products

Preconditioners

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding; communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Inner products: collectives

Examples: sum-reduction, broadcast

These are each other's mirror image, computationally.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Naive realization of collectives

Broadcast:



Single message:

$\alpha = \text{message startup} \approx 10^{-6} \text{ s}$,
Iterative methods: convergence, available methods

Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods

- Time for message of n words:
Parallel and distributed direct solvers
Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms

N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes

Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

Basic concepts

Theoretical concepts

Floating point math

Examples

Matrix-vector product

Preconditioners

$$\beta = \text{time per word} \approx 10^{-9} \text{ s}$$

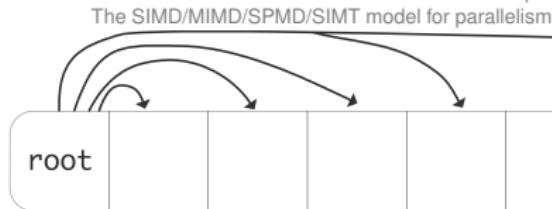
$$\alpha + \beta n$$

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Naive realization of collectives

Broadcast:



Single message:

$\alpha = \text{message startup} \approx 10^{-6} \text{ s}$,
Iterative methods: convergence, parallelizable methods

Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods

- Time for message of n words:
Parallel collective operations
Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms

N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems

- Single inner product: $n=1$
Parallel collective operations
Communicator manipulation

Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

Matrix-vector product
Preconditioners

$$\beta = \text{time per word} \approx 10^{-9} \text{ s}$$

$$\alpha + \beta n$$

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Naive realization of collectives

Broadcast:

The SIMD/MIMD/SPMD/SIMT model for parallelism



Single message:

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

$$\alpha = \text{message startup} \approx 10^{-6} \text{ s}$$

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

$$\bullet \text{ Time for message of } n \text{ words:}$$

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

$$\bullet \text{ Single inner product: } n=1$$

Communicator manipulation

$$\bullet \text{ Time for collective?}$$

Picking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Matrix-vector product

Preconditioners

$$\beta = \text{time per word} \approx 10^{-9} \text{ s}$$

$$\alpha + \beta n$$

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Naive realization of collectives

Broadcast:

The SIMD/MIMD/SPMD/SIMT model for parallelism



Single message:

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

$$\alpha = \text{message startup} \approx 10^{-6} \text{ s}$$

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

$$\bullet \text{ Time for message of } n \text{ words:}$$

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

$$\bullet \text{ Single inner product: } n=1$$

Communicator manipulation

$$\bullet \text{ Time for collective? }$$

Picking collectives

One-sided communication

$$\bullet \text{ Can you improve on that? }$$

Matrix-vector product

Preconditioners

$$\beta = \text{time per word} \approx 10^{-9} \text{ s}$$

$$\alpha + \beta n$$

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

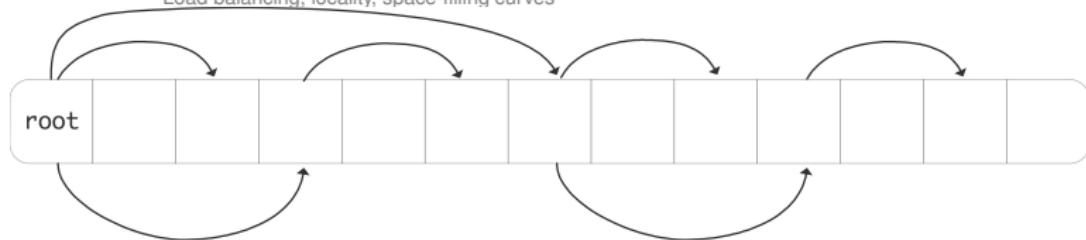
COLLECTIVE/SUPERBLOCKS FOR PARALLELISATION

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves



Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

- **What is the running time now?**

Sparse matrix-vector product

Latency hiding, communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

Better implementation of collectives

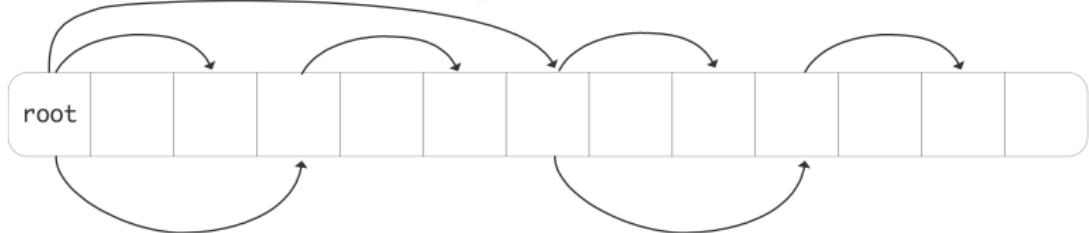
Theoretical concepts
and empirical results

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves



Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

- What is the running time now?
 - Sparse matrix-vector product
 - Latency Hiding - Community detection
 - Can you come up with lower bounds on the α, β terms? Are these achieved here?
 - Computational aspects of iterative methods
 - Parallelization techniques
 - Incomplete approaches to matrix factorization
 - Parallelism and efficiency of low-rank approximation

Parallelism achieved here?

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

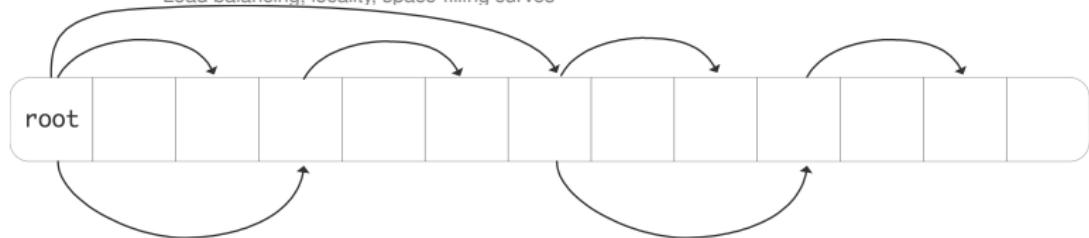
Scalability analysis of dense matrix-vector product

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves



Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding, communication minimizing

Computational aspects of iterative methods

Parallelization of iterative methods

Incomplete approaches to matrix factorization

Parallelism in scientific computing: available time approximation

Multicore block algorithms

- **How about the case of really long buffers?**

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math

- Only operation that intrinsically has a p dependence
 - Examples
 - More
 - Essential aspects of LU factorization
 - Data structures and memory access patterns
 - Iterative methods, basic concepts and available methods
 - Scalability analysis of dense matrix-vector product
 - Parallelizing matrix-vector product
 - Scalability analysis of dense matrix-vector product
 - Research in approaches to hiding: overlapping with other operations
 - Sparse matrix-vector product
 - Latency hiding
 - Non-communication minimizing
 - Computational aspects of iterative methods
 - Parallel LU through nested dissection
 - Incomplete approaches to matrix factorization
 - Parallelism and implicit operations: wavefronts, approximation
 - Multicore block algorithms
 - N-body problems: naive and equivalent formulations
 - Graph analytics, interpretation as sparse matrix problems
 - Derived datatypes
 - Communicator manipulation
 - Non-blocking collectives
 - One-sided communication
 - Profiling and debugging; optimization and programming strategies.

Inner products

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

Vector space, L1/L2/Norms, MD/SMT, Cache parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

- Orthogonality of residuals

Load balancing, locality, space-filling curves

- Basic algorithm: Gram-Schmidt

First we digitize bits
Integers

- one step: given u, v

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods: basic concepts and available methods

then $v \perp u$ Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

- bunch of steps: given U, v

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

then $v \perp U$ Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

$$v' \leftarrow v - \frac{u^t v}{u^t u} u.$$

$$v' \leftarrow v - \frac{U^t v}{U^t U} U.$$

Gram-Schmidt algorithm

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

Modified Gram-Schmidt

The SIMD/MIMD/SIMDSIMT model; parallelism
Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

For $i = 1, \dots, n$: Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic/concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

More numerical stable

Parallelism: nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Matrix-vector product

Preconditioners

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SIMD/Memory parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Let r_0 be given

Space filling curves

First we dig into bits

For $i \geq 0$:

Integers

let $s \leftarrow K^{-1}r_i$

Floating point numbers

Floating point math

let $t \leftarrow AK^{-1}r_i$

Examples

More

for $j \leq i$:

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

let γ_j be the coefficient so that $t - \gamma_j r_j \perp r_j$

Scalability analysis of dense matrix-vector product

for $j \leq i$:

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

let $x_{i+1} = (\sum_j \gamma_j)$, $s_i, r_{i+1} = (\sum_j \gamma_j)^{-1} t$.

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Full Orthogonalization Method

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD model: parallelism and memory hierarchy

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

Let r_0 be given First we dig into bits

Integers

For $i \geq 0$: Floating point numbers

Floating point math

Examples

let $s \leftarrow K^{-1} r_i$ More

Essential aspect of LU factorization

Sparse matrices: storage and algorithms

for $j \leq i$: Sparse matrices: storage and algorithms

Collectives as building blocks: complexity

Scalability analysis for dense matrix-vector product

let γ_j be the coefficient so that $t - \gamma_j r_j \perp r_j$

form $s \leftarrow s - \gamma_j x_j$

Latency hiding / communication minimizing

and $t \leftarrow t - \gamma_j r_j$

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: we follow: approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Modified Gramm-Schmidt

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

Practical differences

The SIMD/MIMD/SPMD/SIMT model differentiation

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

- **Modified GS more stable**

Iterative methods, basic concepts and available methods

Collectives as building blocks: complexity

Scalability analysis of dense matrix-vector product

Matrix-vector product

Preconditioners

- **Inner products are global operations: costly**

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Matrix-vector product

More
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Matrix-vector product

Preconditioners

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies, load balance

The power question

Basic concepts

Theoretical concepts

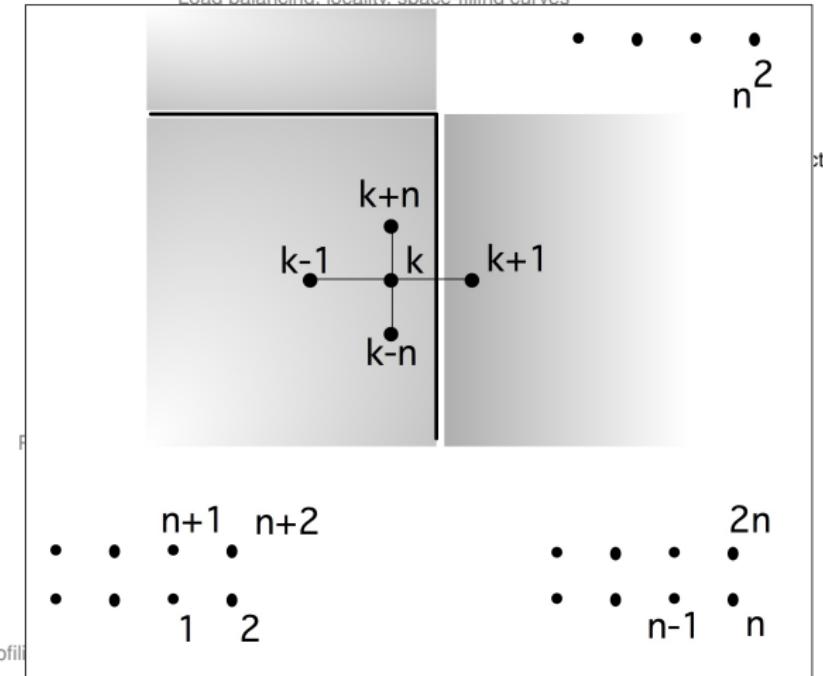
The SIMD/MIMD/SIMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and tasking, theoretical models

Programming Models

Load balancing, locality, space-filling curves



Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model of parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves

Assume each process has the matrix values and vector values in part of the domain.

Integers
Floating point numbers
Floating point math
Examples
More
 $\bar{y} \leftarrow A\bar{x}$
Essential aspects of LU factorization

Matrix-vector product
Preconditioners



$\frac{dy}{dx}$
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
 $-x_{i-1} + 2x_i - x_{i+1}$
Multicore block algorithms

N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model of parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves

Assume each process has the matrix values and vector values in part of the domain.

Integers
Floating point numbers
Floating point math
Examples
More
 $\bar{y} \leftarrow A\bar{x}$
Essential aspects of LU factorization

Matrix-vector product
Preconditioners



Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product



Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
 $-x_{i-1} + 2x_i - x_{i+1}$
Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Processor needs to get values from neighbors.

Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for multicore

The power question

Basic concepts

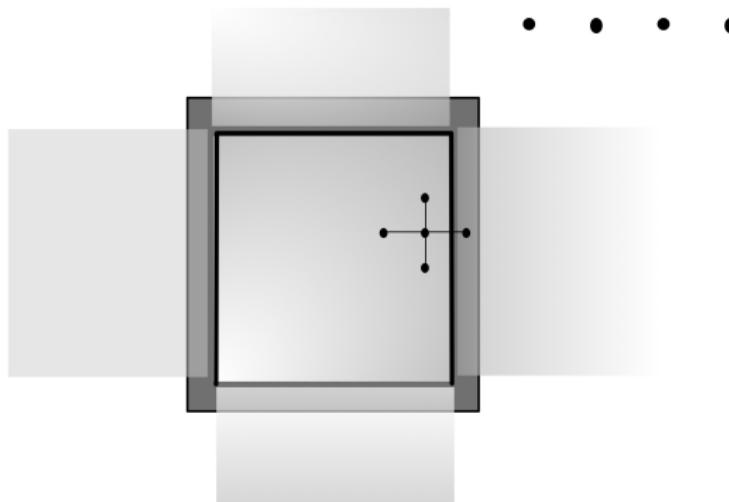
Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Halo region

The 'halo' region of a process, induced by a stencil

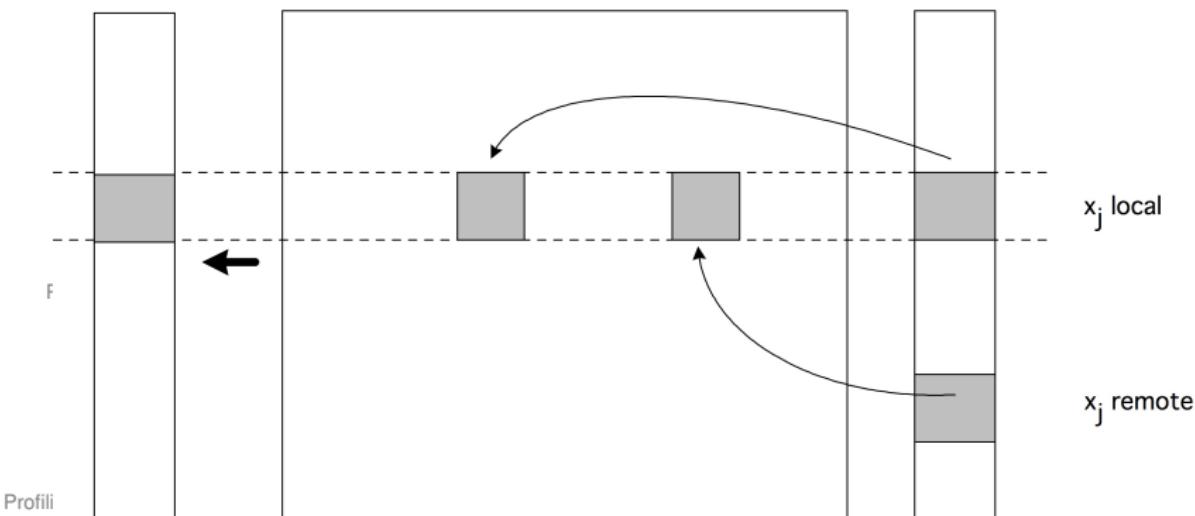
The SIMD/MIMD/SPMD/SIMT model for parallelism



Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Fixed point numbers
Floating point math
Examples

and A, x, y all distributed!

$$y \leftarrow Ax$$



Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

Matrix-vector product performance

Theorem (VMP): $\text{FLOPs} \leq \frac{1}{2} \log_2 n$

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

– partition for scalability

More

– minimize communication (Metis, Zoltan: minimize edge cuts)

Matrix-vector product
Preconditioners

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Beware of optimizations that change the math!

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Integers

Preconditioners

More
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Matrix-vector product

Preconditioners

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model and parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

More

Example: iterative LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks, complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Preconditioners

- There's much that can be said here.

- Some comments to follow

Matrix-vector product

Preconditioners

- There is intrinsic dependence in solvers, hence in

preconditioners,

parallelism is very tricky

approximate inverses

Table of Contents

Processor Architecture⁴

- Structure of a modern processor

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

- Memory hierarchy: caches, register, TLB.

Interconnects and topologies; theoretical concepts

Programming models

Load balancing, locality, space-filling curves

- Multicore issues

First we dig into bits

Integers

- Programming strategies for performance

Floating point numbers

Floating point math

Examples

More

- The power question

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Parallelism⁸⁵

Iterative methods: basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

- Basic concepts

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

- Theoretical concepts

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

- The SIMD/MIMD/SPMD/SIMT model for parallelism

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Programming models

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Parallel LU through nested dissection

more

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model of parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Fill-in: index (i,j) where $a_{ij} = 0$ but $\ell_{ij} \neq 0$ or $u_{ij} \neq 0$.

Floating point numbers

Floating point math

2D BVP: Ω is $n \times n$, gives matrix of size $N = n^2$, with bandwidth n .

Examples
More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Matrix storage $O(N)$: sparse matrix concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

LU storage $O(N^{3/2})$: sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Multiple approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

Cute fact: storage can be computed linear in #nonzeros

Matrix problems and applications

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Fill-in during LU

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIN/MAX SIMD model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks, complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Fill-in is a function of ordering

$$\begin{pmatrix} \dots & * \\ * & 0 \\ \vdots & \ddots \\ * & 0 \end{pmatrix}$$

After factorization the matrix is dense.
Can this be permuted?

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

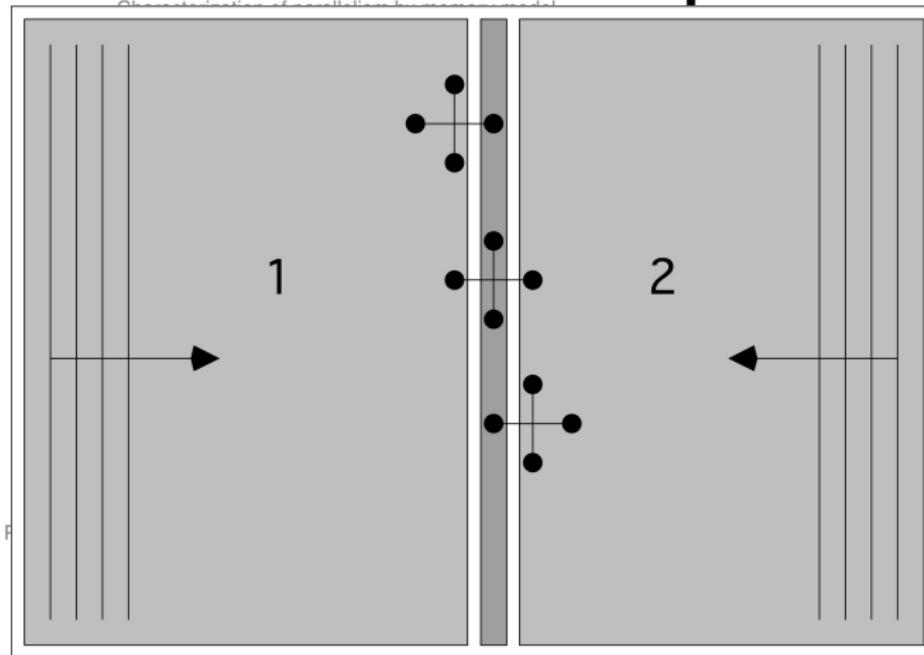
Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

Domain decomposition



Profili

	Structure of a modern processor		
	Memory hierarchy: caches, register, TLB.		
	Multicore issues		
	Programming strategies for performance		
	The power question		
	Basic concepts		
	Theoretical concepts		
★	The SIMD/MIMD/SPMD/SIMT model for parallelism	0	
	Characterization of parallelism by memory model	⋮	
	Interconnects and topologies, theoretical concepts	⋮	
★	★ Programming models	⋮	
	Load balancing, locality, space-filling curves	⋮	
	First we dig into bits	⋮	
	Integers	⋮	
★	★ Floating point numbers	0	
	Floating point math	⋮	
★	★ Examples	⋮	
	More	★	
	Essential aspects of LU factorization	0	
	Sparse matrices: storage and algorithms	⋮	
	Iterative methods, basic concepts and available methods	⋮	
	Collectives as building blocks; complexity	⋮	
	Scalability analysis of dense matrix-vector product	⋮	
0	Sparse matrix-vector product	⋮	
	Latency hiding / communication minimizing	⋮	
	Computational aspects of iterative methods	★	
	Parallel LU through nested dissection	★	
	Incomplete approaches to matrix factorization	★	
Parallelism and implicit operations: wavefronts, approximation	0	0	
	Multicore clock algorithms	★	
N-body problems: naive and equivalent formulations	0	★	
Graph analytics, interpretation as sparse matrix problems	⋮	★	
Derived datatypes	⋮	★	
Communicator manipulation	⋮	★	
Non-blocking collectives	⋮	★	
One-sided communication	⋮	★	
Profiling and debugging; optimization and programming strategies.	⋮	★	

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

DD factorization

The SIMD/MIMD/SPMD/SIMT and their parallelism
Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

$$S = A_{33} - A_{31}A_{11}^{-1}A_{13} - A_{32}A_{22}^{-1}A_{23}$$

Parallelism and implicit operations: wavefronts approximation

Multicore block algorithms

Parallel problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Parallelism...

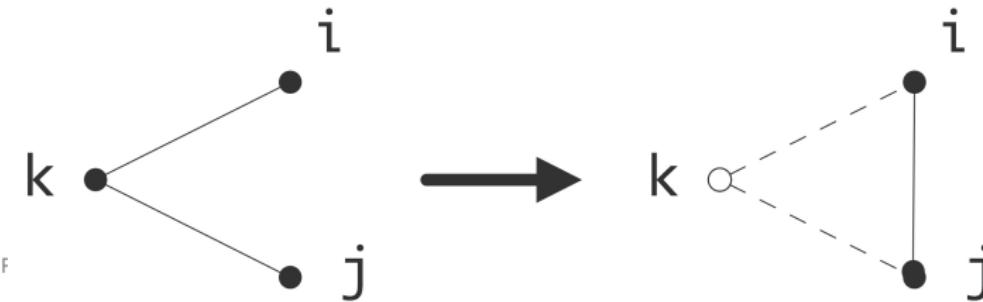
Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The C-MPI-MV-SPM/USP interface for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts

Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Integers
Floating point numbers

$$a_{ij} \leftarrow a_{ij} - a_{ik} a_{kk}^{-1} a_{kj}$$

More



A brief presentation: trees and equivalent formulations

Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

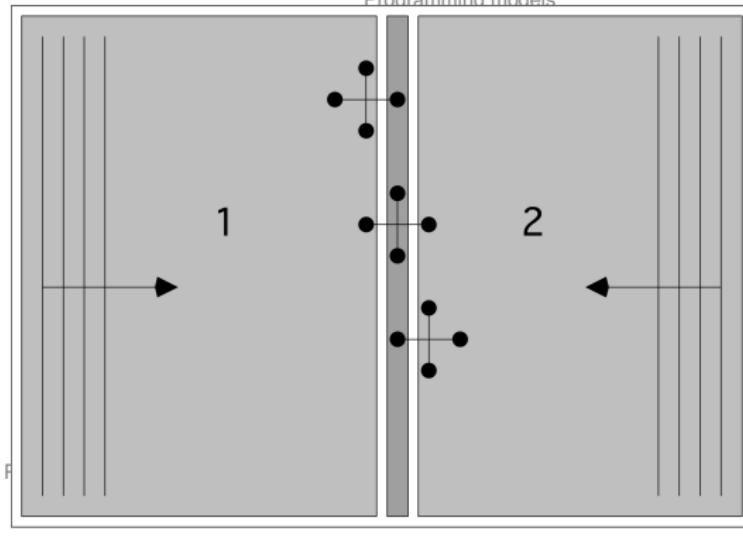
Graph theory of sparse elimination

The C-M-M-M-SPM/USP scheme for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

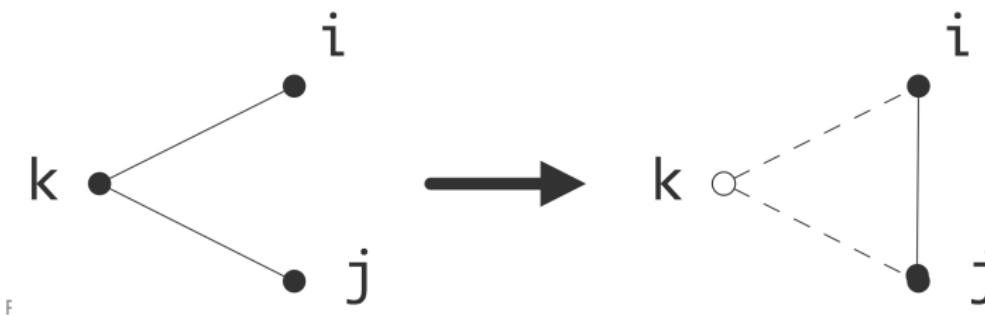


Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The C-M-M-M-SPM/LSI cache for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples



Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics: interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

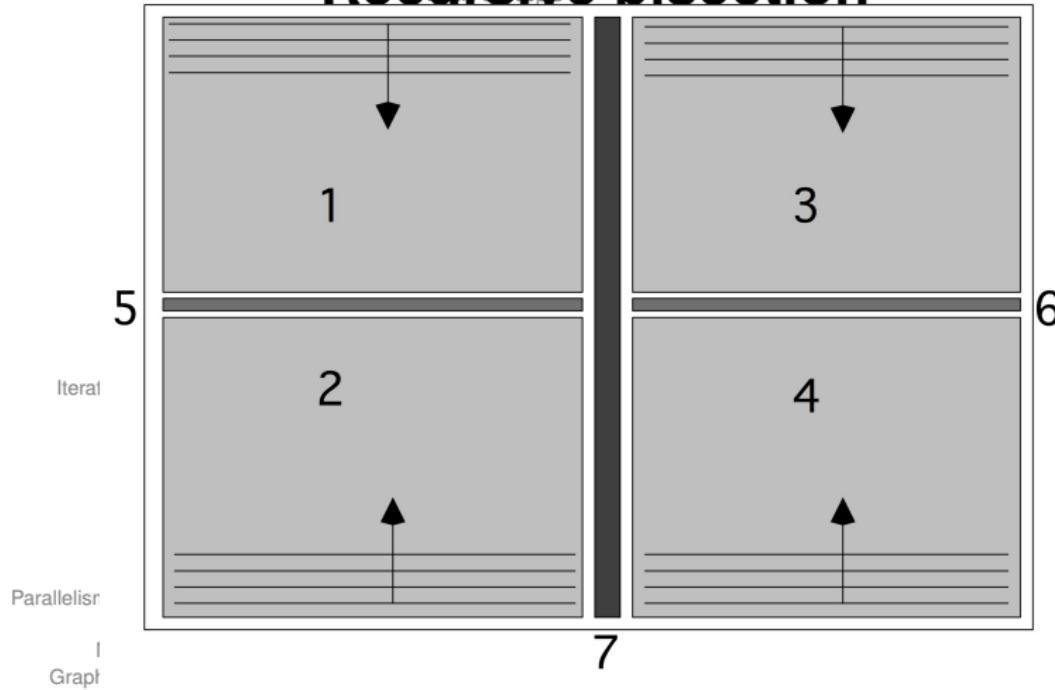
Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SIMT/EMT model and its implications
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More

- This is known as ‘domain decomposition’ or ‘substructuring’
Sparse matrix-vector product with
Iterative methods, basic concepts and available methods
- Separators have better spectral properties
Collectives as building blocks: complexity
Scalability analysis of dense matrix-vector product

Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

More about separators

Recursive bisection



Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication

Figure: A four-way domain decomposition.

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Load balancing, locality, space-filling curves
First word in bits
Integers
Floating point numbers
Floating point math
Examples
More
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks: complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding /communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

$$A^{DD} = \begin{pmatrix} A_{11} & & & & & & \\ & A_{15} & & & & & \\ & & A_{25} & & & & \\ & & & A_{36} & & & \\ & & & & A_{44} & & \\ & & & & & A_{55} & \\ & & & & & & A_{57} \\ A_{51} & A_{52} & & & & & \\ A_{63} & & A_{64} & & & & \\ & & & A_{66} & & & \\ A_{71} & A_{72} & A_{73} & A_{74} & A_{75} & A_{76} & A_{77} \end{pmatrix}$$

The domain/operator/graph view is more insightful, don't you think?

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

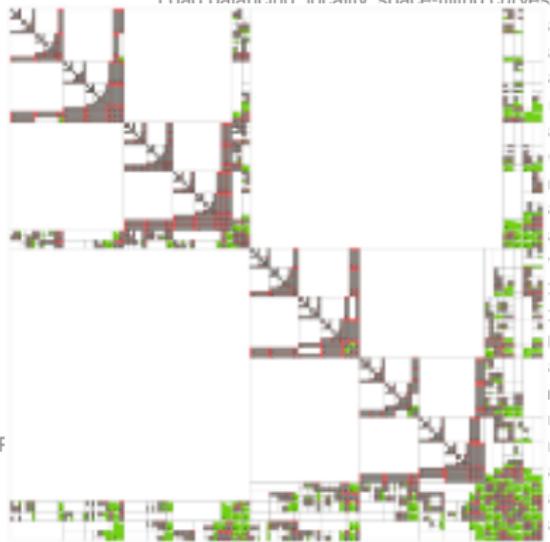
The SIMD/MIMD/SPMD/DAG parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves



Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD and SIMD model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More
Minimum degree, multifrontal,
Implementation of multifrontal factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Finding good separators and domain decompositions is tough in general.
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

More direct factorizations

- Structure of a modern processor

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

- Memory hierarchy: caches, register, TLB.

Load balancing, locality, space-filling curves

- Multicore issues

First we dig into bits

Integers

- Programming strategies for performance

Floating point numbers

Floating point math

Examples

More

- The power question

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Parallelism 85

Iterative methods: basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

- Basic concepts

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

- Theoretical concepts

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

- The SIMD/MIMD/SPMD/SIMT model for parallelism

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Incomplete approaches to matrix factorization

more

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
 Memory hierarchy: caches, register, TLB.
 Multicore issues
 Programming strategies for performance
 The power question
 Basic concepts
 Theoretical concepts
 The LU and Cholesky/SVD factorizations
 Characterization of parallelism by memory model
 Interconnects and topologies, theoretical concepts
 Programming models
 Load balancing, locality, space-filling curves
Mvp $y = Ax$
 First we dig into bits
 Integers
 Floating point numbers
 Floating point math
 Examples
 More
 Essential aspects of LU factorization
 $y[i] = \text{sum over } j=1..n a[i,j]*x[j]$
 Iterative methods, basic concepts and available methods
 Collectives as building blocks; complexity
 Scalability analysis of dense matrix-vector product
In parallel:
 Sparse matrix-vector product
 Latency hiding / communication minimizing
 Computational aspects of iterative methods
 Parallel LU through nested dissection
Incomplete approaches to matrix factorization
 Parallelism and implicit operations: wavefronts, approximation
 $y[i] = \text{sum over } j=1..n a[i,j]*x[j]$
 Multicore block algorithms
 N-body problems: naive and equivalent formulations
 Graph analytics, interpretation as sparse matrix problems
 Derived datatypes
 Communicator manipulation
 Non-blocking collectives
 One-sided communication
 Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
 Memory hierarchy: caches, register, TLB.
 Multicore issues
 Programming strategies for performance
 The power question
 Basic concepts
 Theoretical concepts
How about ILU solve?
 The SIMD/MIMD/SPMD paradigm
 Characterization of parallelism by memory model
 Interconnects and topologies, theoretical concepts
Consider $Lx = y$
 Programming models
 Load balancing, locality, space-filling curves
 First we dig into bits
 Integers
for i=1..n
 Floating point numbers
 Floating point math
 $x[i] = (y[i] - \sum_{j=1..i-1} ell[i,j]*x[j])$
 Essential aspects of LU factorization
 Sparse matrices: storage and algorithms
 Iterative methods, basic concepts and available methods
 Collectives as building blocks; complexity
 Scalability analysis of dense matrix-vector product
Parallel code:
 Sparse matrix-vector product
 Latency hiding / communication minimizing
 Computational aspects of iterative methods
 Parallel LU through nested dissection
 Incomplete approaches to matrix factorization
for i=myfirstrow..mylastrow
 $x[i] = (y[i] - \sum_{j=1..i-1} ell[i,j]*x[j])$
 Multicore block algorithms
 N-body problems: naive and equivalent formulations
 Graph analytics, interpretation of sparse matrix problems
 Derived datatypes
 Communicator manipulation
 Non-blocking collectives
 One-sided communication
Problems?
 Profiling, cache management, optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model of computation

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

LU factorization with iteration

Incomplete approaches to matrix factorization

Block Jacobi with local GS solve

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Block method

for i=myfirstrow..mylastrow

x[i] = (y[i]) - sum over j=myfirstrow..i-1 ell[i,j]*x[j])

/ all, 11

Computational aspects of iterative methods

LU factorization with iteration

Incomplete approaches to matrix factorization

Block Jacobi with local GS solve

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

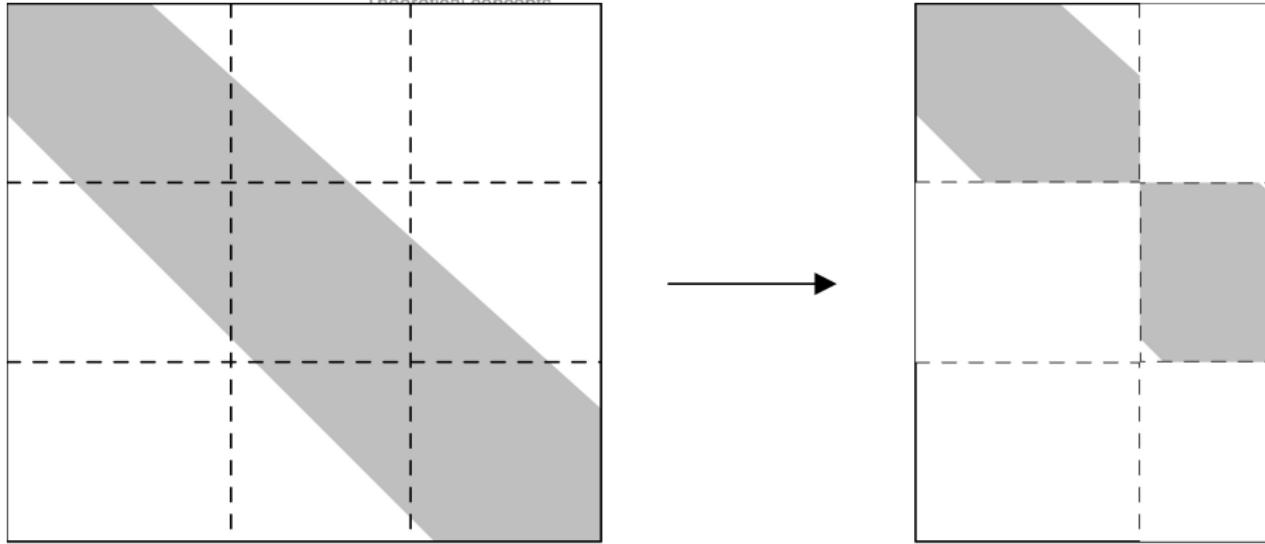
Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts



Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Figure: Sparsity pattern corresponding to a block Jacobi preconditioner.

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model and algorithm

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

a_{11} a_{12} a_{21} a_{22} a_{23} First write into bits

a_{32} a_{33} Integers

a_{31} Floating point numbers

Floating point math

Examples

More

with redblack Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods: basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects a_{55} Iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

a_{21} a_{23} Multicore block algorithm a_{22}

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

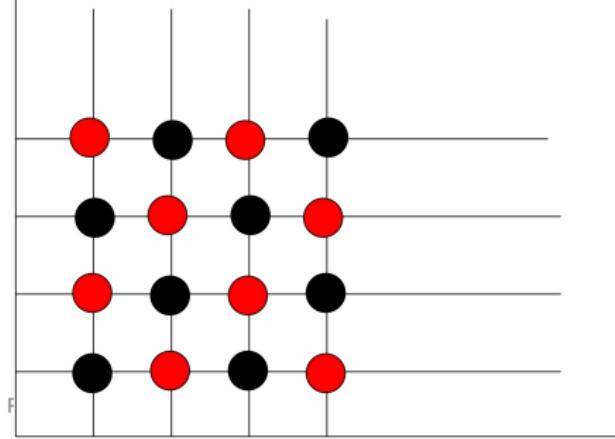
Two processor parallel Gauss-Seidel or ILU
Profiling and debugging, optimization and programming strategies.

$$\begin{pmatrix} 0 \\ a_{11} & a_{12} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \end{pmatrix}$$

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \\ a_{41} & a_{42} \\ a_{34} & a_{32} \\ \vdots & \vdots \\ a_{44} & a_{42} \end{pmatrix} \begin{pmatrix} x_1 \\ x_3 \\ x_5 \\ \vdots \\ x_2 \\ x_4 \\ \vdots \end{pmatrix} = \begin{pmatrix} y_1 \\ y_3 \\ y_5 \\ \vdots \\ y_2 \\ y_4 \\ \vdots \end{pmatrix}$$

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models

2D redblack



N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
In general, colouring, colour number
Balanced distribution
Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

Multicolour ILU

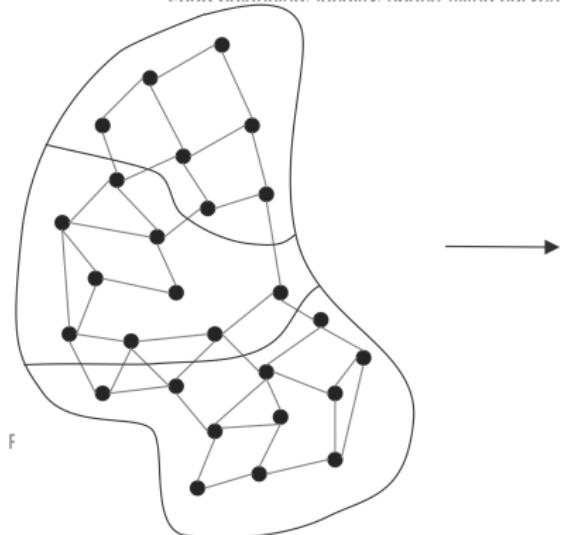
The SIMD/MIMD/SPMD/SIMT model of parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves



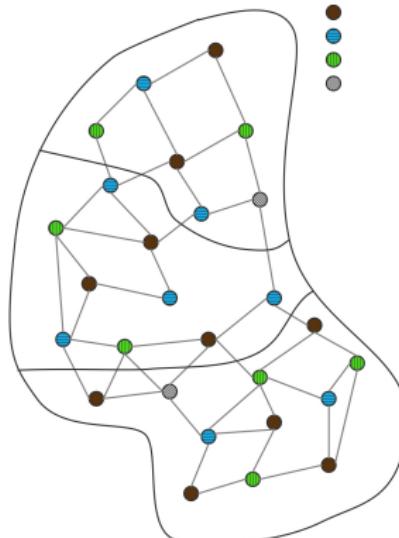
Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.



Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

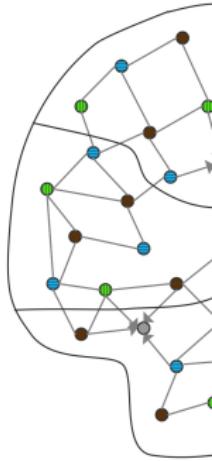
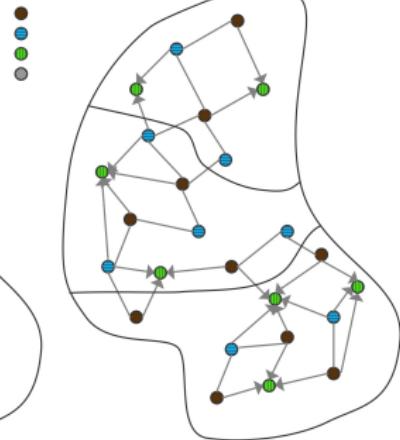
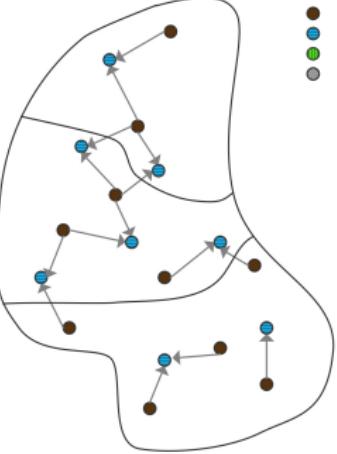
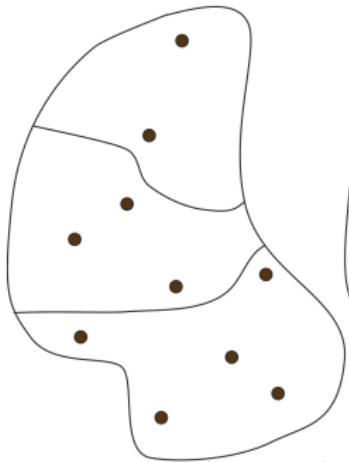
Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism



Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

How do you get a multi-colouring?

THEORY AND DESIGN SIMPLIFIED FOR PARALLELISM

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Exactly colour number is NP-completely: don't bother.

For preconditioner an approximation is good enough:

Luby / Jones - Plassman algorithm

More

Assessing the cost of local computation

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of collective matrix product

Sparse matrix-vector product

Local memory organization, memory reuse

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph-based interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

- Give every node a random value

- First colour: all nodes with a higher value than all their neighbours

- Second colour: higher value than all neighbours except in first

colour

• et cetera

Processor Architecture

Table of Contents

- Structure of a modern processor

- The SIMD/MIMD/SPMD/SIMT model for parallelism

- Characterization of parallelism by memory model

- Memory hierarchy: caches, register, TLB.

- Interconnects and topologies; theoretical concepts

- Programming models

- Load balancing, locality, space-filling curves

- Multicore issues

- First we dig into bits

- Integers

- Programming strategies for performance

- Floating point numbers

- Floating point math

- Examples

- More

- The power question

- Essential aspects of LU factorization

- Sparse matrices: storage and algorithms

Parallelism 85

- Iterative methods: basic concepts and available methods

- Collectives as building blocks; complexity

- Scalability analysis of dense matrix-vector product

- Basic concepts

- Sparse matrix-vector product

- Latency hiding / communication minimizing

- Computational aspects of iterative methods

- Theoretical concepts

- Parallel LU through nested dissection

- Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

- The SIMD/MIMD/SPMD/SIMT model for parallelism

- Multicore block algorithms

- N-body problems: naive and equivalent formulations

- Graph analytics, interpretation as sparse matrix problems

- Characterization of parallelism by memory model

- Derived datatypes

- Communicator manipulation

- Non-blocking collectives

- One-sided communication

Profiling and debugging; optimization and programming strategies.

Programming models

- Structure of a modern processor
- Memory hierarchy: caches, register, TLB.
- Multicore issues
- Programming strategies for performance
 - The power question
 - Basic concepts
 - Theoretical concepts
- The SIMD/MIMD/SPMD/SIMT model for parallelism
 - Characterization of parallelism by memory model
 - Interconnects and topologies, theoretical concepts
 - Programming models
 - Load balancing, locality, space-filling curves

Parallelism and implicit operations: wavefronts, approximation

- Sparse matrices: storage and algorithms
- Iterative methods, basic concepts and available methods
 - Collectives as building blocks; complexity
 - Scalability analysis of dense matrix-vector product
 - Sparse matrix-vector product
 - Latency hiding / communication minimizing
 - Computational aspects of iterative methods
 - Parallel LU through nested dissection
 - Incomplete approaches to matrix factorization
- Parallelism and implicit operations: **wavefronts, approximation**
 - Multicore block algorithms
 - N-body problems: naive and equivalent formulations
 - Graph analytics, interpretation as sparse matrix problems
 - Derived datatypes
 - Communicator manipulation
 - Non-blocking collectives
 - One-sided communication
 - Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

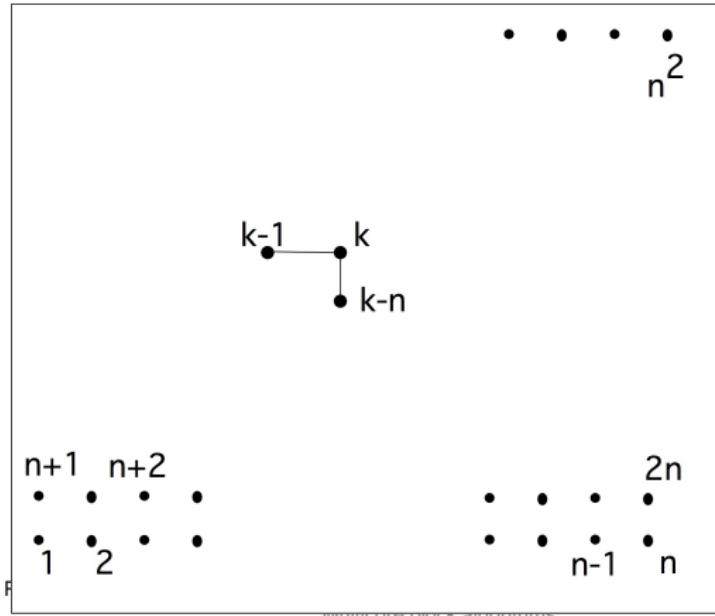
Multicore issues

Programming strategies for performance

The clever question

EPIC, OpenMP

Recurrences



N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

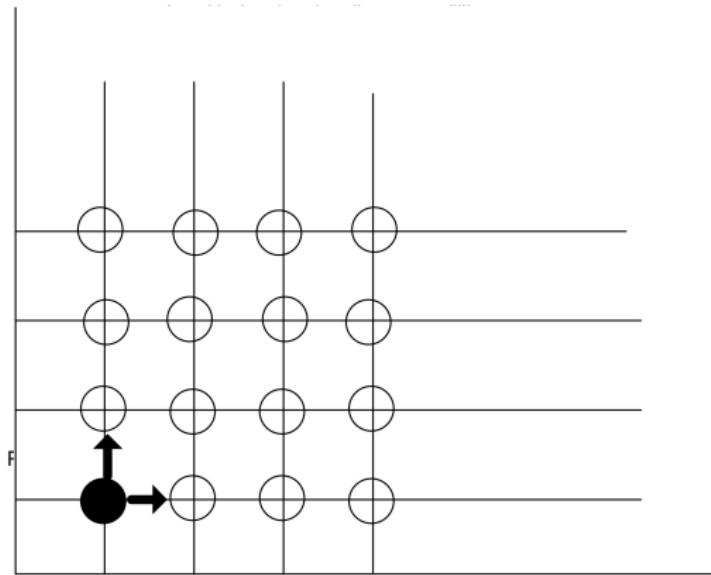
$$x_{i,j} = f(x_{i-1,j}, x_{i,j-1})$$

Profiling and optimizing; optimization and planning strategies.

Intuitively: recursion length n^2

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models

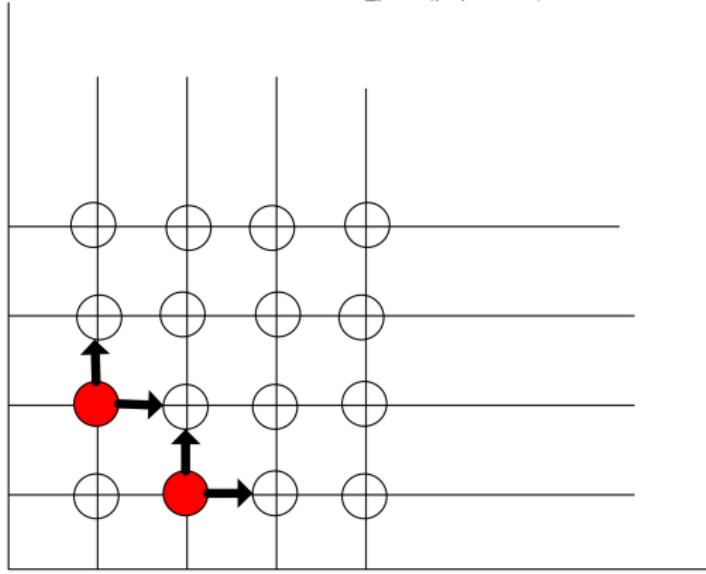
However...



Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

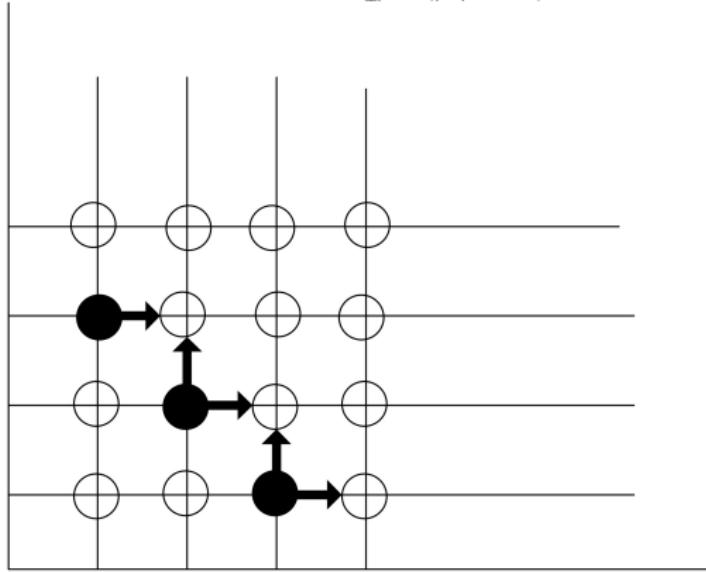
Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts



Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts

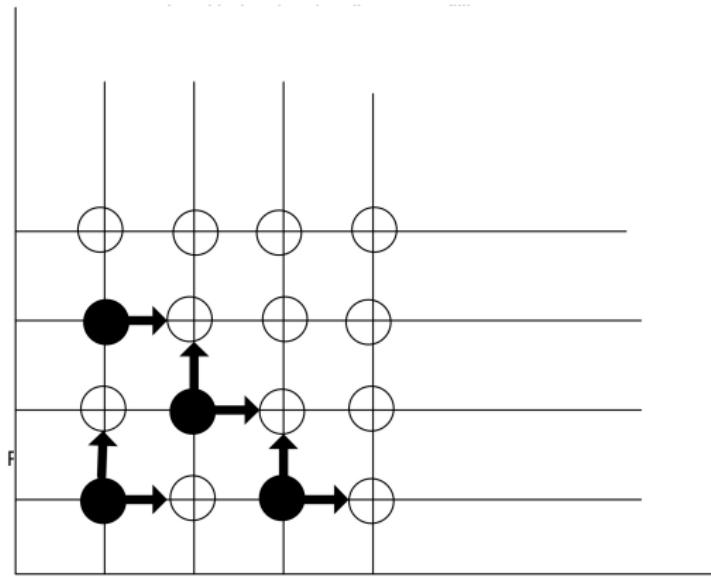


Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models

And in fact

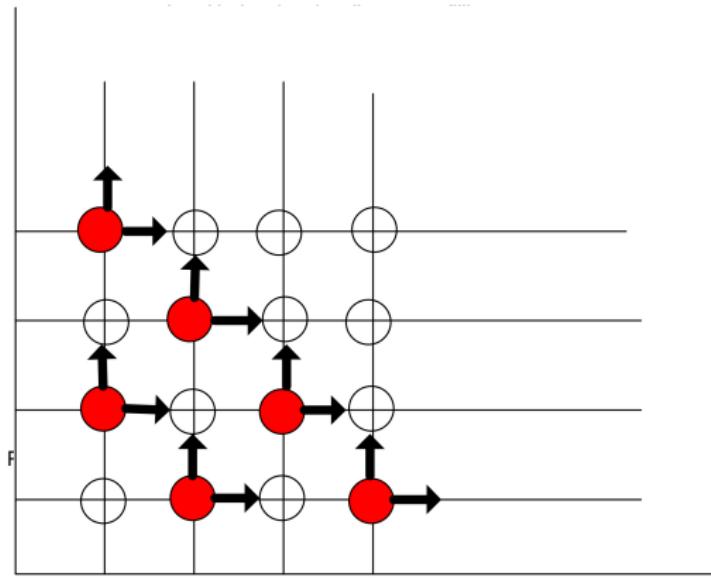


Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models

But then too

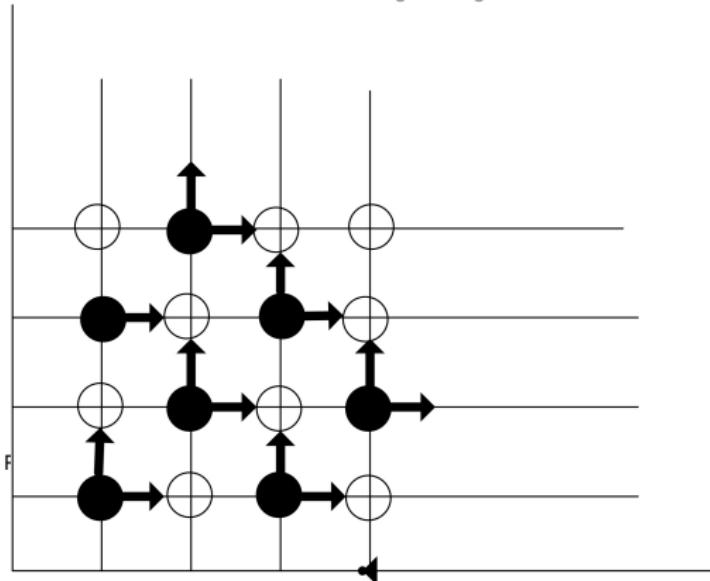


Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models

And



Communication manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More

1. Wavefronts have sequential length $2n$,
average parallelism $n/2$
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks, complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
2. Equivalency of wavefronts and multicolouring
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Conclusion

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT/AF paradigms

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Computing with floating point numbers

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Fast matrix multiplication operations

Wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Recursive doubling

Write recurrence $x_i = b_i + a_{i+1}x_{i+1}$ as

$$\begin{pmatrix} 1 & & & \\ & \ddots & & 0 \\ a_{21} & 1 & & \\ & & \ddots & 0 \\ 0 & & & a_{n,n} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

for short: $A \cdot x = b$

Incomplete approaches to matrix factorization

Fast matrix multiplication operations

Wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Transform

Structure of a modern processor

Multicore issues

Programming strategies for performance

The power question

Basic concepts

The SIMD/MIMD/SPMD/SIMD_{m32} model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts

Programming models

7. space-filling curves

First we dig into bits

Floating point numbers

Floating point math

Examples

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

we methods, basic concepts and available methods

Collectives as building blocks: complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism Read the first half the elements, approximation

- Parallel calculation of other half
 - N-body problems: naive and equivalent formulations
 - Now recurse
 - Graph interpretation as sparse matrix problems

Graph analytics: interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

$$\begin{pmatrix} 0 \\ \vdots \\ -a_{76} & 1 & \vdots & \vdots \end{pmatrix} \times (I + B) =$$

$$\begin{array}{cccc} & 1 & & \\ a_{65} & & 1 & \\ -a_{76}a_{65} & 0 & & 1 \\ & \vdots & & \vdots \\ & \vdots & & \vdots \end{array}$$

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SIMD/SIMT model parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods: basic concepts and available methods

Collectives as building blocks, complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Conjugate gradient, biconjugate gradient

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Turning implicit operations into explicit

Normalize ILU solve to $(I - L)$ and $(I - U)$

Approximate $(I - L)x = y$ by $x \approx (I + L + L^2)y$

Convergence guaranteed for diagonally dominant

Table of Contents

Processor Architecture

- Structure of a modern processor
 - The SIMD/MIMD/SPMD/SIMT model for parallelism
 - Characterization of parallelism by memory model

- Memory hierarchy: caches, register, TLB.
 - Interconnects and topologies; theoretical concepts
 - Programming models

- Multicore issues
 - Load balancing, locality, space-filling curves
 - First we dig into bits

- Programming strategies for performance
 - Floating point numbers
 - Floating point math
 - Examples
 - More

- The power question
 - Essential aspects of LU factorization
 - Sparse matrices: storage and algorithms

- Parallelism 85
 - Iterative methods: basic concepts and available methods
 - Collectives as building blocks; complexity
 - Scalability analysis of dense matrix-vector product

- Basic concepts
 - Sparse matrix-vector product
 - Latency hiding / communication minimizing
 - Computational aspects of iterative methods

- Theoretical concepts
 - Parallel LU through nested dissection
 - Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

- The SIMD/MIMD/SPMD/SIMT model for parallelism
 - Multicore block algorithms

N-body problems: naive and equivalent formulations

- Characterization of parallelism by memory model
 - Graph analytics, interpretation as sparse matrix problems
 - Derived datatypes

Communicator manipulation

- Interconnects and topologies; theoretical concepts
 - Non-blocking collectives
 - One-sided communication

Profiling and debugging; optimization and programming strategies.

Programming models

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Multicore block algorithms

more

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

Cholesky algorithm

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

$$\text{Chol} \begin{pmatrix} A_{11} & A_{21} \\ A_{21} & A_{22} \end{pmatrix} = LL^t \quad \text{where} \quad L = \begin{pmatrix} L_{11} & 0 \\ \tilde{A}_{21} & \text{Chol}(A_{22} - \tilde{A}_{21}\tilde{A}_{21}^t) \end{pmatrix}$$

and where $\tilde{A}_{21} = A_{21}L_{11}^{-t}$, $A_{11} = L_{11}L_{11}^t$.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

for $k = 1, \text{nblocks}$:

Chol: factor $L_k L_k^t \leftarrow A_{>k,k} A_{>k,k}^t$
Essential aspects of LU factorization
Matrix storage strategies

Iterative methods, basic concepts and available methods

Trsm: solve $A_{>k,k} \leftarrow A_{>k,k} L_k^{-1} A_{>k,k}^t$
Previous building blocks: transpose

Gemm: form the product $A_{>k,k} A_{>k,k}^t$
Scalability analysis of dense matrix-vector products
Opposite of the transpose

Syrk: symmetric rank-k update $A_{>k,>k} \leftarrow A_{>k,>k} - \tilde{A}_{>k,k} \tilde{A}_{>k,k}^t$
Latency hiding / communication minimizing
Implementation of the rank-k update

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Implementation

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/IMPP/SIMT paradigm

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

finished

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

$A_{k+1,k}$ Collective building blocks complexity

Scalability analysis of dense matrix-vector product

$A_{k+2,k}$ Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

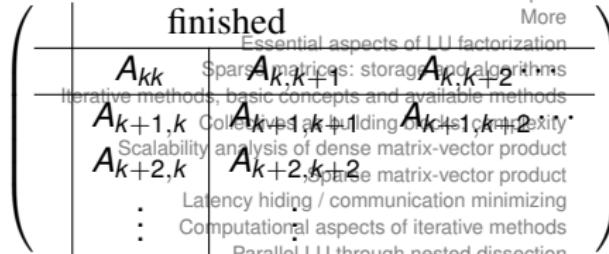
Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Blocked implementation



Extra level of inner loops:

for $k = 1, \text{nblocks}$:

Chol: factor $L_k L_k^t \leftarrow A_{kk}$

for $\ell > k$:

Trsm: solve $\tilde{A}_{\ell,k} \leftarrow A_{\ell,k} L_k^{-t}$

for $\ell_1, \ell_2 > k$:

Gemm: form the product $\tilde{A}_{\ell_1,k} \tilde{A}_{\ell_2,k}^t$

for $\ell_1, \ell_2 > k, \ell_1 \leq \ell_2$:

Syrk: symmetric rank-k update

$$A_{\ell_1,\ell_2} \leftarrow A_{\ell_1,\ell_2} - \tilde{A}_{\ell_1,k} \tilde{A}_{\ell_2,k}^t$$

Structure of a modern processor

Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

You can graph this

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model



Profil: Umwelt- und Naturschutz, Kommunikation, Marketing, PR, Werbung, Medien, Film, Kultur, Tourismus.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

Sometimes...

The SIMD/MIMD/SPMD/SIMT model (Fourier transform)

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

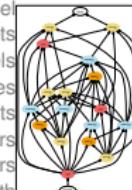
Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math



Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods: convergence analysis, complexity

Computational aspects of iterative methods

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Late binding / communication minimization

Computational aspects of iterative methods

Parallel LU through banded dissection

Incomplete approaches to matrix factorization

Multicore based algorithms

N body problems: naive and equivalent simulations

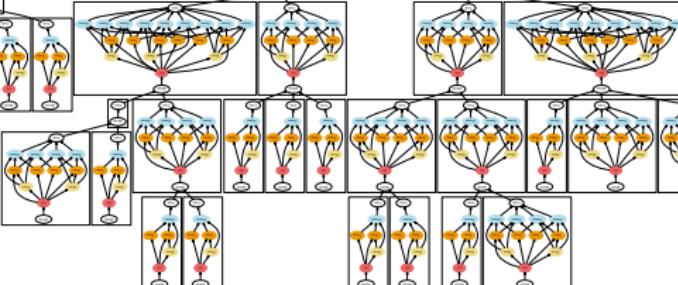
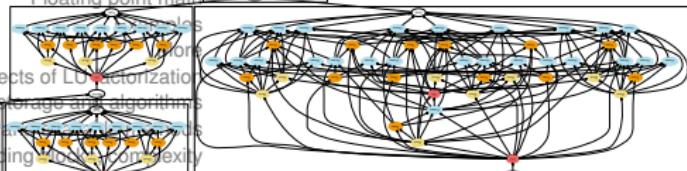
Graph analytics, interpretation approaches, calems

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.



Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Algorithmic concepts

DAG schedulers

The SIMD/MIMD/SPMD/SIMT paradigm is all about
Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts
Programming models

Load balancing, locality, space-filling curves
First we dig into bits
Integers

Floating point numbers
Floating point IEEE
Examples

- Directed Acyclic Graph (dataflow)

- Each node has dependence on other nodes, can execute when
dependencies available

Iterative methods, basic concepts and available methods

- Quark/DaGue (TN): dependence on memory area written
pretty much limited to dense linear algebra

- OpenMP has a pretty good scheduler

- Distributed memory scheduling is pretty hard

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Applications

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More

We briefly discuss two applications that, while at first glance not linear-algebra like, surprisingly can be covered by the foregoing concepts.

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods: basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

Justification

Table of Contents

Processor Architecture

- Structure of a modern processor

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

- Memory hierarchy: caches, register, TLB.

Interconnects and topologies; theoretical concepts

Programming models

Load balancing, locality, space-filling curves

- Multicore issues

First we dig into bits

Integers

- Programming strategies for performance

Floating point numbers

Floating point math

Examples

More

- The power question

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Parallelism

Iterative methods; basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

- Basic concepts

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

- Theoretical concepts

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

- The SIMD/MIMD/SPMD/SIMT model for parallelism

N-body problems: naive and equivalent formulations

Multicore block algorithms

- Characterization of parallelism by memory model

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Programming models

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves

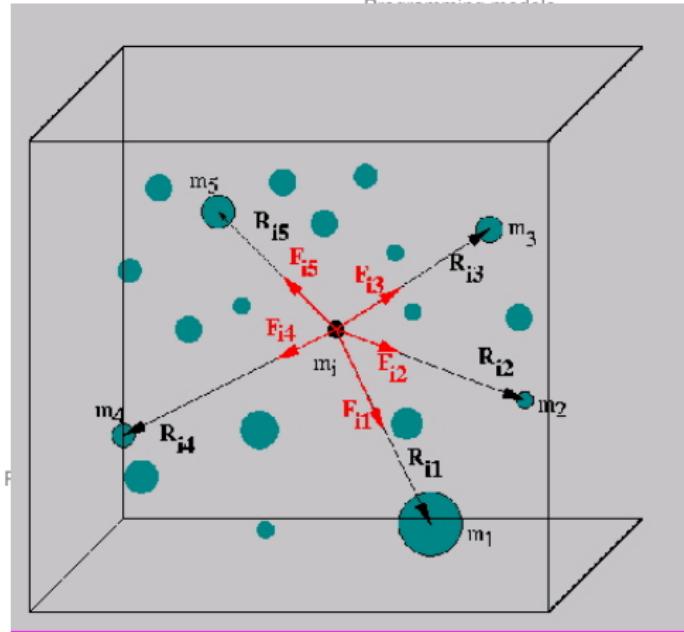
Final exam: discrete bits

N-body problems: naive and equivalent formulations

Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models

Summing forces



Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/piv model of parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

let \bar{r}_{ij} be the vector between i and j ;

essential aspects of LC calculation

Sparse matrices: storage and algorithms

Iterative methods; basic concepts and available methods

Collectives and building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Locality using sparse matrix-vector products

Computational aspects of iterative methods

$f_{ij} = -\frac{m_i m_j}{|\bar{r}_{ij}|^3}$

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Sum forces and move particle over Δt

Parallel and implicit factorizations; weak scaling; approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Particle interactions

for each particle i

for each particle j

let \bar{r}_{ij} be the vector between i and j ;

then the force on i because of j is

$$f_{ij} = -\frac{m_i m_j}{|\bar{r}_{ij}|^3}$$

(where m_i, m_j are the masses or charges) and

$$f_{ji} = -f_{ij}$$

parallel LU through nested dissection

Incomplete approaches to matrix factorization

Sum forces and move particle over Δt

Parallel and implicit factorizations; weak scaling; approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMV/SIMD model of parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

- **Naive all-pairs algorithm: $O(N^2)$**

Sparse matrices: storage and algorithms

- **Clever algorithms: $O(N \log N)$, sometimes even $O(N)$**

Collectives as building blocks, complexity

Scalability analysis of dense matrix-vector product

Scalable matrix multiplication

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Complexity reduction

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

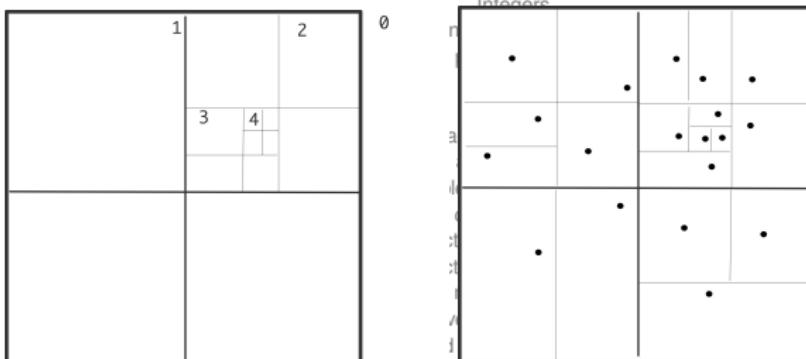
Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits



Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Build concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Decomposition and topology; theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First write bit into bits

Integers

endfor

Floating point numbers

Traverse the Quad_Tree eliminating empty leaves

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, banded, frontal and available methods

// n has 4 children

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and programming strategies; applications

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analysis, interpretation and visualization problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

store particle j in node n

end

Profiling and debugging; optimization and programming strategies.

Dynamic octree creation

Procedure Quad_Tree_Build

Characterization of parallelism by memory model

Decomposition and topology; theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First write bit into bits

Integers

endfor

Floating point numbers

Traverse the Quad_Tree eliminating empty leaves

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, banded, frontal and available methods

// n has 4 children

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and programming strategies; applications

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analysis, interpretation and visualization problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

store particle j in node n

end

Profiling and debugging; optimization and programming strategies.

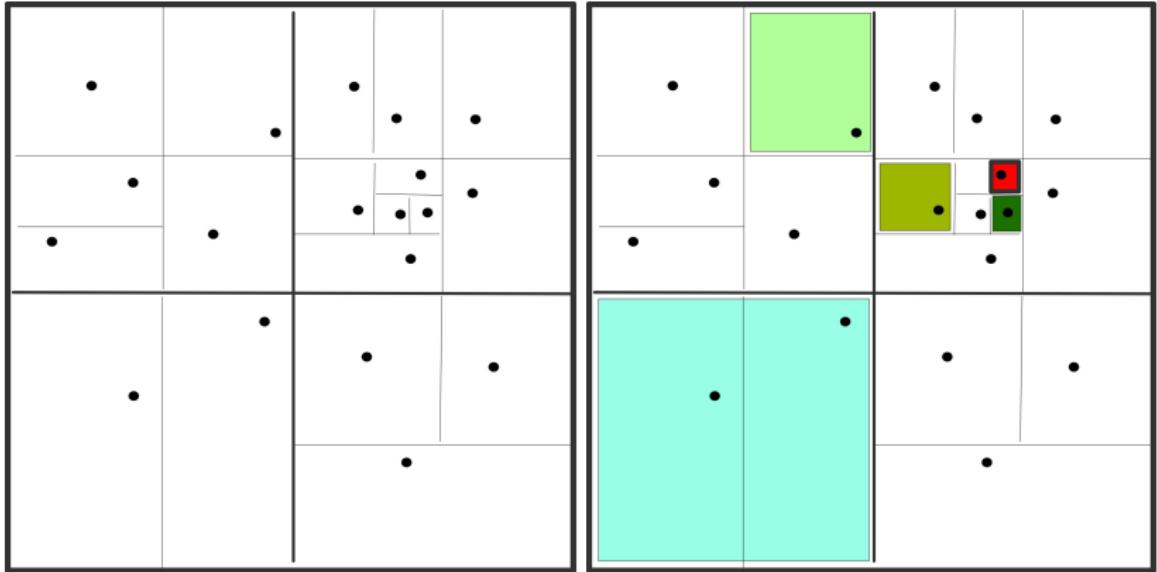
// n empty

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More

- Consider cells on the top level
 - Easier to analyze if LU factorization
 - Sparse matrices: storage and algorithms
 - if distance/diameter ratio small enough, take center of mass
 - Collectives as building blocks, complexity
 - Scalability analysis of dense matrix-vector product
 - Parallel LU through nested dissection
 - Incomplete approaches to matrix factorization
 - otherwise consider children cells
 - Latency hiding / communication minimizing
 - Computational aspects of iterative methods
 - Parallel LU through nested dissection
 - Incomplete approaches to matrix factorization
 - Parallelism and implicit operations: wavefronts, approximation
 - Multicore block algorithms
- N-body problems: naive and equivalent formulations**
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Octree algorithm

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts



Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT memory mechanism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

// Compute the CM = Center of Mass and TM = Total Mass of all the particles

Load balancing, locality, space-filling curves

(TM, CM) = Compute_Mass(root)

Integers

Floating point numbers

Floating point matrix

function (TM, CM) = Compute_Mass(n)

if n contains 1 particle Examples

More

store (TM, CM) at n

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

return (TM, CM)

Iterative methods, basic concepts and available methods

else Collectives and hiding hidden complexity

Scalability analysis of dense matrix-vector product

Dense matrix-vector product

Latency-hiding, communication minimizing

Computational aspects of iterative methods

(TM(j), CM(j)) = Compute_Mass(c(j))

// process parent after all children

for all children c(j) of n

Computational aspects of iterative methods

(TM(j), CM(j)) = Compute_Mass(c(j))

Latency-hiding, communication minimizing

Computational aspects of iterative methods

(TM(j), CM(j)) = Compute_Mass(c(j))

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

TM = sum over children of n: TM(j)

Matrix-vector algorithm

N-body problems: naive and equivalent formulations

// center of mass is weighted sum

Graph analytics, interpretation as sparse matrix problems

Delimited datatype

CM = sum over children j of n: TM(j)*CM(j) / TM

Communicator manipulation

store (TM, CM) at n

Non-blocking collectives

return (TM, CM)

Profiling and debugging; optimization and programming strategies.

Masses calculation

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts

Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples

- **Each cell considers ‘rings’ of equi-distant cells**
 - More
Essential aspects of LU factorization
- **but at doubling distance**
 - Sparse matrices: storage and algorithms
Iterative methods; basic concepts and available methods
 - Collectives as building blocks; complexity
Sparse matrix-vector product
 - Latency hiding; communication minimizing
Computational aspects of iterative methods
 - Parallel LU through nested dissection
 - Incomplete approaches to matrix factorization
- **clog N cells to consider for each particle**
- **$N \log N$ overall**

Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

Complexity

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SMP/SU memory hierarchy
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples

- After position update, particles can move to next box: load redistribution
 - Essential aspects of LU factorization
 - Sparse matrices: storage and algorithms
 - Iterative methods, basic concepts and available methods
- Naive octree algorithm is formulated for shared memory
 - Collectives as building blocks; complexity
 - Spatial traversal, collective vector product
 - Sparse matrix-vector product
 - Octree addition, parallel minimization
 - Computational aspects of iterative methods
 - Parallel LU through nested dissection
 - Incomplete approaches to matrix factorization
- Distributed memory by using inspector-executor paradigm
 - Parallelism and implicit operations: wavefronts, approximation
 - Multicore block algorithms
 - N-body problems: naive and equivalent formulations
 - Graph analytics, interpretation as sparse matrix problems
 - Derived datatypes
 - Communicator manipulation
 - Non-blocking collectives
 - One-sided communication
 - Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/PVM/SPMV model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

for level ℓ from one above the finest to the coarsest:

Essential aspects of LU factorization
Sparse matrices: storage and algorithms

Iterative methods: basic concepts and available methods

Collectives as building blocks; complexity

Scalability (ℓ) analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

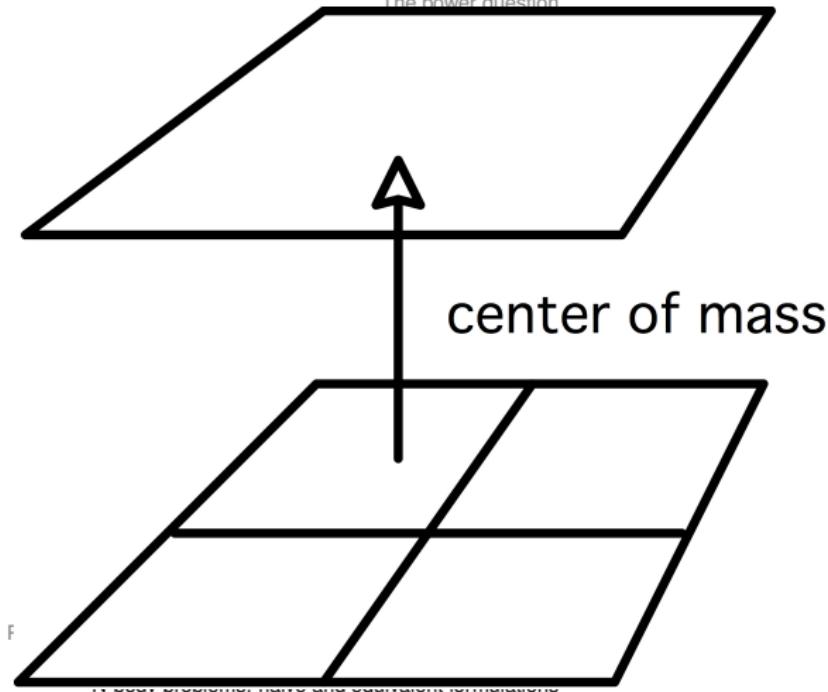
Step 1: force by a particle

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question



Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/PVM/SPMD model for parallel systems

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Arithmetic

More

for level ℓ from one below the coarsest to the finest:

for each cell c on level ℓ :

case diagonal blocks: LU factorization

Sparse matrices: storage and algorithms

Iterative methods, biconjugate gradient-like methods

Collectives as building blocks: complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Step 2: force on a particle

The SIMD/MIMD/PVM/SPMD model for parallel systems

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Arithmetic

More

for level ℓ from one below the coarsest to the finest:

for each cell c on level ℓ :

case diagonal blocks: LU factorization

Sparse matrices: storage and algorithms

Iterative methods, biconjugate gradient-like methods

Collectives as building blocks: complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

let $f_p^{(\ell)}$ be the sum of

1. the force $f_p^{(\ell-1)}$ on the parent p of c , and

2. the sums $g_i^{(\ell)}$ for all i on level ℓ that

satisfy the cell opening criterium

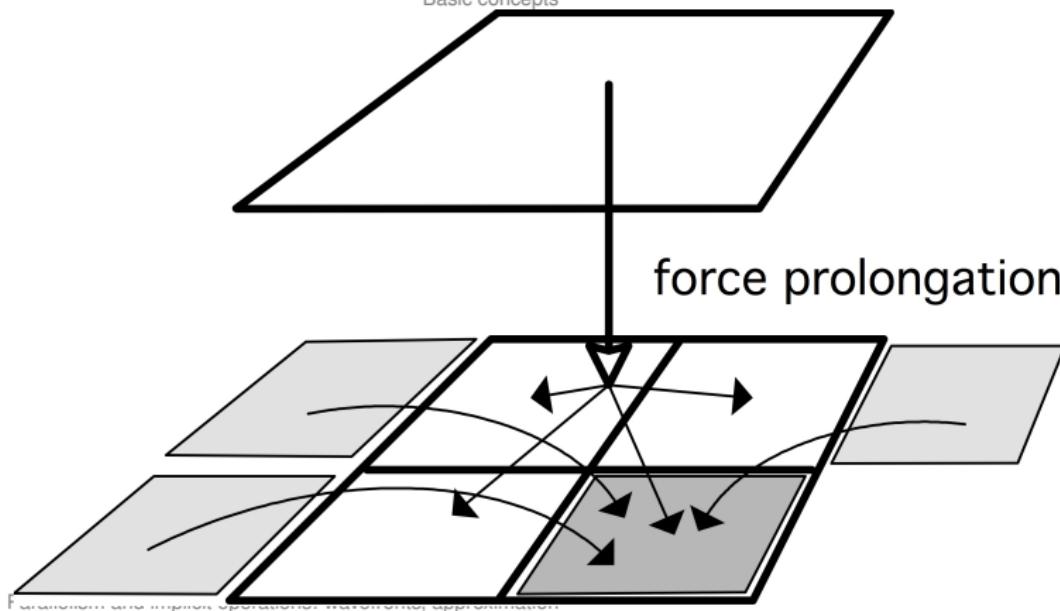
Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts



Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

- Structure of a modern processor
- Memory hierarchy: caches, register, TLB.
- Multicore issues
- Programming strategies for performance
 - The power question
 - Basic concepts
 - Theoretical concepts
- The SIMD/MIMD/SPMD/SIMT model for parallelism
 - Characterization of parallelism by memory model
 - Interconnects and topologies, theoretical concepts
 - Programming models
 - Load balancing, locality, space-filling curves
- Center of mass calculation and force prolongation are local**
- Force from neighbouring cells is a neighbour communication**
- Neighbour communication can start before up/down tree calculation is finished! latency hiding**
- Iterative methods, basic concepts and available methods
 - Floating point numbers
 - Exact vs. floating point arithmetic
 - Examples
 - Implementation
 - Essential aspects of LU factorization
 - Parallelization techniques and algorithms
- Collectives as building blocks; complexity
- Scalability analysis of dense matrix-vector product
 - Sparse matrix-vector product
 - Latency hiding / communication minimizing
 - Computational aspects of iterative methods
 - Parallel LU through nested dissection
 - Incomplete approaches to matrix factorization
- Parallelism and implicit operations: wavefronts, approximation
 - Multicore block algorithms
- N-body problems: naive and equivalent formulations**
- Graph analytics, interpretation as sparse matrix problems
 - Derived datatypes
 - Communicator manipulation
 - Non-blocking collectives
 - One-sided communication
- Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model and its variants

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

- **Traditional algorithm: distribute particles, for each particle gather and update compute**
 - Essential aspects of LU factorization
 - Sparse matrices: storage and algorithms
 - Iterative methods, basic concepts and available methods
- **Problem: allgather has $O(N)\beta$ cost**
 - Collectives as building blocks: complexity
 - Scalability analysis based on matrix-vector product
 - Sparse matrix-vector product
 - Linear scaling of communication
 - Computational aspects of iterative methods
- **does not go down with P , so does not scale weakly**

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

All-pairs methods

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

1.5D calculation

The SIMD/MIMD/SPMD/SIMT model of parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

- Better algorithm: use $\sqrt{P} \times \sqrt{P}$ processor grid,

Sparse matrices: storage and algorithms

- Divide particles in bins of N/\sqrt{P}

Collectives as building blocks, complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

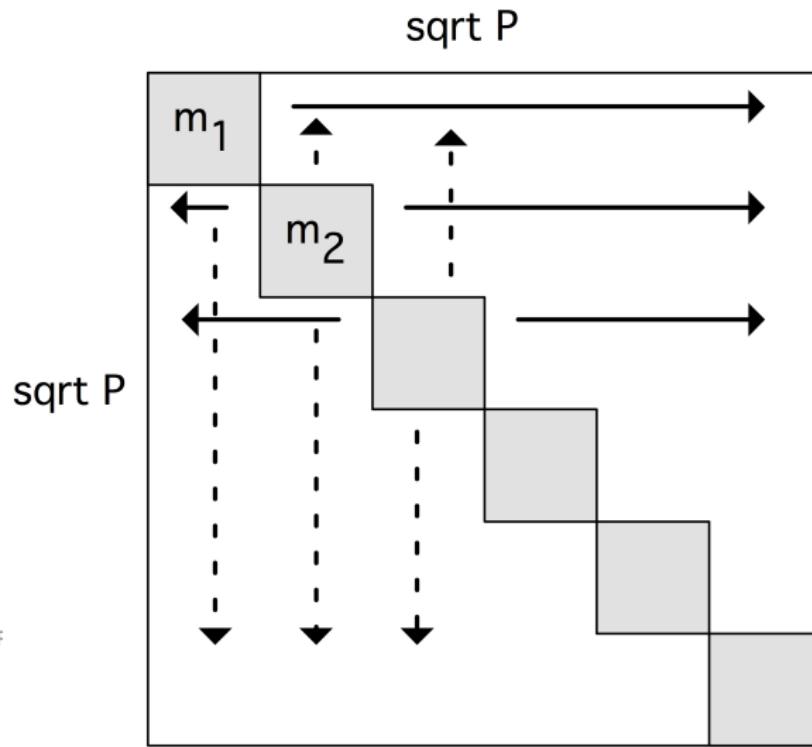
Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

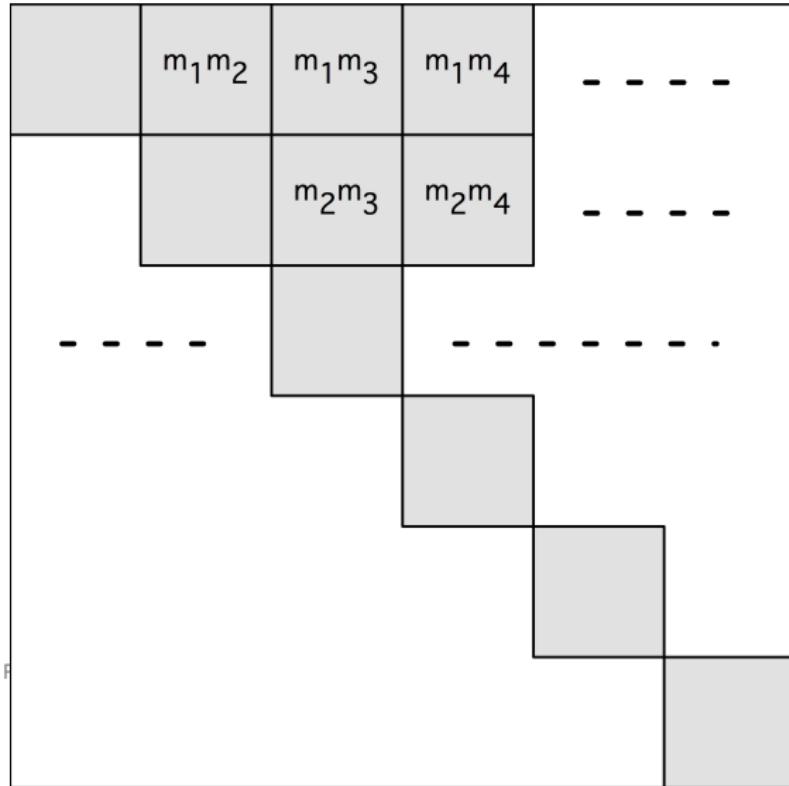
Profiling and debugging; optimization and programming strategies.



Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

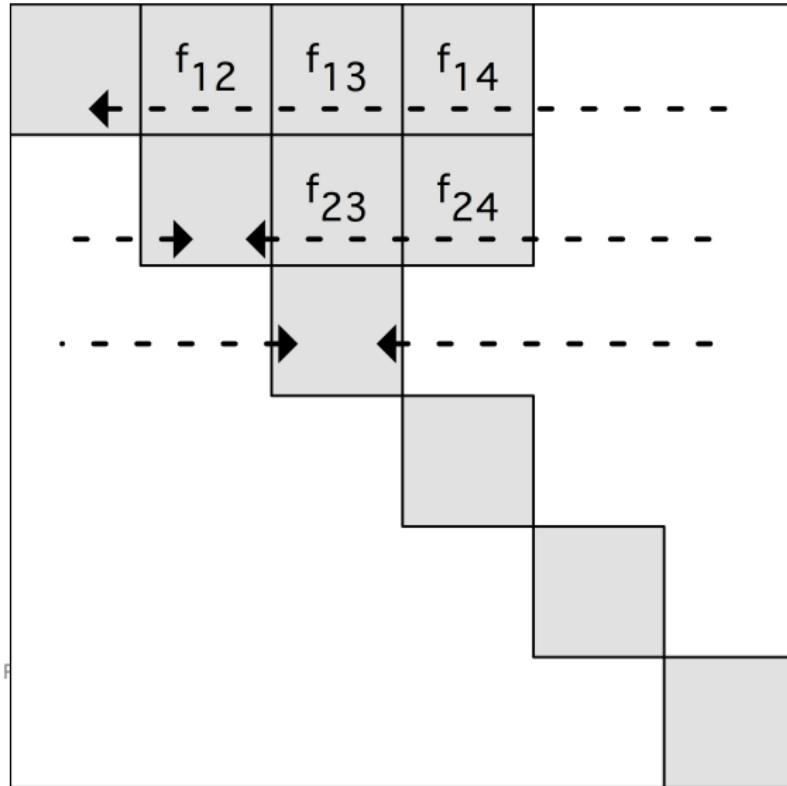
Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues



Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues



Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

- Structure of a modern processor
- Memory hierarchy: caches, register, TLB.
- Multicore issues
- Programming strategies for performance
 - The power question
 - Basic concepts
 - Theoretical concepts
- The SIMD/MIMD/SPMD/SIMT model for parallelism
 - Characterization of parallelism by memory model
 - Interconnects and topologies, theoretical concepts
- Programming models
 - Load balancing, locality, space-filling curves
 - First we dig into bits
- Divide particles in boxes of $M = N/\sqrt{P}$
 - Floating point numbers
 - Floating point math
 - Examples
- Processor (i,j) computes interaction of boxes i and j :
 - More
 - Essential aspects of LU factorization
- Sparse matrices: storage and algorithms
 - Iterative methods, basic concepts and available methods
 - Collectives as building blocks: complexity
 - Matrix-vector product
 - Sparse matrix-vector product
 - Latency hiding / communication minimizing
 - Computational aspects of iterative methods
 - Parallel LU through nested dissection
 - Incomplete approaches to matrix factorization
- Parallelism and implicit operations: wavefronts, approximation
 - Multicore block algorithms
- N-body problems: naive and equivalent formulations
- Graph analytics, interpretation as sparse matrix problems
 - Derived datatypes
 - Communicator manipulation
 - Non-blocking collectives
 - One-sided communication
- Profiling and debugging; optimization and programming strategies.

Processor Architecture

- Structure of a modern processor

Memory hierarchy: caches, register, TLB.

- Memory hierarchy: caches, register, TLB.

Multicore issues

- Multicore issues

Programming strategies for performance

- Programming strategies for performance

The power question

- The power question

Parallelism

85

Basic concepts

- Basic concepts

Theoretical concepts

- Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

- The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

- Characterization of parallelism by memory model

- Structure of a modern processor
- Memory hierarchy: caches, register, TLB.
- Multicore issues
- Programming strategies for performance
 - The power question
 - Basic concepts
 - Theoretical concepts
- The SIMD/MIMD/SPMD/SIMT model for parallelism
 - Characterization of parallelism by memory model
 - Interconnects and topologies, theoretical concepts
 - Programming models
 - Load balancing, locality, space-filling curves

Graph analytics, interpretation as sparse matrix problems

- Sparse matrices: storage and algorithms
- Iterative methods, basic concepts and available methods
 - Collectives as building blocks; complexity
 - Scalability analysis of dense matrix-vector product
 - Sparse matrix-vector product
 - Latency hiding / communication minimizing
 - Computational aspects of iterative methods
 - Parallel LU through nested dissection
 - Incomplete approaches to matrix factorization
- Parallelism and implicit operations: wavefronts, approximation
 - Multicore block algorithms
- N-body problems: naive and equivalent formulations
- Graph analytics, interpretation as sparse matrix problems**
 - Derived datatypes
 - Communicator manipulation
 - Non-blocking collectives
 - One-sided communication
- Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SMT model; parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

- **Traditional: search, shortest path, connected components**

Iterative methods, basic concepts and available methods

- **New: centrality**

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

14.1 Graph algorithms

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

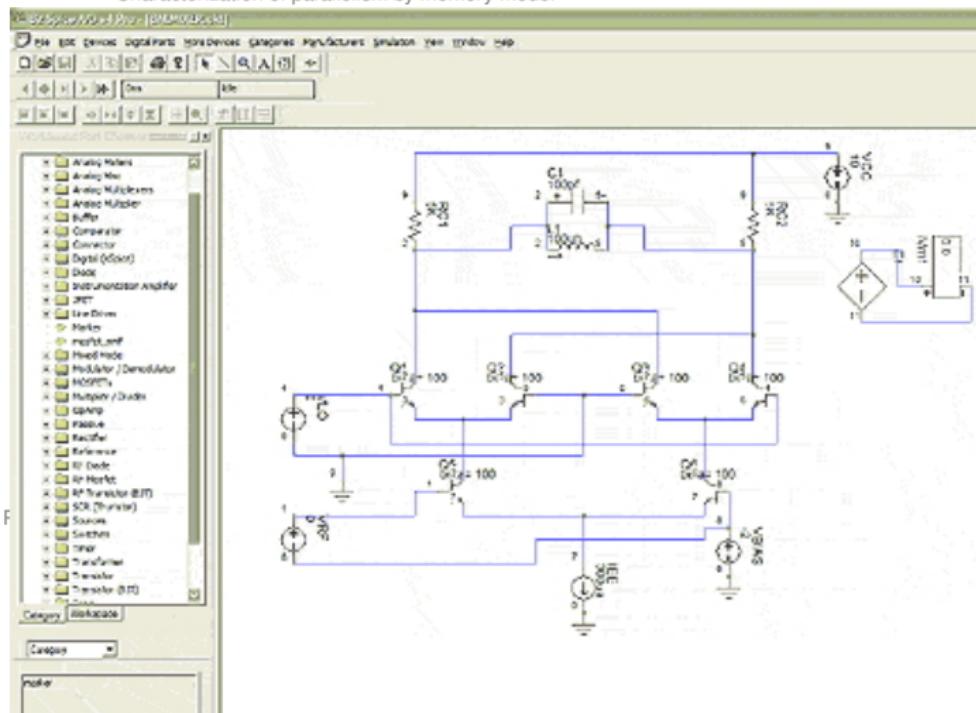
Basic concepts

Theoretical concepts

SIMD/MIMD/SIMT/VM, vector parallelism

142 Traditional use of graph algorithms

Characterization of parallelism by memory model



Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

143 1990s use of graph algorithms

The SIMD/IMM/SIMD model for parallelism
Characterization of parallelism by memory model

Interconnects and topologies, the graph concepts

Programming models

Load balancing, locality, space filling curves

First, we divide bits

Floating point numbers

Floating point math

Examples

More

Aspects of LU factorization

LU factorization, storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding, communication minimizing

Computing with vectors of iterative methods

Parallel LU through nested dissection

Incomplete approximations to matrix factorization

Parallelism and implicit operations, pivots, approximation

LU factorization, block algorithms

Nobody problems, halo and equivalent formulations

Graph analytics interpreted as sparse matrix problems

Derived datatypes

Graph nodes for manipulation

Graph edges for collecting collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

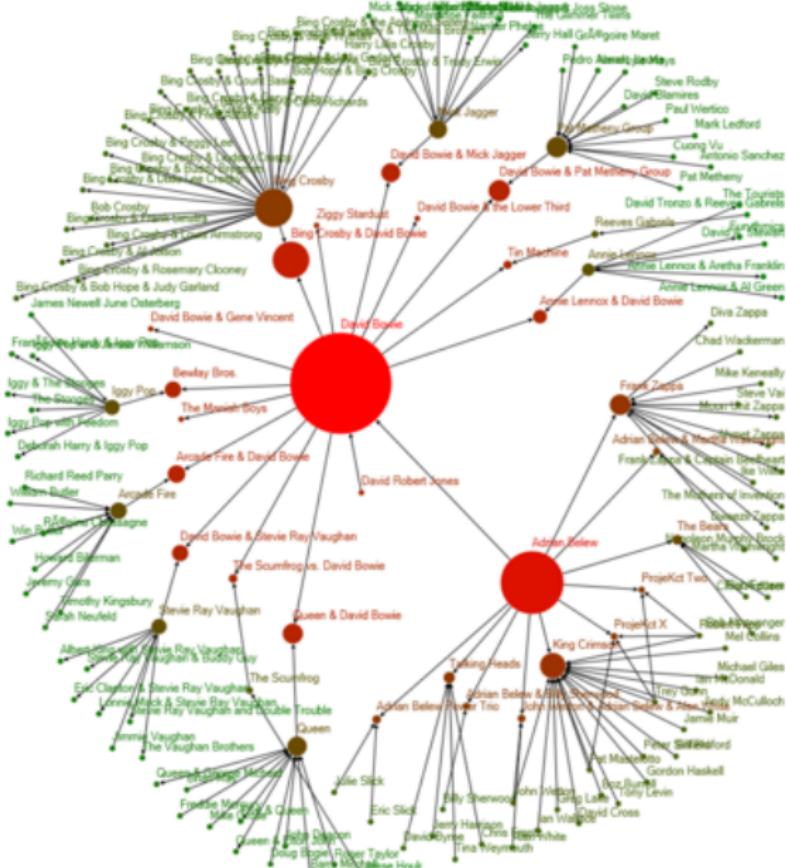
Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The different MPI-2/PVM/Scalapack models and parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves

144 2010 use of graph algorithms



Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

145 2010 use of graph algorithms



Structure of a modern processor
Memory hierarchy: caches, register, TLB.

146 Shortest distance algorithm

Given node s :

The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Basic concepts

Theoretical concepts

$$d: V \rightarrow d(s, v)$$

Programming models

Input: A graph, and a starting node s

Output: A function $d(v)$ that measures the distance from s

Let s be given, and set $d(s) = 0$

Initialize the finished set as $U = \{s\}$

Iterative methods, basic concepts and available methods

Set $c = 1$
Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product
Matrix-vector product

while not finished **do**

Latency hiding / communication minimizing
Computational approaches and methods

Parallel LU through nested dissection
Complete applications to matrix factorization

Parallelism and implicit operations, wavefronts, approximation
Multicore block algorithms

else
N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Set $d(v) = c + 1$ for all $v \in V$.

Packed data type

Communicator manipulation

$U \leftarrow U \cup V$
Non-blocking collectives

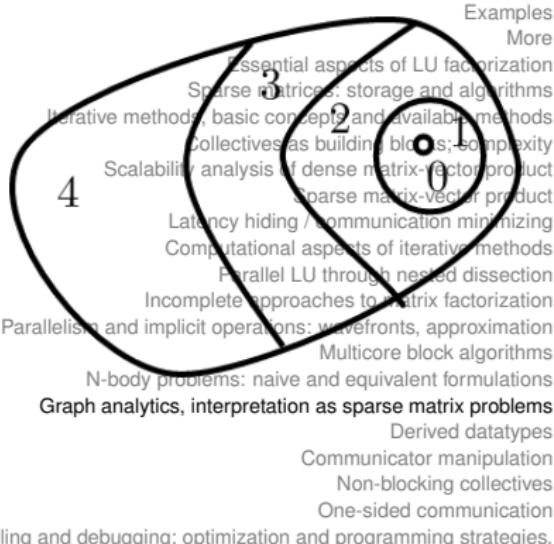
One-sided communication

Increase $c \leftarrow c + 1$

Profiling and debugging; optimization, programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
The reference concepts
The SIMD/MIMD/SPMD/SIMT model in parallel lists
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

The steps in the algorithm are 'level sets':



Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/Multicore debate

148 Computational characteristics

Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts

Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More

- **Uses a queue: central storage**

FIFO discipline, multi-threading
Sparse matrices: storage and algorithms

- **Parallelism not self-evident**

Iterative methods, basic concepts and available methods

Collectives as building blocks, complexity

- **Flexible assignment of work to processors, so no locality**

Scalability analysis of dense matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

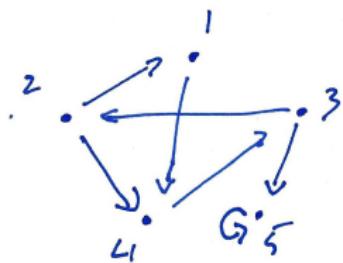
One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

149 Example

Random graph:



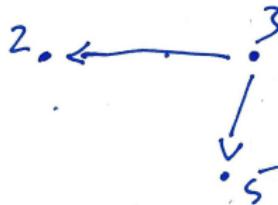
F

Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

150 Level 1

Distance from 3
step 1



N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

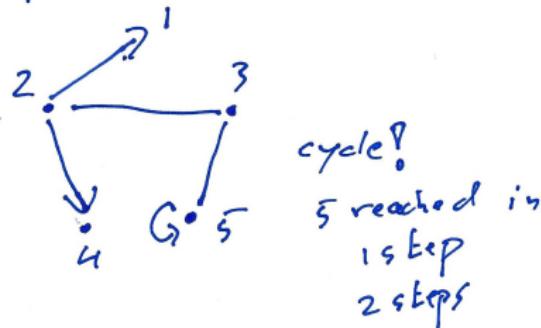
One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

151 Level 2

step 2



N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

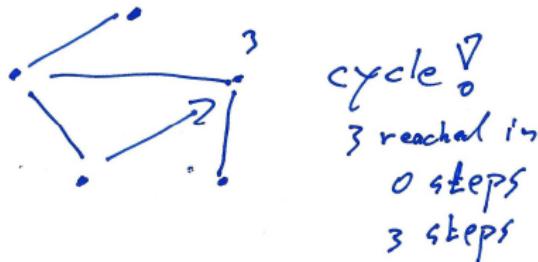
One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers

152 Level 3

step 3



F

Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

opers

math

ples

fore

ition

hms

ods

exity

duct

duct

zing

ods

tion

153 Matrix view

Adjacency matrix

$$\begin{bmatrix} \cdot & \cdot & \cdot & * & \cdot \\ * & \cdot & \cdot & * & \cdot \\ \cdot & * & \cdot & \cdot & * \\ \cdot & \cdot & * & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & * \end{bmatrix}$$

F

multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers

154 Level 1

step 1

$$[\dots \circ \dots] \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & * & \cdot & * \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix} = [\dots 1 \dots 1]$$

Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers

155 Level 2

Step 2

$$\begin{bmatrix} \cdot & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} * & \cdots & * & \cdot \\ \cdot & & & \\ \cdot & & & \\ \cdot & \cdots & \cdots & * \end{bmatrix} = \begin{bmatrix} 2 & \cdots & 2 & \cdot \end{bmatrix} + \begin{bmatrix} \cdots & \cdots & 2 \end{bmatrix}$$

Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

The general concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

models

Summing:

$$\gamma^k + \gamma^k A + \gamma^2 A^2$$

$$[\dots \quad 0 \quad \dots \quad \dots]$$

$$+ [\dots \quad 1 \quad \dots \quad \dots]$$

$$+ [2 \quad \dots \quad 2 \quad 2]$$

$$= [2 \quad 1 \quad 0 \quad 2 \quad 1]$$

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

156 summing up

Gen MatVec product:

$$\gamma^k \xrightarrow{\text{mult}} \otimes \xrightarrow{\text{reduction}} A$$

outerproduct

$$\gamma c \otimes \alpha = \begin{cases} 1 & \text{if } \alpha = 0 \\ \gamma c + 1 & \text{if } \alpha = 1 \end{cases}$$

$$\gamma c \oplus \gamma = \begin{cases} \gamma c & \text{if } \gamma = 1 \\ \gamma & \text{if } \gamma c = 1 \\ \min(\gamma c, \gamma) & \text{otherwise} \end{cases}$$

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

157 All-pairs shortest path

The SIMD/MIMD/PIMD/SIMD model. Multicore issues

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Exceptions

More

$$\Delta_{k+1}(u, v) = \min\{\Delta_k(u, v), \Delta_k(u, k) + \Delta_k(k, v)\}. \quad (10)$$

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Locality hiding, communication minimizing

Computational aspects of iterative methods

Parallelization through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multi-core load balancing

Similarity to Gaussian elimination

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Block matrices

Theoretical concepts

158 Pagerank

T stochastic: all rowsums are 1

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point representation

Floating point math

Examples

More

Essential aspects of LU factorization

Matrix-matrix, storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Implementation of parallel LU matrix factorization

Solution of linear system:

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

Bi-sided communication

$$(I - sT)^{-1} = I + sT + s^2 T^2 + \dots$$

Profiling and debugging; optimization and programming strategies.

Observe

Communicator manipulation

Non-blocking collectives

Bi-sided communication

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SIMT memory model and its variants
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

- Graphs imply sparse matrix-vector product
- ... but the graphs are unlike PDE graphs

Essential aspects of LU factorization
sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods

- low diameter
- high degree
- power law

Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Low-level communication minimizing
Computational aspects of iterative methods

- treat as random sparse: use dense techniques

Parallelism and implicit operations: wavefronts, approximation
incomplete approaches to matrix factorization

- 2D matrix partitioning: each block non-null, but sparse

N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

159 ‘Real world’ graphs

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

160 Parallel treatment

The SIMD/MIMD/SPMD/SIMD model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Arithmetic

More

- Intuitive approach: partitioning of nodes

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

- not scalable \Rightarrow 2D distribution

Scalability analysis of dense matrix-vector product

- equivalent to distribution of edges

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Distributed LU factorization

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

- Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Parallel programming topics

Structure of a modern processor

Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Table of Contents

Processor Architecture

- Structure of a modern processor

- The SIMD/MIMD/SPMD/SIMT model for parallelism

- Characterization of parallelism by memory model

- Memory hierarchy: caches, register, TLB.

- Interconnects and topologies; theoretical concepts

- Programming models

- Load balancing, locality, space-filling curves

- Multicore issues

- First we dig into bits

- Integers

- Programming strategies for performance

- Floating point numbers

- Floating point math

- Examples

- More

- The power question

- Essential aspects of LU factorization

- Sparse matrices: storage and algorithms

Parallelism

85

- Basic concepts

- Collectives as building blocks; complexity

- Scalability analysis of dense matrix-vector product

- Theoretical concepts

- Sparse matrix-vector product

- Latency hiding / communication minimizing

- Computational aspects of iterative methods

- Computational aspects

- Parallel LU through nested dissection

- Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

- The SIMD/MIMD/SPMD/SIMT model for parallelism

- Multicore block algorithms

- N-body problems: naive and equivalent formulations

- Graph analytics, interpretation as sparse matrix problems

- Characterization of parallelism by memory model

- Derived datatypes

- Communicator manipulation

- Non-blocking collectives

- One-sided communication

Profiling and debugging; optimization and programming strategies.

Programming models

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Derived datatypes

more

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
 Memory hierarchy: caches, register, TLB.
 Multicore issues
 Programming strategies for performance
 The power question
 Basic concepts
 Theoretical concepts
 The SIMD/MIMD/SPM/DSM memory model
 Characterization of parallelism by memory model
 Interconnects and topologies, theoretical concepts
 Programming models
 Load balancing, locality, space-filling curves
 First we dig into bits
 Integers

Elementary datatypes

C	F	
	Floating point numbers Floating point math	
	Examples	
MPI_INT	integer	More Essential aspects of LU factorization
MPI_CHAR	signed char	Sparse matrices: band and algorithms Iterative methods, basic concepts and available methods
MPI_LONG	signed long int	Collectives as building blocks, complexity
MPI_UNSIGNED	unsigned int	Scalability analysis of dense matrix-vector product
MPI_FLOAT	float	Efficiency hiding / Communication minimizing
MPI_DOUBLE	double	Computational aspects of iterative methods Parallel LU through nested dissection
MPI_BYTE		Incomplete approaches to matrix factorization
		Parallelism and implicit operations: wavefronts, approximation
		Multicore block algorithms
		N-body problems: naive and equivalent formulations
		Graph analytics, interpretation as sparse matrix problems
		Derived datatypes
		Communicator manipulation
		Non-blocking collectives
		One-sided communication
		Profiling and debugging; optimization and programming strategies.
		(raw I/O by MPI)

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More

- Identify structure in data for efficient transfer
- Recursive construction from elementary types
- Packing is done for you

Sparse matrices: storage and algorithms
Iterative methods: basic concepts and available methods
Collectives as building blocks, complexity
Scalability analysis of dense matrix-vector product
Computational aspects of matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Derived datatypes

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

- **Contiguous: contiguous blocks (kinda pointless)**

Essential aspects of LU factorization

- **Vector: strided blocks**

Sparse matrices: storage and algorithms

- **Indexed: irregular blocks**

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Parallel matrix-vector product

Sparse matrix-vector product

- **Struct: Completely general placement and data types**

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Derived datatypes

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

`MPI_Type_<type>(&oldtype, . . . , &newtype);`

Sparse matrices: storage and algorithms

`MPI_Type_commit(&newtype);`

Iterative methods: basic concepts and iterative methods

Collectives as building blocks; complexity

`MPI_Send(. . . newtype . . .);`

Sparse matrix-vector product

Lateness / latency / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Scheme

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model of parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math

Signature describes the structure of a datatype

Examples
more

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

**Send and receive type do not have to be equal:
only be of equal signature**

Collectives as building blocks; complexity

Scalability: sparse matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Communication patterns: iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Type signature

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers



Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Contiguous

A contiguous datatype consists of a block of elements of a constituent type

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model in parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Data alignment
Floating point math
Blas
More
Computational aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorizations
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Quenching: interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

```
MPI_Datatype newvectortype;
if (mytid==sender) {
    MPI_Type_contiguous(count,MPI_DOUBLE,&newvectortype);
    MPI_Type_commit(&newvectortype);
    MPI_Send(source,1,newvectortype,receiver,0,comm);
    MPI_Type_free(&newvectortype);
} else if (mytid==receiver) {
    MPI_Status recv_status;
    int recv_count;
    MPI_Recv(target,count,MPI_DOUBLE, sender, 0, comm,
             &recv_status);
    MPI_Get_count(&recv_status,MPI_DOUBLE,&recv_count);
    ASSERT(count==recv_count);
}
```

Contiguous

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves



Profiling and debugging; optimization and programming strategies.

A vector datatype is built up out of strided blocks of elements of a constituent type

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, page filling curves

First we dig into bits

MPI_Datatype newvectortype;

if (mytid==sender) { Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Collective communication: block-based matrix multiplication

Collectives as building blocks: complexity

Scalability analysis of dense matrix-vector product

MPI_Type_free(&newvectortype);

} else if (mytid==receiver) {

MPI_Status recv_status; Sparse matrix-vector product

Latency hiding / communication minimizing

int recv_count; Complexity aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to LU factorization

Parallelism and implicit operations: wavefronts, approximation

&recv_status); Multicore block algorithms

MPI_Get_count(&recv_status, MPI_DOUBLE, &recv_count);

Graph analytics, interpretation as sparse matrix problems

ASSERT(recv_count==count); Devoid datatypes

}

Communicator manipulation

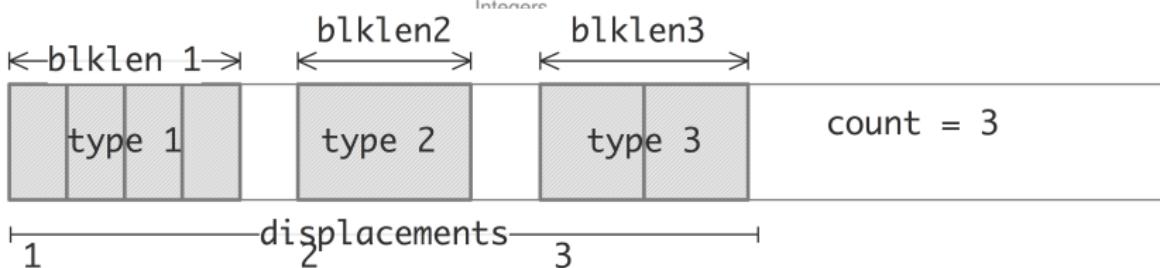
Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Vector

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model of parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits



Latency hiding / communication minimizing
Computational aspects of iterative methods

Indexed and Struct types have arbitrary blocks with arbitrary placements; Struct can have multiple types

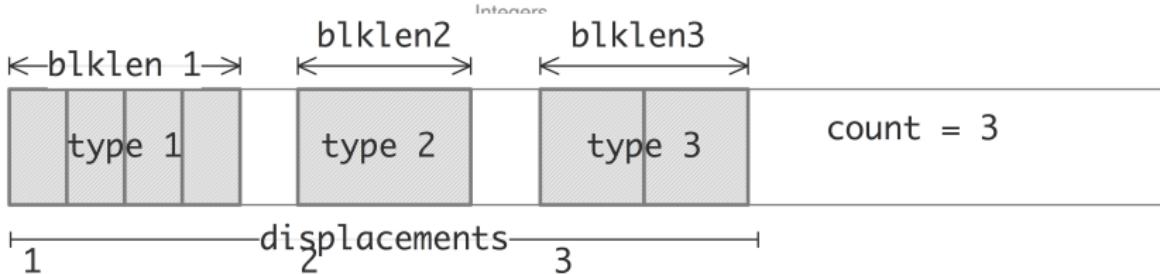
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit coalescing way from approximations
Multicore block algorithms

N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
 Memory hierarchy: caches, register, TLB.
 Multicore
 Programming strategies for performance
 The power question
 indices = (int*) malloc(count*sizeof(int));
 blocklengths = (int*) malloc(count*sizeof(int));
 source = (double*) malloc(totalcount*sizeof(double));
 target = (double*) malloc(count*sizeof(double));
 Load balancing, locality, space-filling curves
 First we dig into bits
 MPI_Datatype newvectortype;
 if (mytid==sender) {
 MPI_Type_indexed(count,blocklengths,indices,MPI_DOUBLE,&newvecto
 Essential aspects of LU factorization
 MPI_Type_commit(&newvectortype);
 Iterative methods, basic concepts and available methods
 MPI_Send(source,1,newvectortype,the_other,0,comm);
 MPI_Type_free(&newvectortype);
 } else if (mytid==receiver) {
 MPI_Status recv_status;
 int recv_count;
 Parallelism and implicit operations: wavefronts, approximation
 MPI_Recv(target,count,MPI_DOUBLE,the_other,0,comm,
 N-body problems: naive and equivalent formulations
 &recv_status);
 MPI_Get_count(&recv_status,MPI_DOUBLE,&recv_count);
 ASSERT(recv_count==count);
 } Profiling and debugging; optimization and programming strategies.

Indexed

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits



Latency hiding / communication minimizing
Computational aspects of iterative methods

Components of a structure are not necessarily aligned:
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallel linear 1-bit operations via sparse approximation
Multicore block algorithms

N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

Struct

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

struct myobject {

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

char c;

Programming models

double x[2];

Load balancing, locality, space-filling curves

First we dig into bits

int i;

Integers

Floating point numbers

Floating point math

Examples

More

MPI_Datatype newstructuretype;

Sparse matrices: storage and algorithms

Iterative multi-level concept and available methods

Collectives as building blocks: complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Multiple approaches to matrix-vector operation

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

displacements[0] = (size_t)&(myobject.c) - (size_t)&myobject;

blocklengths[0] = 1; types[0] = MPI_CHAR;

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

blocklengths[1] = 2; types[1] = MPI_DOUBLE;

displacements[1] = (deriveddatatype)(myobject.x[0]) - (size_t)&myobject;

Communicator manipulation

blocklengths[2] = 1; types[2] = MPI_INT;

Non-blocking collectives

displacements[2] = (size_t)&(myobject.i) - (size_t)&myobject;

One-sided communication

Profiling and debugging; optimization and programming strategies.

Struct

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
MPI_Type_create_struct()
Load balancing, locality, space filling curves
First we dig into bits
&newstructuretype),
Integers
Floating point numbers
Floating point math
MPI_Type_commit(&newstructuretype);
{
 MPI_Aint typesize;
 MPI_Type_extent(&newstructuretype, &typesize);
 if (mytid==0) printf("Type extent: %d bytes\n", typesize);
}
Sparse matrix-vector product
Latency hiding / communication minimizing
if (mytid==sender){
 MPI_Send(&myobject, 1, newstructuretype, the_other, 0, comm);
}
else if (mytid==receiver){
 MPI_Recv(&myobject, 1, newstructuretype, the_other, 0, comm,
 MPI_STATUS_IGNORE);
}
MPI_Type_free(&newstructuretype);
Profiling and debugging, optimization and programming strategies.

Struct (cont'd)

Packing: another approach to heterogeneous types

- The MPI_Pack command adds data to a send buffer;
- the MPI_Unpack command retrieves data from a receive buffer;
- the buffer is sent with a datatype of MPI_PACKED.

```
int MPI_Pack(void *inbuf, int incount, MPI_Datatype datatype,
             void *outbuf, int outcount, int *position,
             MPI_Comm comm);
```

Profile problems: native and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Communicator manipulation
Non-blocking collectives
One-sided communication

```
int MPI_Unpack(void *inbuf, int incount, MPI_Datatype datatype,
               void *outbuf, int outcount, int *position,
               MPI_Comm comm);
```

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
Pack example
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
if (mytid==sender) {
 MPI_Pack(&nsevents, 1, MPI_INT, buffer, buflen, &position, comm);
 for (i=0; i<nsevents; i++) {
 double value = rand() / (double) RAND_MAX;
 MPI_Pack(&value, 1, MPI_DOUBLE, buffer, buflen, &position, comm);
 }
 MPI_Pack(&nsevents, 1, MPI_INT, buffer, buflen, &position, comm);
 MPI_Send(buffer, position, MPI_PACKED, other, 0, comm);
}
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space filling curves

First we dig into bits

int irecv_value; Integers

Floating point numbers

double xrecv_value; Floating point math

MPI_Recv(buffer, buflen, MPI_PACKED, other, 0, comm,

More Examples

MPI_ESTATISTICS_IGNORE); Essential aspects of parallelization

Sparse matrices: storage and algorithms

MPI_Unpack(buffer, buflen, &position, &nends, 1, MPI_INT, comm); Collective Unpack: component and bottom-up methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

MPI_Unpack(buffer, buflen, &position, &xrecv_value, 1,

Latency hiding / communication minimizing

Input-output aspects: domain methods

Parallel LU through nested dissection

} Incomplete approaches to matrix factorization

Parallel direct and implicit operations: wavefronts, approximation

Multicore block algorithms

MPI_Unpack(buffer, buflen, &position, &irecv_value, 1,

N-body problems: tree and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

ASSERT(irecv_value==nends); Denoading

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Pack example (cont'd)

```
} else if (mytid==receiver) {  
    int irecv_value; Integers  
    double xrecv_value; Floating point numbers  
    MPI_Recv(buffer, buflen, MPI_PACKED, other, 0, comm,  
             MPI_ESTATISTICS_IGNORE);  
    MPI_Unpack(buffer, buflen, &position, &nends, 1, MPI_INT, comm);  
    for (i=0; i<nends; i++) {  
        MPI_Unpack(buffer, buflen, &position, &xrecv_value, 1,  
                  MPI_DOUBLE, comm);  
        Parallel LU through nested dissection  
    }  
    Incomplete approaches to matrix factorization  
    MPI_Unpack(buffer, buflen, &position, &irecv_value, 1,  
              MPI_INT, comm);  
    N-body problems: tree and equivalent formulations  
    Graph analytics, interpretation as sparse matrix problems  
    ASSERT(irecv_value==nends); Denoading  
}
```

Table of Contents

Processor Architecture

- Structure of a modern processor

The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model

- Memory hierarchy: caches, register, TLB.

Interconnects and topologies; theoretical concepts
Programming models

Load balancing, locality, space-filling curves

- Multicore issues

First we dig into bits
Integers

- Programming strategies for performance

Floating point numbers
Floating point math
Examples

- The power question

Essential aspects of LU factorization
Sparse matrices: storage and algorithms

Parallelism

Iterative methods: basic concepts and available methods
Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product
Sparse matrix-vector product

- Basic concepts

Latency hiding / communication minimizing

Computational aspects of iterative methods

- Theoretical concepts

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

- The SIMD/MIMD/SPMD/SIMT model for parallelism

N-body problems: naive and equivalent formulations

- Characterization of parallelism by memory model

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Programming models

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Communicator manipulation

more

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SMP/SPMD model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More
• Communicator duplication
Essential aspects of LU factorization
Scalability analysis of LU factorization
Iterative methods, basic concepts and available methods
• Disjoint subcommunicators
Scalability analysis of LU factorization
Scalability analysis of dense matrix-vector product
• Nondisjoint subcommunicators
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
• Topologies, inter-communicators, spawning (sorry, not in this lecture)
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Communicator trickery

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/distributed paradigm

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Simplest new communicator: identical copy

Equivalent to MPI's MPI_COMM_SELF

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Useful for libraries

Scalability analysis of dense matrix-vector product

separate library traffic from application

Latency Hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Communicator duplication

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Equivalent to MPI's MPI_COMM_SELF

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Scalability analysis of dense matrix-vector product

Latency Hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Comm dup
Programming strategies for performance
The power question
class library {
private:
 New SIMD/MIMD/SPMD/SIMT model for parallelism
 Characterization of parallelism by memory model
 Interconnects and topologies, theoretical concepts
 Programming models
 Load balancing, locality, space-filling curves
 Bit manipulation; big into bits
 MPI_Request request[2];
public:
 library(MPI_Comm incomm) {
 comm = incomm;
 MPI_Comm_rank(comm, &mytid);
 other = 1 - mytid;
 };
 void communication_start() {
 int sdata=6, rdata;
 MPI_Isend(&sdata, 1, MPI_INT, other, 2, comm, &(request[0]));
 Parallel LU through nested dissection
 Incomplete approaches to matrix factorization
 Parallelism and implicit operations: wavefronts, approximation
 MPI_Irecv(&rdata, 1, MPI_INT, other, MPI_ANY_TAG, comm, &(request[1]));
 Datacube blocking algorithm
 N-body problems: naive and equivalent formulations
 Graph analysis: interpretation as sparse matrix problems
 Derived datatypes
 Communicator manipulation
 MPI_Status status[2];
 MPI_Waitall(2, request, status); }
 Profiling; and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
Comm dup (cont'd)
The SIMD/MIMD/SPMD/SMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
MPI_Irecv(&sdata, 1, MPI_INT, other, 1, comm, &(request[0]));
my_library.communication_start();
MPI_Irecv(&rdata, 1, MPI_INT, other, MPI_ANY_TAG, comm,
Collectives as building blocks; complexity
Scalability analysis: matrix-vector product
&(request[1]));
Sparse matrix-vector product
Latency hiding: communication minimizing
Computational aspects: iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

Communicator splitting

The SIMD/MIMD/SPMD/SIMT model for parallelism

Processor grid:

Distribution of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

`MPI_Comm_rank(&mytid);`

Floating point to bits

Integers

`proc_i = mytid % proc_column_length;`

Floating point numbers

`proc_j = mytid / proc_column_length;`

Floating point math

Equations

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Communicator per column:

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

`MPI_Comm_column_comm;`

Parallel LU through nested dissection

Simple approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Matrix-free methods: element local matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

`MPI_Bcast(data, /* tag: */ 0, column_comm);`

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies; performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

MPI_Comm_size(comm, &npes);

Interconnects and topologies, theoretical concepts

Programming models

MPI_Comm_rank(comm, &rank);

Load balancing, locality, space-filling curves

First we dig into bits

MPI_Comm_split(comm, icolor, key, &newcomm);

Floating point numbers

MPI_Comm_rank(newcomm, &newrank);

Examples

More

Essential aspects of LU factorization				
rank	icolor	key	newrank	color zero
Iterative methods, basic concepts and available methods				
0	0	9	2	←
	Collectives as building blocks; complexity			
1	1	8	2	Sparse matrix-vector product
	Scalability analysis of dense matrix-vector product			
2	2	7	2	Latency hiding / communication minimizing
	Computational aspects of iterative methods			
3	0	6	1	Parallel LU through nested dissection
	complete approaches to matrix factorization			
4	1	5	1	←
	Parallelism and implicit operations: wavefronts, approximation			
5	2	4	1	Multicore block algorithms
	N-body problems: naive and equivalent formulations			
6	0	3	0	Graph analytics, interpretation as sparse matrix problems
	Derived datatypes			
7	1	2	0	Communicator manipulation
	Non-blocking collectives			
8	2	1	0	One-sided communication
	Profiling and debugging; optimization and programming strategies.			

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SIMI/MIMI model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Locating point mass

Communicator to group to communicator:

More
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks: complexity
Scalability analysis of dense matrix-vector product

```
MPI_Comm_group( comm, &group );  
MPI_Comm_create( old_comm, group, &new_comm );
```

Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel matrix-vector multiplication: distribution
Incomplete approaches to matrix factorization
Parallelism and multilevel methods: hierarchical computation
Multicore block algorithms

**and groups are manipulated with MPI_Group_incl,
MPI_Group_excl, MPI_Group_difference and a few more.**

N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Table of Contents

Processor Architecture

- Structure of a modern processor

The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model

- Memory hierarchy: caches, register, TLB.

Interconnects and topologies; theoretical concepts
Programming models

Load balancing, locality, space-filling curves

- Multicore issues

First we dig into bits

Integers

- Programming strategies for performance

Floating point numbers

Floating point math

Examples

More

- The power question

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Parallelism

85

Iterative methods; basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

- Basic concepts

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

- Theoretical concepts

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

- The SIMD/MIMD/SPMD/SIMT model for parallelism

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

- Interconnects and topologies; theoretical concepts

Profiling and debugging; optimization and programming strategies.

Programming models

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Non-blocking collectives

more

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model

Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts

Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples

- **Collectives are blocking**
More
Essential aspects of LU factorization
- **Any process delay visible to every other process**
Sparse matrices: storage and algorithms
Iterative methods; basic concepts and available methods
Collectives as building blocks; complexity
- **Some architectures have separate network for collectives**
Matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
- **Idea: let collective progress independently, test for completion**

Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

General idea

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Example

Non-blocking collective gives MPI_Request pointer

```
int MPI_Iallreduce(const void *sendbuf, void *recvbuf,  
                   int count, MPI_Datatype datatype,  
                   MPI_Op op, MPI_Comm comm,  
                   MPI_Request *request)
```

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods: basic concepts and available methods
Collectives as building blocks, complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding (communication minimizing)
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
MPI_Datatype datatype
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Test for completion with MPI_Wait and MPI_Test and such.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model and parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples

Overlapping collective communication with useful computation, Example: in iterative methods More

Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks, complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

$x^t \leftarrow x$ non-blocking start

$y \leftarrow Ax$
 $y \leftarrow y/x^t$

Processor Architecture

- Structure of a modern processor

- Memory hierarchy: caches, register, TLB.

- Multicore issues

- Programming strategies for performance

- The power question

Parallelism

85

- Basic concepts

- Theoretical concepts

- The SIMD/MIMD/SPMD/SIMT model for parallelism

- Interconnects and topologies; theoretical concepts

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers

One-sided communication

More
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model and its variants

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Binaries

More

- One-sided access: Put, Get, Accumulate

- Window: memory designated for one-sided communication

- Origin: process that makes the one-sided call

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks, complexity

Scalability analysis of dense matrix-vector product

Latency hiding / communication minimizing

Implementation details of iterative methods

Parallel LU through nested dissection

Incomplete factorizations: matrix factorization

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Basic notions

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space filling curves

First we dig into bits

int disp_unit, MPI_Info info,

Floating point numbers

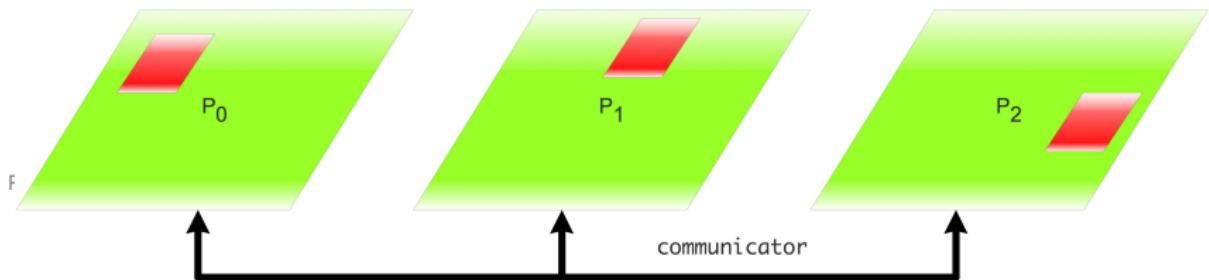
MPI_Comm comm, MPI_Win *win)

Examples

More

Essential aspects of LU factorization

Window



Note: collective on communicator

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/pMIMD model of parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

MPI_Put (void *origin_addr, int origin_count,
 MPI_Datatype origin_datatype, int target_rank,
 MPI_Aint target_disp, int target_count,
 MPI_Datatype target_datatype,
 MPI_Win window)
Iterative methods: basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product

MPI_Accumulate (void *origin_addr, int origin_count,
 MPI_Datatype origin_datatype, int target_rank,
 MPI_Aint target_disp, int target_count,
 MPI_Datatype target_datatype,
 MPI_Op op, MPI_Win window)

Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
One-timestep communication: sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

Put/Get/Accumulate

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/MP model: shared memory parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

`MPI_Win_fence((MPI_MODE_NOPUT | MPI_MODE_NOPRECEDE), win);`

Essential aspects of LU factorization

`MPI_Get(/* operands */, win);`

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

`MPI_Win_fence(MPI_MODE_NOSUCCEED, win);`

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU factorization: nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Globally defined epochs

Induces synchronization

Implicit operations: nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD model (SPP, SIMD, vector parallelism)
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples

MIPS
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
• Data in window undefined until fence synchronization
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Restrictions from memory model

Windows are not coherent with local memory:

- **Data in window undefined until fence synchronization**
- **No put and get in same epoch**

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/Reduced/Memory parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Exposure:

`MPI_Win_post(/* group of origin processes */)`

`MPI_Win_wait()`

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Access:

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete and preconditioned factorization

Parallelism and implicit operations: wavefronts, approximation

`// access operations`

Wavefront block algorithms

`MPI_Win_complete()`

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Fine-grained synchronization

- Structure of a modern processor
- Memory hierarchy: caches, register, TLB.
- Multicore issues
- Programming strategies for performance
- The power question
- Basic concepts
- Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallel computation

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Example

```

if (mytid==origin) {
    MPI_Group_incl(all_group,1,&target,&two_group);
    // access
    MPI_Win_start(two_group,0,the_window);
    MPI_Put( /* data on origin: */ &my_number, 1,MPI_INT,
        /* data on target: */ target,0, 1,MPI_INT,
        the_window);
    MPI_Win_complete(the_window);
}
if (mytid==target) {
    MPI_Group_incl(all_group,1,&origin,&two_group);
    // exposure
    MPI_Win_post(two_group,0,the_window);
    MPI_Win_wait(the_window);
}

```

Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More
Essential aspects of LU factorization
Sparse matrices, storage and algorithms
Iterative methods, basic concepts and available methods
The building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallelizing through nested dissection
Incomplete approaches to matrix factorization
Parallel and implicit operations: movements, update function
Multicore block algorithms
N-body problems: naive and equivalent formulations
MPI functions, implementation and parallel problems
Derived datatypes
Communication optimization
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Interconnects and topologies, theoretical concepts
Start/complete/post/wait example
The SIMD/MMMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
if (mytid==origin) {
 MPI_Group_incl(all_group, 1, &target, &two_group);
 // access
 MPI_Win_start(two_group, 0, the_window);
 MPI_Put(/* data on origin */ &my_number, 1, MPI_INT,
 /* data on target */ target, 0, 1, MPI_INT,
 the_window);
 MPI_Win_complete(the_window);
}
if (mytid==target) {
 MPI_Group_incl(all_group, 1, &origin, &two_group);
 // exposure
 MPI_Win_post(two_group, 0, the_window);
 MPI_Win_wait(the_window);
}
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

Passive target synchronization

The SIMD model: shared memory and parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

- Emulates shared memory

More
Essential aspects of LU factorization

- Origin process locks window on target

Sparse matrices: storage and algorithms

- Iterative methods, basic concepts, visualization techniques

Collectives as building blocks; complexity

Scalability issues: matrix structure and load

Sparse matrix-vector product

- MPI-2 was lacking atomic operations

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

Passive target example

The SIMD/MIMD/ShMIMD model: parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

```
for (int i=0; i<ninputs; i++)
```

 Integers
 Floating point numbers

```
        myjobs[i] = 0;
```

 Floating point math

```
    if (mytid!=repository) {
```

 More

```
        float contribution=(float)mytid,table_element;
```

 Essential aspects of LU factorization
 Sparse matrices: storage and algorithms

```
        Int loc=0;
```

 Iterative methods: basic concepts and available methods

 Collectives as building blocks; complexity

```
        MPI_Win_lock(MPI_LOCK_EXCLUSIVE,repository,0,the_window);
```

 Available methods: LU factorization, parallel LU

 Sparse matrix-vector product

```
        // read the table_element by getting the result from adding zero
```

 Computational aspects of iterative methods

 Parallel LU through nested dissection

 Incomplete approaches: incomplete factorization

 Parallelism and implicit operations: wavefronts, approximation

```
        MPI_Win_unlock(repository,the_window);
```

 N-body problems: naive and equivalent formulations

```
}
```

 Graph analytics, interpretation as sparse matrix problems

 Derived datatypes

 Communicator manipulation

 Non-blocking collectives

 One-sided communication

 Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models
Load balancing, locality, space-filling curves

First we dig into bits
Integers

Floating point numbers
Floating point math
Examples

VarList: list internal variables More
Essential aspects of LU factorization

https://computation.rnd.llnl.gov/mpi_t/varList.php

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product
Sparse matrix-vector product

Latency hiding / communication minimizing
Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems
Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

MPI_T tool interface

● Structure of a modern processor

The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model

● Memory hierarchy: caches, register, TLB.

Interconnects and topologies, theoretical concepts
Programming models

Load balancing, locality, space-filling curves

● Multicore issues

First we dig into bits

Integers

● Programming strategies for performance

Floating point numbers

Floating point math

Examples

● The power question

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Parallelism85

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

● Basic concepts

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

● Theoretical concepts

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

● The SIMD/MIMD/SPMD/SIMT model for parallelism

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

● Characterization of parallelism by memory model

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Programming models

Message passing

Task-based

Object-oriented

Parallel C/C++

OpenMP

Fortran 90/95

OpenCL

OpenACC

OpenMP 4.0

OpenCL 2.0

OpenACC 2.0

OpenMP 4.5

OpenCL 3.0

OpenACC 3.0

OpenMP 5.0

OpenCL 4.0

OpenACC 4.0

OpenMP 6.0

OpenCL 5.0

OpenACC 5.0

OpenMP 7.0

OpenCL 6.0

OpenACC 6.0

OpenMP 8.0

OpenCL 7.0

OpenACC 7.0

OpenMP 9.0

OpenCL 8.0

OpenACC 8.0

OpenMP 10.0

OpenCL 9.0

OpenACC 9.0

OpenMP 11.0

OpenCL 10.0

OpenACC 10.0

OpenMP 12.0

OpenCL 11.0

OpenACC 11.0

OpenMP 13.0

OpenCL 12.0

OpenACC 12.0

OpenMP 14.0

OpenCL 13.0

OpenACC 13.0

OpenMP 15.0

OpenCL 14.0

OpenACC 14.0

OpenMP 16.0

OpenCL 15.0

OpenACC 15.0

OpenMP 17.0

OpenCL 16.0

OpenACC 16.0

OpenMP 18.0

OpenCL 17.0

OpenACC 17.0

OpenMP 19.0

OpenCL 18.0

OpenACC 18.0

OpenMP 20.0

OpenCL 19.0

OpenACC 19.0

OpenMP 21.0

OpenCL 20.0

OpenACC 20.0

OpenMP 22.0

OpenCL 21.0

OpenACC 21.0

OpenMP 23.0

OpenCL 22.0

OpenACC 22.0

OpenMP 24.0

OpenCL 23.0

OpenACC 23.0

OpenMP 25.0

OpenCL 24.0

OpenACC 24.0

OpenMP 26.0

OpenCL 25.0

OpenACC 25.0

OpenMP 27.0

OpenCL 26.0

OpenACC 26.0

OpenMP 28.0

OpenCL 27.0

OpenACC 27.0

OpenMP 29.0

OpenCL 28.0

OpenACC 28.0

OpenMP 30.0

OpenCL 29.0

OpenACC 29.0

OpenMP 31.0

OpenCL 30.0

OpenACC 30.0

OpenMP 32.0

OpenCL 31.0

OpenACC 31.0

OpenMP 33.0

OpenCL 32.0

OpenACC 32.0

OpenMP 34.0

OpenCL 33.0

OpenACC 33.0

OpenMP 35.0

OpenCL 34.0

OpenACC 34.0

OpenMP 36.0

OpenCL 35.0

OpenACC 35.0

OpenMP 37.0

OpenCL 36.0

OpenACC 36.0

OpenMP 38.0

OpenCL 37.0

OpenACC 37.0

OpenMP 39.0

OpenCL 38.0

OpenACC 38.0

OpenMP 40.0

OpenCL 39.0

OpenACC 39.0

OpenMP 41.0

OpenCL 40.0

OpenACC 40.0

OpenMP 42.0

OpenCL 41.0

OpenACC 41.0

OpenMP 43.0

OpenCL 42.0

OpenACC 42.0

OpenMP 44.0

OpenCL 43.0

OpenACC 43.0

OpenMP 45.0

OpenCL 44.0

OpenACC 44.0

OpenMP 46.0

OpenCL 45.0

OpenACC 45.0

OpenMP 47.0

OpenCL 46.0

OpenACC 46.0

OpenMP 48.0

OpenCL 47.0

OpenACC 47.0

OpenMP 49.0

OpenCL 48.0

OpenACC 48.0

OpenMP 50.0

OpenCL 49.0

OpenACC 49.0

OpenMP 51.0

OpenCL 50.0

OpenACC 50.0

OpenMP 52.0

OpenCL 51.0

OpenACC 51.0

OpenMP 53.0

OpenCL 52.0

OpenACC 52.0

OpenMP 54.0

OpenCL 53.0

OpenACC 53.0

OpenMP 55.0

OpenCL 54.0

OpenACC 54.0

OpenMP 56.0

OpenCL 55.0

OpenACC 55.0

OpenMP 57.0

OpenCL 56.0

OpenACC 56.0

OpenMP 58.0

OpenCL 57.0

OpenACC 57.0

OpenMP 59.0

OpenCL 58.0

OpenACC 58.0

OpenMP 60.0

OpenCL 59.0

OpenACC 59.0

OpenMP 61.0

OpenCL 60.0

OpenACC 60.0

OpenMP 62.0

OpenCL 61.0

OpenACC 61.0

OpenMP 63.0

OpenCL 62.0

OpenACC 62.0

OpenMP 64.0

OpenCL 63.0

OpenACC 63.0

OpenMP 65.0

OpenCL 64.0

OpenACC 64.0

OpenMP 66.0

OpenCL 65.0

OpenACC 65.0

OpenMP 67.0

OpenCL 66.0

OpenACC 66.0

OpenMP 68.0

OpenCL 67.0

OpenACC 67.0

OpenMP 69.0

OpenCL 68.0

OpenACC 68.0

OpenMP 70.0

OpenCL 69.0

OpenACC 69.0

OpenMP 71.0

OpenCL 70.0

OpenACC 70.0

OpenMP 72.0

OpenCL 71.0

OpenACC 71.0

OpenMP 73.0

OpenCL 72.0

OpenACC 72.0

OpenMP 74.0

OpenCL 73.0

OpenACC 73.0

OpenMP 75.0

OpenCL 74.0

OpenACC 74.0

OpenMP 76.0

OpenCL 75.0

OpenACC 75.0

OpenMP 77.0

OpenCL 76.0

OpenACC 76.0

OpenMP 78.0

OpenCL 77.0

OpenACC 77.0

OpenMP 79.0

OpenCL 78.0

OpenACC 78.0

OpenMP 80.0

OpenCL 79.0

OpenACC 79.0

OpenMP 81.0

OpenCL 80.0

OpenACC 80.0

OpenMP 82.0

OpenCL 81.0

OpenACC 81.0

OpenMP 83.0

OpenCL 82.0

OpenACC 82.0

OpenMP 84.0

OpenCL 83.0

OpenACC 83.0

OpenMP 85.0

OpenCL 84.0

OpenACC 84.0

OpenMP 86.0

OpenCL 85.0

</div

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves

Profiling and debugging; optimization and programming strategies.

Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

Parallel Debugging

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMD model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Defensive programming

Code review

Memory debugging

Parallel debugging

- Measurements: repeated and controlled

Iterative analysis of sparse matrix-vector product

beware of transients, do you know where your data is?

- Document everything

Collectives as building blocks; complexity

Locality analysis of sparse matrix-vector product

Sparse matrix-vector product

Communication patterns: communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

161 Analysis basics

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD SIMD model
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math

- **Defaults are a starting point**
Examples
More
Essential aspects of LU factorization
Iterative methods, basic concepts and available methods
useful to check if optimization happened / could not happen
- **use reporting options: opt-report, vec-report**
useful to check if optimization happened / could not happen
- **test numerical correctness before/after optimization change**
(there are options for numerical correctness)

Scalability analysis of dense matrix-vector product
Latency hiding / communication minimizing
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Defensive programming
Debugging
Memory debugging
Parallel debugging

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SIMD+MIMD+Other paradigm

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

- **Use libraries when possible: don't reinvent the wheel**

Iterative methods, basic concepts and available methods

Collectives as building blocks: complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Defensive programming

Debugging

Memory debugging

Parallel debugging

- **Premature optimization is the root of all evil (Knuth)**

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

164 Code design for performance

The SIMD, SIMD-SIMT, SIMD-SIMT-
Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Defensive programming

Memory safety

Memory debugging

Parallel Debugging

- Keep inner loops simple: no conditionals, function calls, casts

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

- Avoid small functions; try macros or inlining

Iterative methods; basic concepts and available methods

Collectives as building blocks; complexity

Communication reduction; memory access patterns

Sparse matrix-vector product

Matrix hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

165 Multicore / multithread

The SIMD/MIMD/OMD/ISM model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Defensive programming

Debugging

Performance debugging

Parallel Debugging

- Use `numactl`: prevent process migration

Essential aspects of multicore in linear algebra

Sparse matrices: storage and algorithms

- Iterative methods: basic concepts and available methods

Collectives as building blocks, complexity

Scalability analysis of dense matrix-vector product

Scalability analysis of iterative methods

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

166 Multinode performance

The SIMD/MIMD/SIMD/SIMD/MIMD model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Iterative methods: basic insight

Sparse matrices: storage and algorithms

Iterative methods: basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Defensive programming

Debugging

Memory debugging

Parallel Debugging

- Influenced by load balancing

- Use HPC toolkit, Scalasca, TAU for plotting

- Explore 'eager' limit (mvapich2: environment variables)

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

167 Classes of programming errors

The SW/MINDSHIFT classification for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Defensive programming

Debugging

Memory debugging,

Profiling, Debugging

Logic errors:

functions behave differently from how you thought,
or interact in ways you didn't envision

Iterative methods, basic concepts and available methods

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/AMC/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Defensive programming

Debugging

Memory debugging

Parallel Debugging

Coding errors:

Essential aspects of LU factorization

Parallel LU factorization storage and algorithms

send without receive forget to allocate buffer

Iterative methods, basic concepts and available methods

Scalability analysis of iterative methods using blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding, communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

168 More classes of errors

Debuggers can help

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits

Defensive programming

more
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Debugging
Memory debugging
Parallel Debugging

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/DRAM/Dense matrix computation

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

- **Keep It Simple ('restrict expressivity')**

Defensive programming

Debugging

Memory debugging

Latency hiding / minimization

- **Example: use collective instead of spelling it out**

Iterative methods, basic concepts and available methods

Collectives as building blocks: complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SIMD/MIMD classification
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More

Defensive programming
Debugging
Memory debugging
Parallel Debugging

Beware of memory leaks:
Memory leak detection and analysis
LU factorization
Sparse matrices: storage, and algorithms
Optimal distribution of data objects and available memory
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product

C++ does this automatically with RAII

Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theorem: concerns

The SIMD/MIMD/SPMD/SMT model of parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

Defensive programming

Robustness

Memory debugging

Parallel Debugging

Design for debuggability, also easier to optimize

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks, complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Computational complexity of iterative methods

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

171 Modular design

Separation of concerns; try to keep code aspects separate

Premature optimization is the root of all evil (Knuth)

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MMI/SIMD32 model of parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
ability to reuse the matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Incomplete LU factorization
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication

Defensive programming
Debugging
Memory Debugging
Parallel Debugging

Be aware of latencies: bundle messages

(this may go again separation of concerns)

Consider eager limits
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
ability to reuse the matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Incomplete LU factorization
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms

N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Process placement, reduction in number of processes

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers

Debugging

more
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Debugging
Memory debugging
Parallel Debugging

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Debugging is like being the detective in a crime movie where you are also the murderer. (Filipe Fortes, 2013)

173

Examples

More

Floating point numbers

Normalizing floating point numbers

Defensive programming

Debugging

Memory debugging

Symbolic debugging

What do you do when your program misbehaves?

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Latency hiding / communication minimizing

Communication latency minimization

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavelets, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD SIMD/SIMD/MIMD parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

#include <stdlib.h> Storage and algorithms

Iterative methods, basic concepts and available methods

#include <stdio.h> Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

int main() { Sparse matrix-vector product

 latency hiding / communication minimizing

 Computational aspects of iterative methods

 return 0; Parallel LU through nested dissection

 Incomplete approaches to matrix factorization

}

Parallelism and implicit operations: wavefronts, approximation

 Multicore block algorithms

 N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

 Derived datatypes

 Communicator manipulation

 Non-blocking collectives

 One-sided communication

Profiling and debugging; optimization and programming strategies.

tutorials/gdb/c/hello.c

Defensive programming

Debugging

Memory debugging

Parallel Debugging

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Implementation

175 Simple example: running

The SIMD/MIMD/SPMD/CIMI Model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

%% cc -g -o hello hello.c

Programming models

Loadbalancing, locality, space-filling curves

regular invocation:

First we dig into bits

%% ./hello

Integers

hello world

Floating point numbers

Floating point math

invocation from gdb:

More

%% gdb hello

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

GNU gdb 6.3.50-20050815 #....

[version info]

Defensive programming

Debugging

Memory debugging

Parallel Debugging

Copyright 2004 Free Software Foundation, Inc.

[copyright info]

(gdb) run

Sparse matrix-vector product

Latency hiding / communication minimizing

Starting program: /home/eijkhout/tutorials/gdb/hello

Parallel LU through nested dissection

Reading symbols for shared libraries +. done

Parallelism and implicit operations: wavefronts, approximation

hello world

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Program exited normally.

Derived datatypes

Communicator manipulation

(gdb) quit

Non-blocking collectives

One-sided communication

%%

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theistic concepts

176 Source listing

The SIMD/MIMD/SPMD/SIM model or parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

%% cc -o hello hello.c Examples More

%% gdb hello Essential aspects of LU factorization Sparse matrices: storage and algorithms

GNU gdb 6.3.50-20050815 # Iterative methods, basic concepts and available methods Collectives as building blocks, complexity

(gdb) list Scalability analysis of dense matrix-vector product Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Collective operations: communication

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Defensive programming

Debugging

Memory debugging

Parallel Debugging

..... version info

Important to use the -O compile option!

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

177 Run with arguments

tutorials/gdb/c/say.c

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model
Interconnects and topologies; theoretical concepts

#include <stdlib.h>

Programming models

Load balancing, locality, space-filling curves

#include <stdio.h>

First few digits of pi

int main(int argc, char *argv) {

Integers

int i;

Floating point numbers

for (i=0; i<atoi(argv[1]); i++)

Floating point math

Examples

printf("hello world\n");

More
Essential aspects of LU factorization

return 0;

Sparse matrices: storage and algorithms

}

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

%% gdb say

Latency hiding / communication minimizing

Computational aspects of iterative methods

.... the usual messages ...

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

(gdb) run 2

Multicore block algorithms

Starting program: /home/eijkhout/tutorials/gdb/c/say 2

No debug info found. This may indicate optimization or compilation without debugging symbols.

Reading symbols for shared libraries +. done

hello world

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging: optimization and programming strategies.

Defensive programming

Debugging

Memory debugging

Parallel Debugging

Structure of a modern processor

Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

Performance modeling

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interactions and anomalies, theoretical concepts

Programming models

Load balancing, quality of filling curves

First we dig into bits

Integers

Floating point numbers

Floating point NaN

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

}

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Parallel block algorithms

5000 N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Segmentation fault

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging: optimization and programming strategies

The debugger will stop at the problem.

Eijkhout: HPC intro

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves

179 Stack trace

The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves

First we dig into bits

Displaying a stack trace

Integers
Floating point numbers

Floating point math

gdb

Example
More

lldb

Defensive programming
Debugging

Essential aspects of LU factorization
(gdb) where (lldb) thread backtrace
Sparse matrices, storage and algorithms
Iterative methods, basic concepts and available methods

Memory debugging
Data corruption

Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Lateness hiding / communication minimizing

(gdb) backtrace

Computational aspects of iterative methods

Parallel LU through nested dissection

#0 0x0000071ff824295ca in _svfscanf_l ()

Implicit dependencies in matrix factorization

#1 0x0000071ff8244011b in fscanf ()

Parallelism and implicit operations: wavefronts, approximation

#2 0x00000001000000e89 in main (argc=1, argv=0x7fff5fbfc7c0) at sq

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/PIMD/DS/MPS/MPSA paradigm
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math

Investigate a specific frame

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks; complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Communication patterns and collective methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Defensive programming

Debugging

Memory debugging

Parallel Debugging

Then print variables and such.

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SIMD/SIMD model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating-point numbers

Floating point math

Examples

More

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods

Collectives as building blocks: complexity

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

$s = 0.$; Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

double di = (double)i;

Multidimensional algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpartition sparse matrix problems

Derived datatypes

}

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

181 Out-of-bounds errors

```
// up.c
int nlocal = 100; i;
double s, *array = (double*) malloc(nlocal*sizeof(double))
for (i=0; i<nlocal; i++) {
    double di = (double)i;
    array[i] = 1/(di*di);
}
s = 0.;
```

Defensive programming

Debugging

Memory debugging

Parallel Debugging

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SMP/Multicore Model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Fractional part

Examples

Program received signal EXC_BAD_ACCESS, Could not access mem

Reason: KERN_INVALID_ADDRESS at address: 0x0000000100200000

0x0000000100000f43 in main (argc=1, argv=0x7fff5fbfe2c0) at

15 s += array[i];

Collectives as building blocks, complexity

(gdb) print array

Scalability analysis of dense matrix-vector product

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

(gdb) print i

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation

\$2 = 128608

Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Defensive programming

Debugging

Memory debugging

Parallel Debugging

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
The several concepts
The SIMD/MIMD/SPMD/SIMT model to parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models

Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples

Defensive programming

More
Essential aspects of numerical computation
Sparse matrices: storage and algorithms
Debugging
Memory debugging
Parallel Debugging

Iterative methods, basic concepts and available methods
gdb Collectives as building blocks; complexity

break foo.c:12 lldb Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing

Computational aspects of iterative methods
Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms

N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes

Communicator manipulation
Non-blocking collectives
One-sided communication

Profiling and debugging; optimization and programming strategies.

183 Breakpoints

Set a breakpoint at a line

break foo.c:12

lldb

breakpoint set [-f foo.c] -l 12

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models

Load balancing, locality, space-filling curves
First we dig into bits
Integers

Floating point numbers
Floating point math

Examples

Defensive programming

More

Debugging

Memory debugging
Parallel Debugging

gdb **lldb** meaning
 Essential aspects of LU factorization

Sparse matrices: storage and algorithms

run start a run
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity

cont continue from breakpoint
Scalability analysis of the matrix-vector product

Sparse matrix-vector product

next next statement on same level
Latency hiding, communication minimization

Computational aspects of iterative methods

step next statement, this level or next
Parallel iteration, collective effect

Incomplete approaches to matrix factorization

Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms

N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation
Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

184 Stepping

Stepping through a program

Defensive programming

Debugging

Memory debugging
Parallel Debugging

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers

Memory debugging

more
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Debugging
Memory debugging
Parallel Debugging

185 Program with problems

tutorials/gdb/c/square1.c

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

#include <stdio.h>

Programming models

//codesnippet gdb/square1c

Load balancing, locality, space-filling curves

First we dig into bits

int main(int argc, char **argv) {

Floating point numbers

int nmax, i;

Floating point math

float *squares, sum;

Examples

Essential aspects of LU factorization

Sparse matrices: storage and algorithms

Iterative methods: Jacobi, Gauss-Seidel, available methods

Collectives as building blocks; complexity

Latency analysis, matrix chain product

squares = (float*) malloc(nmax*sizeof(float));

for (i=1; i<=nmax; i++) {

Sparse matrix-vector product

Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Implementation of incomplete (matrix) factorization

Parallelism and implicit operations: wavefronts, approximation

} Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

//codesnippet end Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

Defensive programming

Debugging

Memory debugging

Parallel Debugging

Structure of a modern processor

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

186 Valgrind output

Characterization of parallelism by memory model

Characterization of parallelism by memory model: Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

%% valgrind square1 Integers

==53695== Memcheck, a memory error detector

==53695== [stuff]

10 Essential aspects of LU factorization
Sparse matrices: storage and algorithms

Iterative methods, basic concepts and available methods
—53695—Collective building tools complexity

--E368E-- Step 0 to 1000000FB

Latency hiding / communication minimizing

Address 0x10002
Parallel LU through nested dissection

Parallelism and implicit operations: wavefronts, approximation
 Parallelism and implicit operations: wavefronts, approximation

==53695== by Multi-Precision Algorithms
 N-body problems: naive and equivalent formulations

==5.3695==
Graph analytics, interpretation as sparse matrix problems
Derived datasets

Communicator manipulation

-blocking collectives

One-sided communication

programming strategies.

Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
Theoretical concepts
The SIMD/MIMD/SPMD/SIMT model for parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers

Parallel Debugging

more
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks; complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

Debugging
Memory debugging
Parallel Debugging

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

The power concepts

The SIMD/MIMD/SPMD/SIMT model for parallelism

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

Load balancing, locality, space-filling curves

First we dig into bits

Integers

Floating point numbers

Floating point math

Examples

More

- Interactive use of gdb, starting up multiple xterms

Essential aspects of parallel factorization
Sparse matrices: storage and algorithms

Iterative methods: basic concepts and available methods
Collectives as building blocks; complexity

- Use gdb to inspect dump:

Sparse matrix vector product
Latency hiding / communication minimizing

Computational aspects of iterative methods

Parallel LU through nested dissection

Incomplete approaches to matrix factorization

Parallelism and implicit dependencies: wavefronts, approximation
Multicore block algorithms

N-body problems: naive and equivalent formulations

Graph analytics, interpretation as sparse matrix problems

Derived datatypes

Communicator manipulation

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.

187 Debugging

I assume you know about gdb and valgrind...

Defensive programming

Debugging

Performance modeling

Parallel Debugging

Note: compile options -g -O0

188 Parallel debuggers

Allinea DDT 4.2-34404

File View Control Search Tools Window Help

Current Group: All Focus on current: Group Process Thread Step Threads Together

All 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Create Group

Project Files

Search (Ctrl+K)

Application Code / Sources problem1.c External Code

problem1.c

```

18     MPI_Finalize();
19     MPI_Barrier(comm);
20 }
21
22
23 int main(int argc,char **argv) {
24     MPI_Comm comm;
25
26     MPI_Init(&argc,&argv);
27     comm = MPI_COMM_WORLD;
28
29     loop_for_awhile(comm);
30
31     MPI_Finalize();
32     return 0;
33 }
34

```

Locals Current Line(s) Current Stack

Variable Name	Value
argc	1
argv	0x7fffffff7958

Type: none selected

Input/Output Breakpoints Watchpoints Stacks Tracepoints Tracepoint Output Logbook

Stacks

Processes Function

#16 main (problem1.c:26)

Evaluate

Expression Value

Profilin

Ready

Structure of a modern processor
Memory hierarchy: caches, register, TLB.
Multicore issues
Programming strategies for performance
The power question
Basic concepts
The general concepts
The SIMD/MIMD/SPMD/SIMT model of parallelism
Characterization of parallelism by memory model
Interconnects and topologies, theoretical concepts
Programming models
Load balancing, locality, space-filling curves
First we dig into bits
Integers
Floating point numbers
Floating point math
Examples
More
Essential aspects of LU factorization
Sparse matrices: storage and algorithms
Iterative methods, basic concepts and available methods
Collectives as building blocks: complexity
Scalability analysis of dense matrix-vector product
Sparse matrix-vector product
Latency hiding / communication minimizing
Computational aspects of iterative methods
Parallel LU through nested dissection
Incomplete approaches to matrix factorization
Parallelism and implicit operations: wavefronts, approximation
Multicore block algorithms
N-body problems: naive and equivalent formulations
Graph analytics, interpretation as sparse matrix problems
Derived datatypes
Communicator manipulation
Non-blocking collectives
One-sided communication
Profiling and debugging; optimization and programming strategies.

189 Buggy code

190 Parallel inspection

File View Control Search Tools Window Help

Current Group: All Focus on current: Group Process Thread Step Threads Together

All 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Create Group

Project Files

Search (Ctrl+K)

Application Code / Sources problem1.c / loop_for_awh main(int argc) External Code

```

10 // Initialize the random number generator
11 srand((int)(mytid*(double)RAND_MAX/ntids));
12
13 for (it=0; ; it++) {
14     double randomnumber = ntids * ( rand() / (double)RAND_MAX );
15     printf("[%d] iteration %d, random %e\n", mytid, it, randomnumber);
16     if (randomnumber>mytid && randomnumber<mytid+1./ntids)
17         MPI_Finalize();
18     MPI_Barrier(comm);
19 }
20
21
22
23 int main(int argc,char **argv) {
24     MPI_Comm comm;
25
26     MPI_Init(&argc,&argv);
27     comm = MPI_COMM_WORLD;

```

Locals

Variable Name	Value
-comm	1140850688
-it	31
-mytid	1
-ntids	16
-randomnumber	1.056087621

Input/Output* Breakpoints Watchpoints Stacks Tracepoints Tracepoint Output Logbook Evaluate Expression Value

Stacks

Processes	Function
16	main (problem1.c:29)
1	loop for awhile (problem1.c:18)
15	loop_for_awhile (problem1.c:19)
15	MPI_Barrier (barrier.c:411)
15	MPIR_Barrier_Impl (barrier.c:266)
15	MPIR_Barrier_MV2 (barrier_osu.c:198)
15	MPIR_Barrier_intra_MV2 (barrier_osu.c:166)
1	MPIR_shmem_barrier_MV2 (barrier_osu.c:104)
1	MPIIDI_CH3I_SHMEM_COLL_Barrier_gather (ch3_shmem_coll.c:940)
1	MPIIDI_CH3I_Progress_test (ch3_progress.c:471)
1	MPIIDI_CH3I_SMP_read_progress (ch3_smp_progress.c:743)
1	MPIIDI_CH3I_SMP_pull_header (ch3_smp_progress.c:4345)
14	MPIR_shmem_barrier_MV2 (barrier_osu.c:113)

Profile Ready

Structure of a modern processor
 Memory hierarchy: caches, register, TLB.
 Multicore issues
 Programming strategies for performance
 The power question
 Basic concepts
 Theoretical concepts
 The SIMD/MIMD/SPMD/SIMT model for parallelism
 Characterization of parallelism by memory model
 Interconnects and topologies, theoretical concepts
 Programming models
 Load balancing, locality, space-filling curves
 First we dig into bits.

191 Stack trace

Stacks	
Processes	Function
16	main (problem1.c:29)
1	+ loop_for_awhile (problem1.c:18)
15	loop_for_awhile (problem1.c:19)
15	MPIR_Barrier (barrier.c:411)
15	MPIR_Barrier_Impl (barrier.c:266)
15	MPIR_Barrier_MV2 (barrier_osu.c:198)
15	MPIR_Barrier_intra_MV2 (barrier_osu.c:166)
1	MPIR_shmem_barrier_MV2 (barrier_osu.c:104)
1	MPIIDI_CH3I_SHMEM_COLL_Barrier_gather (ch3_shmem_coll.c:940)
1	MPIIDI_CH3I_Progress_test (ch3_progress.c:471)
1	MPIIDI_CH3I_SMP_read_progress (ch3_smp_progress.c:743)
1	MPIIDI_CH3I_SMP_pull_header (ch3_smp_progress.c:4345)
14	+ MPIR_shmem_barrier_MV2 (barrier_osu.c:113)

Graph analytics, interpretation as sparse matrix problems
 Derived datatypes
 Communicator manipulation
 Non-blocking collectives
 One-sided communication
 Profiling and debugging; optimization and programming strategies.

Structure of a modern processor
Memory hierarchy: caches, register, TLB.

Multicore issues

Programming strategies for performance

The power question

Basic concepts

Theoretical concepts

The SIMD/MIMD/SPMD model, memory models

Characterization of parallelism by memory model

Interconnects and topologies, theoretical concepts

Programming models

192 Variable inspection

Locals	Current Line(s)	Current Stack
Locals		
Variable Name		Value
comm		— 1140850688
it		— 31
mytid		↗ 1
ntids		— 16
randomnumber		⌚ 1.056087621

Non-blocking collectives

One-sided communication

Profiling and debugging; optimization and programming strategies.