Parallel Computing Workshop

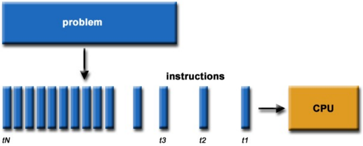
Adam Larios

University of Nebraska Lincoln

26 Feb 2015

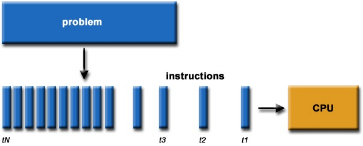
What is parallel computing?

Serial Computing

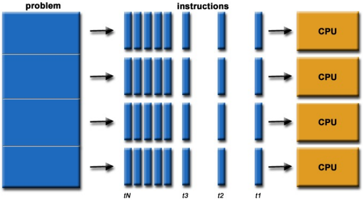


What is parallel computing?

Serial Computing



Parallel Computing



What is parallel computing?

Major Uses

- Solve many small problems at the same time.

What is parallel computing?

Major Uses

- Solve many small problems at the same time.
- Solve one big problem quickly by breaking it into small pieces.

What is parallel computing?

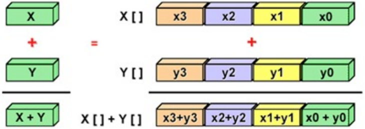
Major Uses

- Solve many small problems at the same time.
- Solve one big problem quickly by breaking it into small pieces.

What is parallel computing?

Major Uses

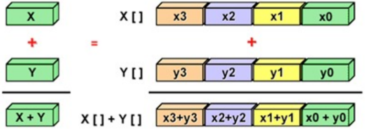
- Solve many small problems at the same time.
- Solve one big problem quickly by breaking it into small pieces.

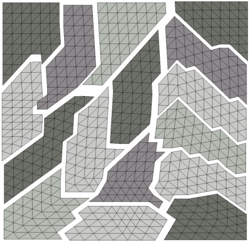


What is parallel computing?

Major Uses

- Solve many small problems at the same time.
- Solve one big problem quickly by breaking it into small pieces.

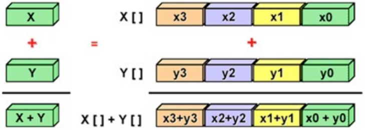


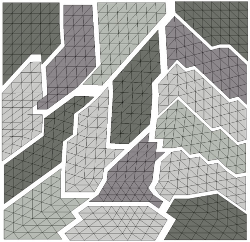


What is parallel computing?

Major Uses

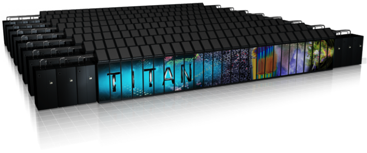
- Solve many small problems at the same time.
- Solve one big problem quickly by breaking it into small pieces.





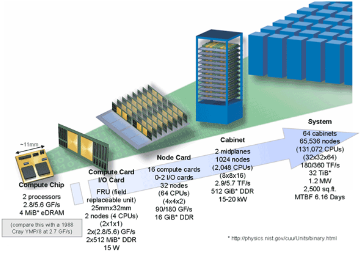
- Repeated associative binary operations (addition, multiplication, maximization)
- Matrix-vector multiplication

Supercomputers

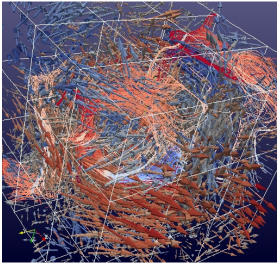


Titan

Blue Gene



Processors Assembly



Simulations

Why do parallel computing?

Good Reasons

- Massive speed up.
- Massive increase in problem size.
- Concurrency (the universe is inherently parallel in space)

Why do parallel computing?

Good Reasons

- Massive speed up.
- Massive increase in problem size.
- Concurrency (the universe is inherently parallel in space)

What is parallel computing?

What is the bad news?

Not as useful when

- Iteration is required.

What is parallel computing?

What is the bad news?

Not as useful when

- Iteration is required.
- Global communication is very frequent.

What is parallel computing?

What is the bad news?

Not as useful when

- Iteration is required.
- Global communication is very frequent.

What is parallel computing?

What is the bad news?

Not as useful when

- Iteration is required.
- Global communication is very frequent.

Difficulties

- Often harder to think about.

What is parallel computing?

What is the bad news?

Not as useful when

- Iteration is required.
- Global communication is very frequent.

Difficulties

- Often harder to think about.
- Often harder to program.

What is parallel computing?

What is the bad news?

Not as useful when

- Iteration is required.
- Global communication is very frequent.

Difficulties

- Often harder to think about.
- Often harder to program.
- More problem dependent.

What is parallel computing?

What is the bad news?

Not as useful when

- Iteration is required.
- Global communication is very frequent.

Difficulties

- Often harder to think about.
- Often harder to program.
- More problem dependent.
- Scalability and overhead costs.

What is parallel computing?

What is the bad news?

Not as useful when

- Iteration is required.
- Global communication is very frequent.

Difficulties

- Often harder to think about.
- Often harder to program.
- More problem dependent.
- Scalability and overhead costs.

What is parallel computing?

What is the bad news?

Not as useful when

- Iteration is required.
- Global communication is very frequent.

Difficulties

- Often harder to think about.
- Often harder to program.
- More problem dependent.
- Scalability and overhead costs.

Is it right for you?

- It depends...

What is parallel computing?

What is the bad news?

Not as useful when

- Iteration is required.
- Global communication is very frequent.

Difficulties

- Often harder to think about.
- Often harder to program.
- More problem dependent.
- Scalability and overhead costs.

Is it right for you?

- It depends...
- ... but probably yes!

What is parallel computing?

What is the bad news?

Not as useful when

- Iteration is required.
- Global communication is very frequent.

Difficulties

- Often harder to think about.
- Often harder to program.
- More problem dependent.
- Scalability and overhead costs.

Is it right for you?

- It depends...
- ... but probably yes!
- E.g., do you have even one parameter in your problem?

How do you access parallel resources?

Parallel Machines

- Your own multi-core laptop or desktop

How do you access parallel resources?

Parallel Machines

- Your own multi-core laptop or desktop
- A cluster, like at the HCC

How do you access parallel resources?

Parallel Machines

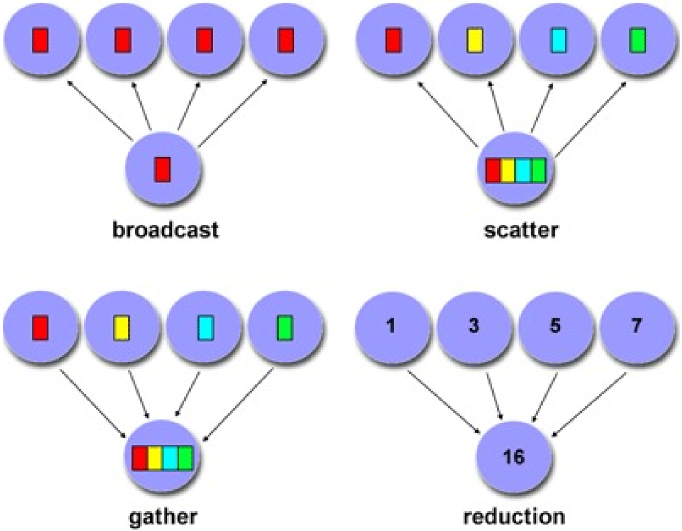
- Your own multi-core laptop or desktop
- A cluster, like at the HCC
- “Cloud computing”, Amazon HPC, etc.

How do you access parallel resources?

Parallel Machines

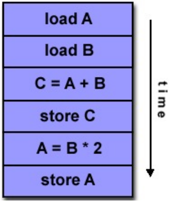
- Your own multi-core laptop or desktop
- A cluster, like at the HCC
- “Cloud computing”, Amazon HPC, etc.
- Open Science Grid

How does it work?



How does it work?

Serial Logic



load A

load B

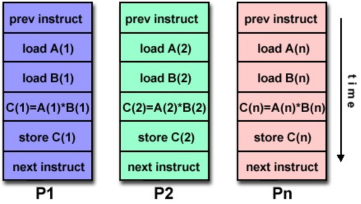
C = A + B

store C

A = B * 2

store A

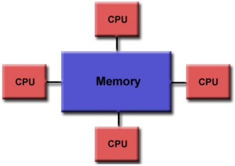
Parallel Logic



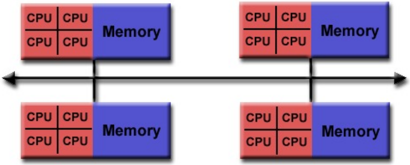
P1	P2	Pn
prev instruct	prev instruct	prev instruct
load A(1)	load A(2)	load A(n)
load B(1)	load B(2)	load B(n)
C(1)=A(1)*B(1)	C(2)=A(2)*B(2)	C(n)=A(n)*B(n)
store C(1)	store C(2)	store C(n)
next instruct	next instruct	next instruct

Memory

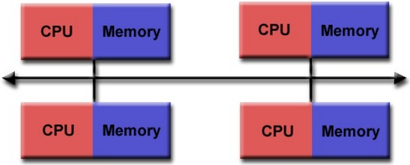
Shared Memory



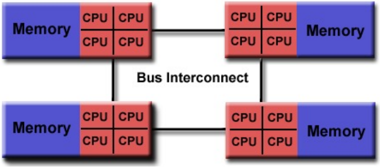
Hybrid Memory



Distributed Memory



Non-Uniform Memory



How can you get started?

What do you need?

- Libraries: OpenMPI, MPICH (Open Source)

How can you get started?

What do you need?

- Libraries: OpenMPI, MPICH (Open Source)
- A parallel compiler (mpif90, mpicc, etc.)

How can you get started?

What do you need?

- Libraries: OpenMPI, MPICH (Open Source)
- A parallel compiler (mpif90, mpicc, etc.)
- Some knowledge of Linux/Unix (probably)

How can you get started?

What do you need?

- Libraries: OpenMPI, MPICH (Open Source)
- A parallel compiler (mpif90, mpicc, etc.)
- Some knowledge of Linux/Unix (probably)

How can you get started?

What do you need?

- Libraries: OpenMPI, MPICH (Open Source)
- A parallel compiler (mpif90, mpicc, etc.)
- Some knowledge of Linux/Unix (probably)

Other notes

- Vendors: MPI, Intel MPI, Scali MP
- Matlab parallel programming

A basic MPI program

```
program who_am_I
  use mpi
  implicit none
  integer :: nb_procs,rank,code

  call MPI_INIT(code)
  call MPI_COMM_SIZE(MPI_COMM_WORLD ,nb_procs,code)
  call MPI_COMM_RANK(MPI_COMM_WORLD ,rank,code)

  print *,'I am the process ',rank,' of ',nb_procs

  call MPI_FINALIZE (code)
end program who_am_I
```

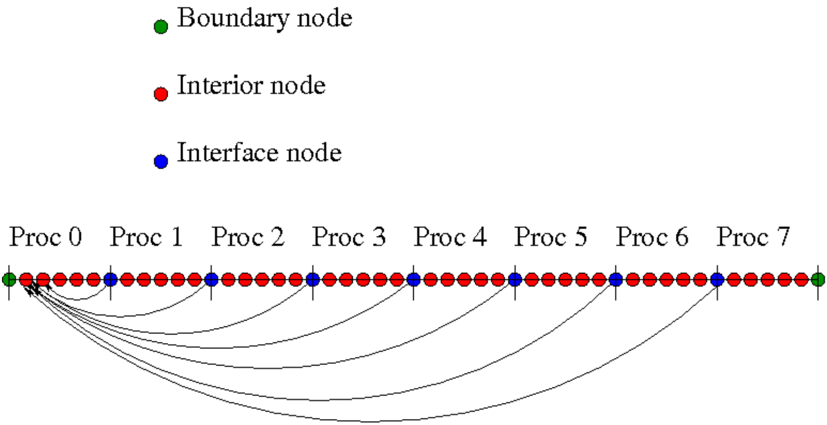
Communications

```
program point_to_point
use mpi
implicit none

integer, dimension( MPI_STATUS_SIZE ) :: status
integer, parameter  :: tag=100
integer  :: rank,value,code

call MPI_INIT(code)
call MPI_COMM_RANK (MPI_COMM_WORLD ,rank,code)

if (rank == 2) then
  value=1000
  call MPI_SEND(value,1,MPI_INTEGER,5,tag,MPI_COMM_WORLD,code)
elseif (rank == 5) then
  call MPI_RECV(value,1,MPI_INTEGER,2,tag,MPI_COMM_WORLD,status,code)
  print *,'I, process 5, have received value ',value,'&
           from process 2'
end if

call MPI_FINALIZE (code)

end program point_to_point
```

Communications

```
> mpiexec -n 7 point_to_point
I, process 5 , have received value 1000 from process 2
```

Communication Overhead

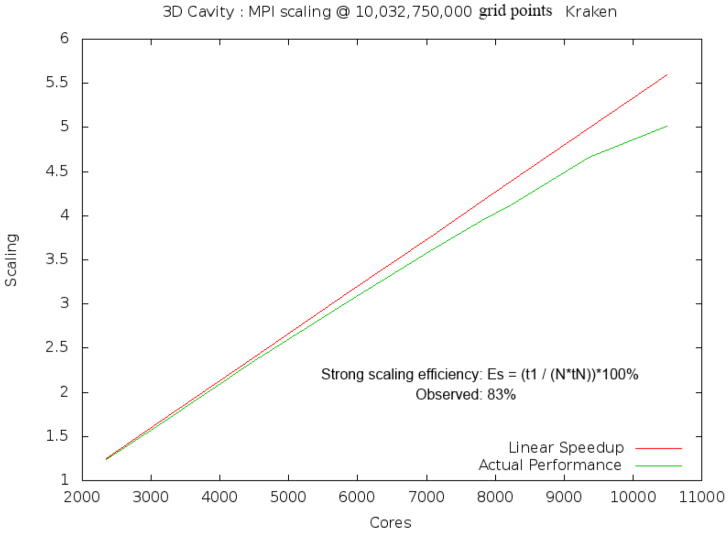


● Boundary node

● Interior node

● Interface node

Scaling Tests



Resources

- “Introduction to Parallel Computing”, Blaise Barney
  `https://computing.llnl.gov/tutorials/parallel_comp`
- “A Parallel Multigrid Tutorial”, Jim Jones
  `https://computing.llnl.gov/casc/linear_solvers/present.html`
- “Introduction to Parallel Computing, Design and Analysis of Algorithms”, Vipin Kumar, Ananth Grama, Anshul Gupta, George Karypis
- IDRIS tutorial with great coding examples:
  `http://www.idris.fr/data/cours/parallel/openmp/`

Thank you, and have fun!