Designing and Building Applications for Extreme Scale Systems

Learn how to design and implement applications for extreme scale systems, including analyzing and understanding the performance of applications, the primary causes of poor performance and scalability, and how both the choice of algorithm and programming system impact achievable performance.The course covers multi-and many-core processors, interconnects in HPC systems, parallel I/O, and the impact of faults on program and algorithm design.

Sessions

Note: Each lecture was recorded; unfortunately, the college of Engineering, which had possesion of the videos, no longer makes them available and they may be lost.

Introduction

- Designing and Building Applications for Extreme Scale Systems CS598: Introduction to the Class

Performance Modeling and Extreme Scale Systems

Covers what extreme scale systems are today and the scale of problems they are used to solve. Introduces basic performance modeling and its use to gain insight into the performance of applications.

- Basic Performance Models For Extreme Scale Systems
- Example of clock timing granularity (on Blue Waters)
- Code for clock granularity: clockgran.c

Benchmarking and Sparse Matrix Vector Multiply

This session introduces some of the most important HPC benchmarks.

The second presentation analyzes a sparse matrix-vector multiply and shows how a simple performance model gives insight into application performance.

- Benchmarks
- Modeling Sparse Matrix-Vector Multiply
- Sparse-matrix vector multiply code

Cache Memory and Performance

This session adds cache memory to the performance model. The second lecture discusses some of the challenges in measuring performance.

- More on Cache Memory
- Measuring Performance
- Reading: When Prefetching Works, When it Doesn't, and Why
- Cache Oblivious matrix transpose code

Spatial Locality

This session discusses the importance of spatial locality and how to model the performance of a simple cache memory system.

- Matrix Transpose
- Automating Code Tuning: An Example with Transpose

The Cache Oblivious Approach

This session discusses a way to think about caches and developing algorithms and implementations which is (mostly) independent of the specific details of the cache. The second presentation covers some other features of caches that can impact performance.

- The Cache Oblivious Approach
- More on Caches

More on How the Processor Works

This session discusses a simple execution model and introduces the issue of aliasing of memory.

The second lecture revisits the dense matrix-matrix multiply operation using the ideas developed in this class.

- Processing Instructions
- Matrix-Matrix Multiply

Instruction Execution and Pipelining

This session covers how modern processors execute instructions in a series of steps and how it affects performance.

- Instruction Execution and Pipelining
- Reading: Exploring performance and power properties of modern multi-core chips via simple machine models

Vectors

This session introduces vectors in modern processors and discusses some of their features and challenges.

- Vectors
- Vectorization Test ProgramsThis is a large collection of test programs for vectorization. There is a autoconf-style configure file and a Makefile that contains vectorization options for *some* compilers.
- Vectorization results on Blue Waters
- Spreadsheet of vectorization results

Moore's Law and Speedup

This session discusses Moore's Law, Dennard Scaling, and some limits on speedup of applications.

- Discussing Speedup
- Moore's Law and Dennard Scaling

Threads

This section introduces threads and thread parallelism, including some advantages and disadvantages of shared memory.

- Threads
- Reading: You Don't Know Jack about Shared Variables or Memory Models

OpenMP Basics

This session introduces OpenMP, the most widely used approach for threaded programming for scientific applications, and how to use OpenMP to parallelize loops.

- OpenMP Basics
- Reading: 32 OpenMP Traps for C++ Developers. Many of these also apply to C.

OpenMP and MAXLOC

This session discusses critical sections and atomic operations in loops in OpenMP, with an emphasis on understanding the performance implications of different approaches.

- OpenMP and MAXLOC

OpenMP and General Synchronization

This session discusses more general loop parallelism in OpenMP, using a linked list as an example in discussing thread locks and performance.

- OpenMP and General Synchronization

Distributed Memory Parallelism

An overview of parallelism, from single processors to massively parallel distributed memory systems. Interconnects and bisection bandwidth.

- Distributed Memory Parallelism

Parallel Programming Models and Systems

Illustrates different ways to program the same simple loop, using different parallel programming systems. Emphasizes the role of the data decomposition in expressing parallelism.

- Parallel Programming Models for Scientific Computing

MPI Basics

An introduction to the Message Passing Interface (MPI), including simple send and receive.

- MPI Basics
- Reading: Reproducible MPI Micro-Benchmarking Isn't As Easy As You Think

MPI Point to Point Communication

Understanding MPI point to point communication, including the effects of the MPI implementation

- More on Point-to-Point Communication
- Buffering and Message Protocols
- Ping-pong Performance on Blue Waters

Strategies for Designing Parallel Applications

An introduction to several ways to think about the design of a parallel application, with a halo exchange as an example

- Strategies for Parallelism and Halo Exchange
- Comparison of blocking and nonblocking pingpong

Performance Models

- Performance Models for Distributed Memory Parallel Computing
- Least Squares for Performance Modeling

More on Efficient Halo Exchange

This session uses halo exchange to explain the importance of deferring synchronization. It also discusses the use of MPI datatypes to avoid extra memory motion.

- Halo Exchange and Contention
- MPI Datatypes

MPI Process Topologies

This session discusses virtual and actual (physical) topologies and how to use MPI to influence the layout of processes on the parallel computer

- Process Topology and MPI
- Enhancing the Communication Performance Models for SMPs

Collective Communication in MPI

This session discusses the rich collective communication and computation features available in MPI

- Collective Communication and Computation in MPI

More on Collective Communication

This session discusses some performance considerations of collective operations, including an example of when using an optimal collective routine gives poorer performance than simpler choices for communication

- Considerations When Using Collective Operations

Introduction to Parallel I/O

This session introduces some of the issues in effectively using I/O with massively parallel computers.

- Introduction to Parallel I/O

Introduction to MPI I/O

An introduction to the parallel I/O features provided by MPI and how they compare to POSIX I/O.

- Introduction to MPI I/O

More on MPI I/O Performance

This session covers more on MPI I/O, including performance optimizations.

- More on MPI I/O
- ioda.c

One-Sided Communication in MPI

This session introduces a different model of parallel computing, one-sided communication, and describes how this model is represented in MPI.

- One-sided Communication in MPI

More on One-Sided Communication

This session covers passive target synchronization and understanding the MPI RMA memory models.

- More on One-sided Communication
- Example program ga_mpi_ddt_rma.c

MPI, Hybrid Programming, and Shared Memory

An introduction to using the MPI+X hybrid programming approach. MPI with threads and the different thread levels are covered. Also introduced is the MPI-3 shared-memory feature, sometimes called the MPI+MPI programming approach.

- MPI, Hybrid Programming, and Shared Memory

New Features of MPI-3

This session provides a summary of the new features in MPI-3.

- New Features of MPI-3