Introduction to Parallel Programming

Professor Seokkoo Kang
Department of Civil & Environmental Engineering
kangsk78@hanyang.ac.kr

Outline

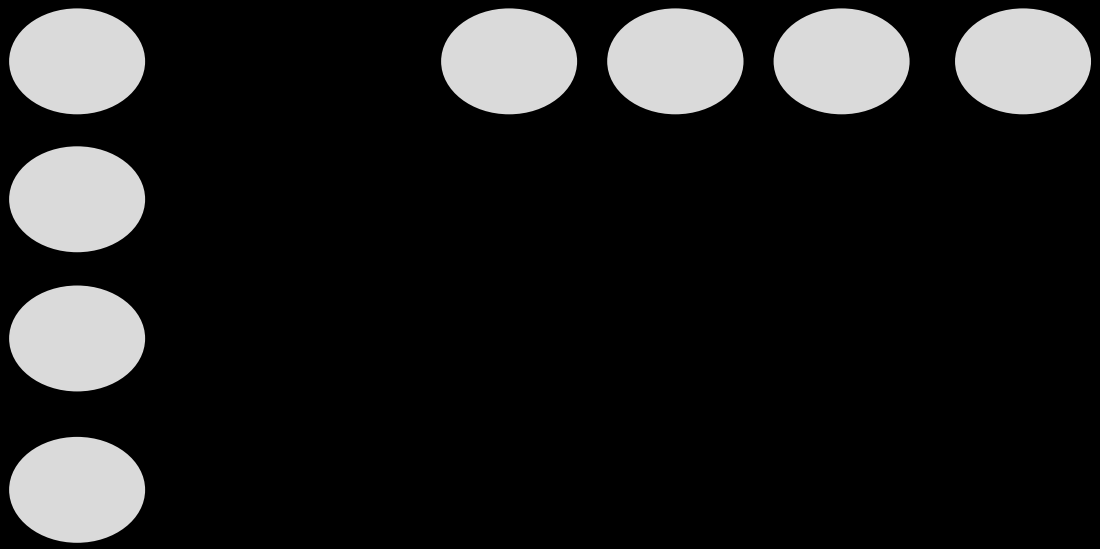
- Introduction to high-performance computing.
- Two types of HPC system
  - Distributed-memory system
  - Shared-memory system
- Introduction to parallel programming.
  - Distributed-memory programming (MPI)
  - Shared-memory programming (OpenMP, GPU)
  - Hybrid programming

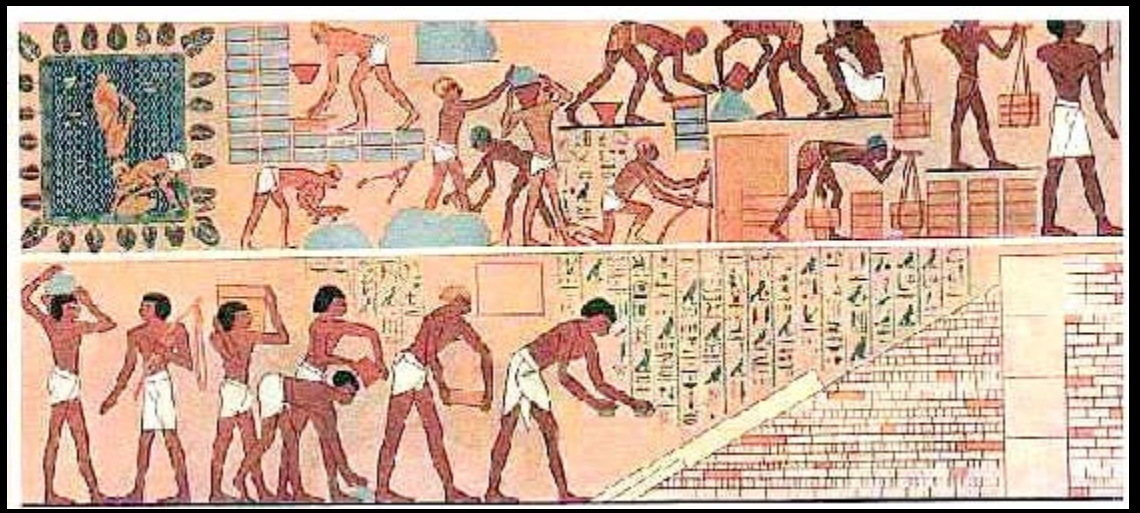
HPC?

- High-Performance Computing
- Use many CPUs to perform massive computations.
- Scientific & engineering calculations
- Fluid mechanics, solid mechanics, weather forecast, …

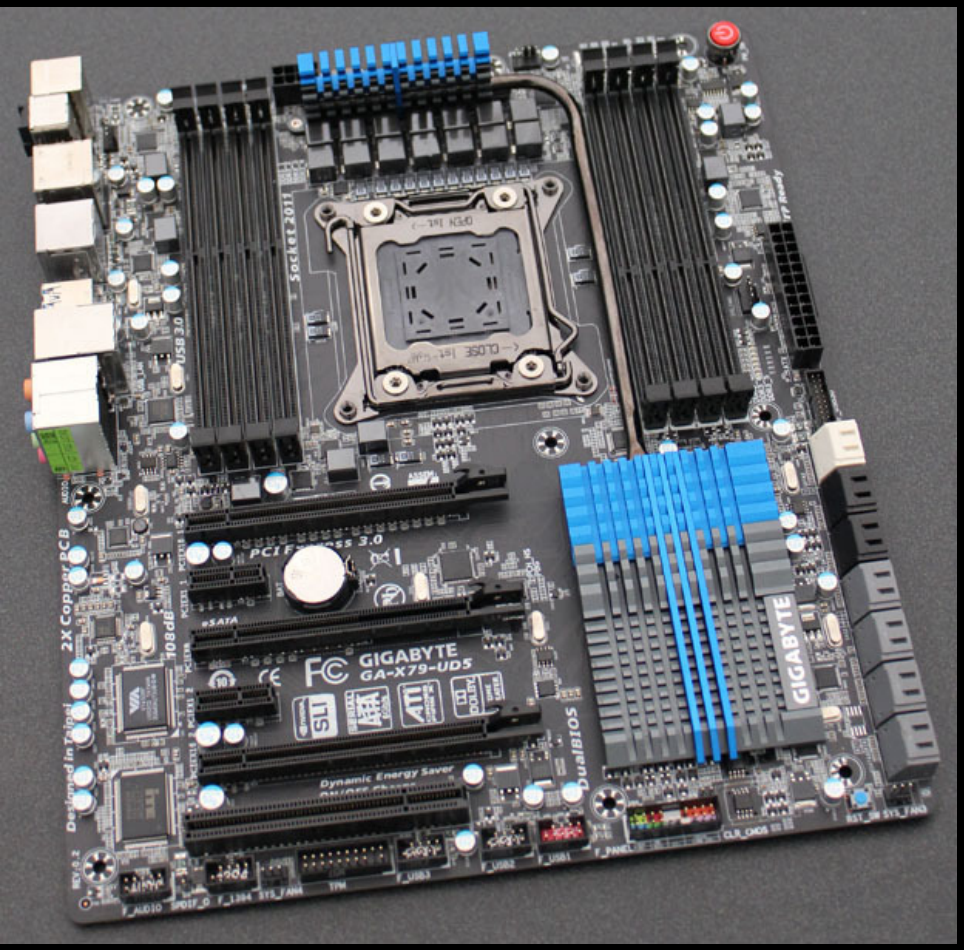
Parallel Computing

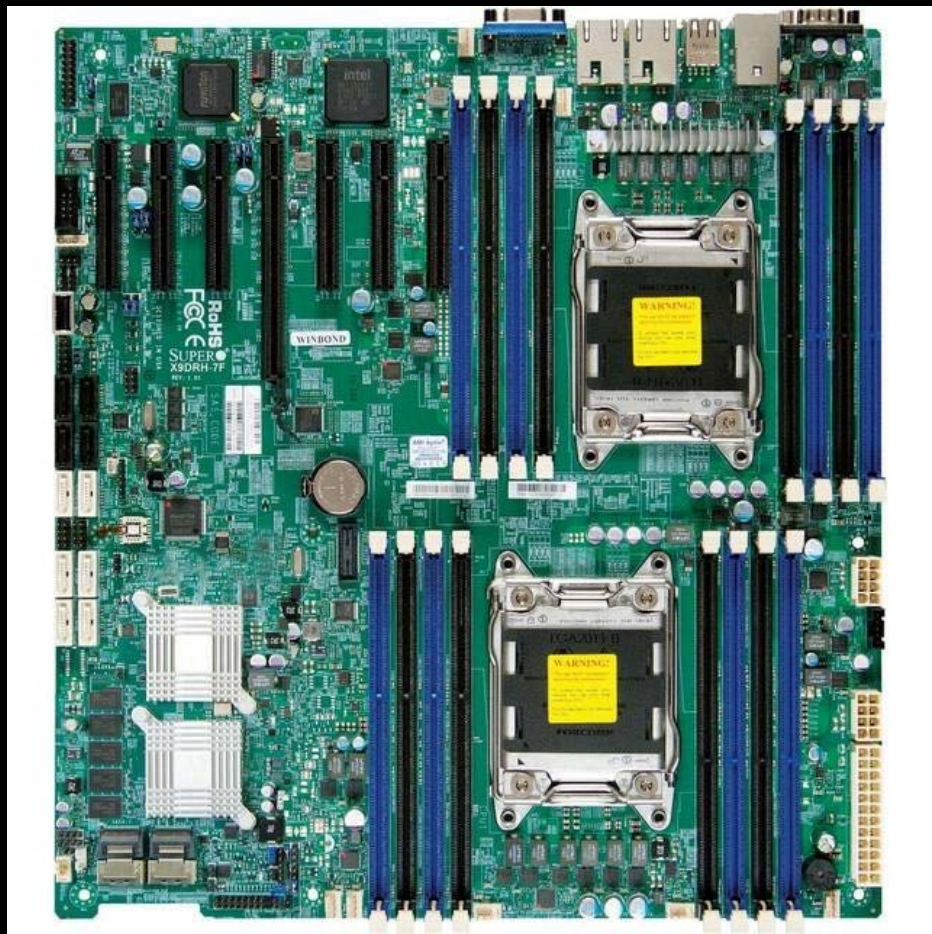
- What is Parallel computing?
- Parallel ←→ ?





PC Architecture





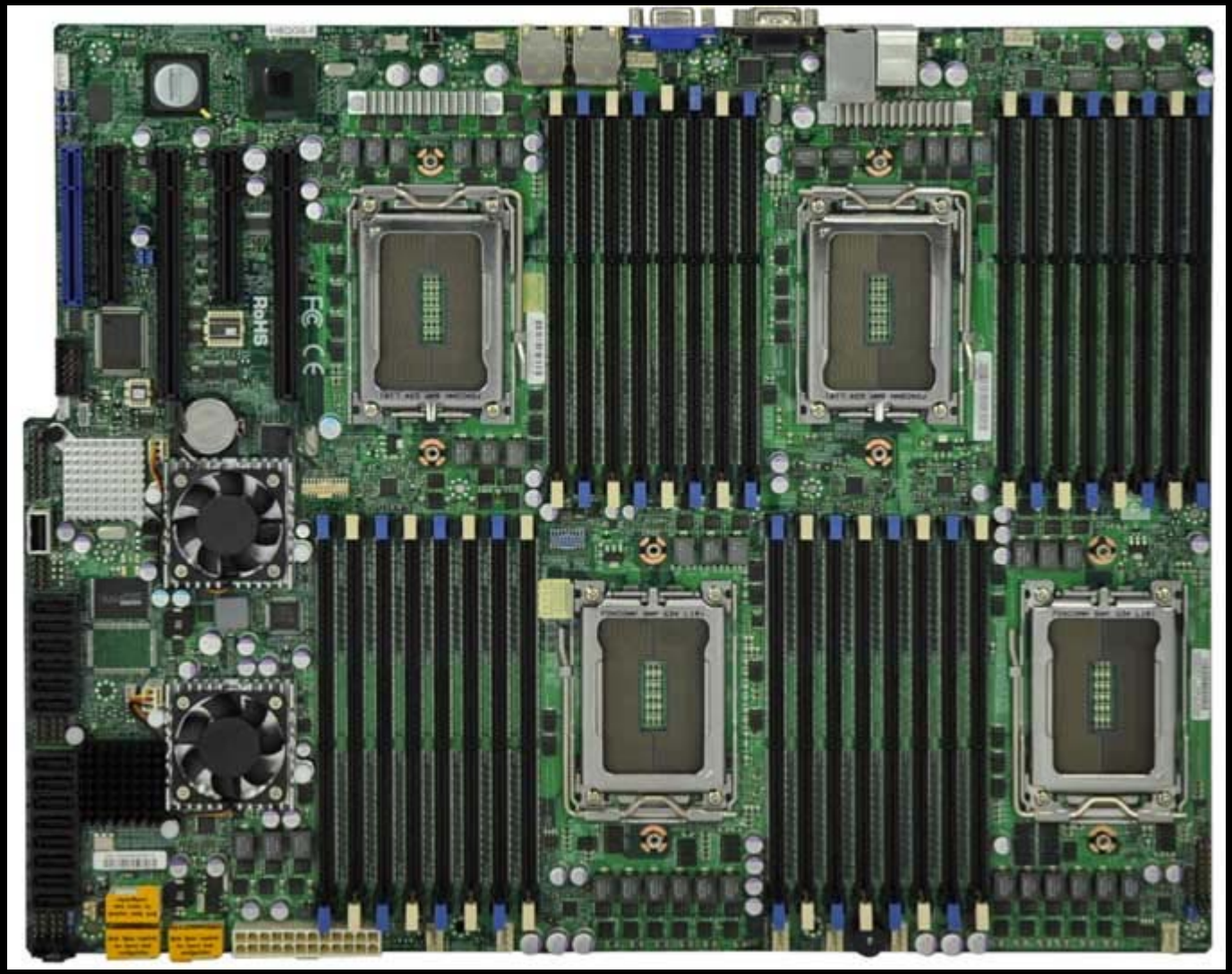
Basic terminology

- Dual-core, Quad-core, …
- Processor (CPU), Core, Node
- Hyper-threading?

Parallel Computer?

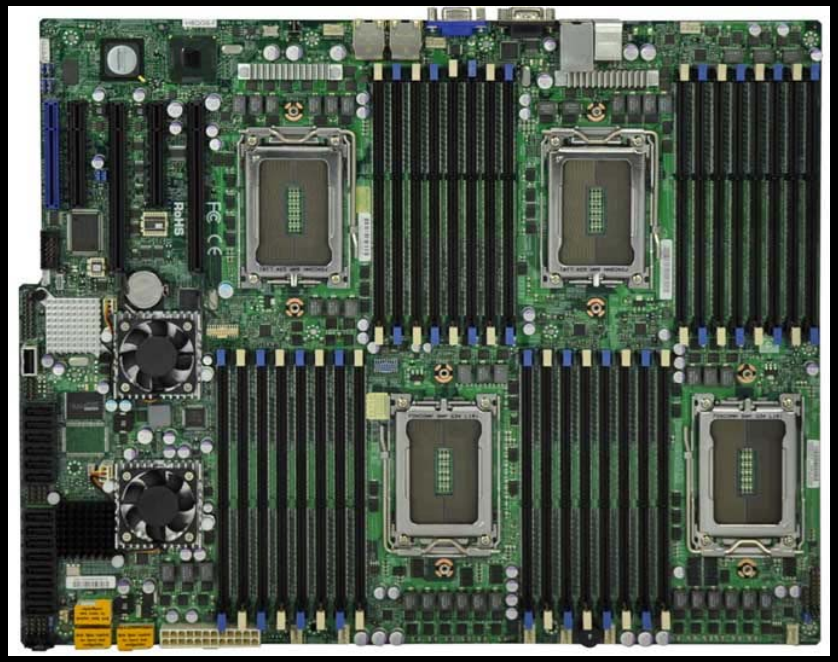
- What is the parallel computer?
  - Many CPUs
- Two ways
  - Increase the number of cores within the node
  - Connect many nodes

Increase the number of cores



Shared-memory system

- 4 CPUs
- 16 cores each CPU
- 1TB DDR3 RAM



- Disadvantage?

Distributed Memory System

Distributed Memory System

node node node

Infiniband cable

Infiniband switch (10/20/40Gbps)

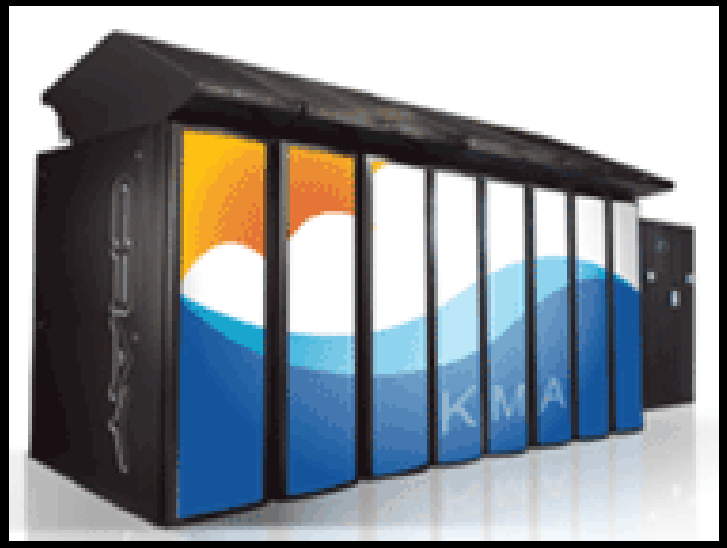
Infiniband

- 100Mbps: High-speed internet (10Mbyte/s)
- 1Gbps: 100Mbyte/s
- 40Gbps: 400 times faster than internet connections (4Gbyte/s)
- Drawback: expensive
  - Switch: 10,000,000 won
  - Card: 1,000,000 won
  - 2m cable: 100,000 won

How large could the cluster be?

Tianhe-2 (China): 3,100,000 cores

BlueGene/Q (USA): 1,500,000 cores



한국 기상청: 90,000 cores

Type of Processors

- Intel CPU
- AMD CPU
- PowerPC (BlueGene): low power. 4096 cores/rack
- GPU

Operating System

- Most (>90%) HPC clusters use Linux as an OS
- Why?
  - Fast
  - Stable
  - No memory limitation
  - User friendly
  - Free (or cheap)
  - Far better than Windows in all aspects

PARALLEL PROGRAMMING

Serial vs. Parallel

- Serial: One core
  - Sequential; One at a time
- Parallel: Multi core
  - Breaking tasks into smaller ones
- Parallel hardware does not guarantee parallel performance

Application

- Material mechanics
- Fluid mechanics
- Weather/climate modeling
- Structural deformation
- Aerodynamics

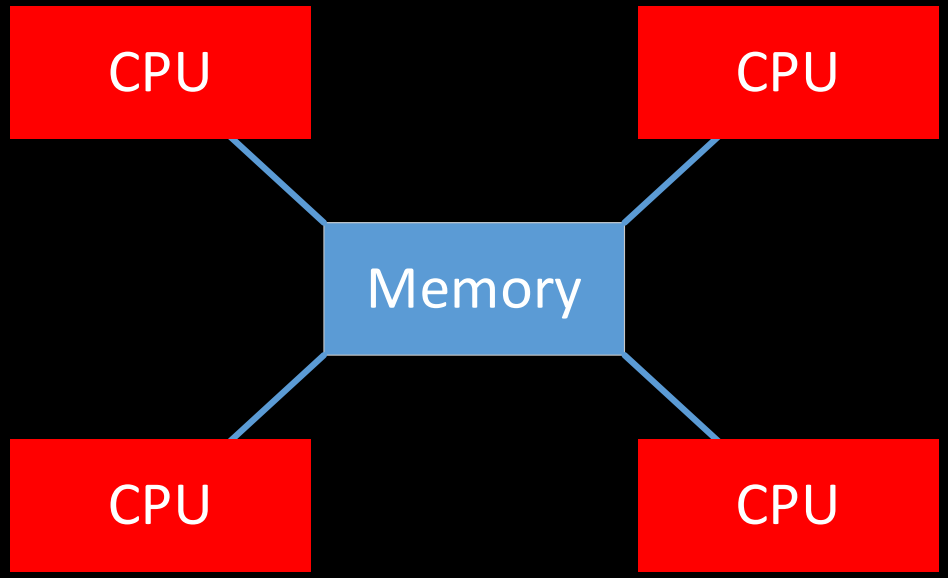
Parallel Programming

- Domain decomposition
- Distributing parts to multiple cores
- Communication among nodes/cores

Parallel Programming Environment

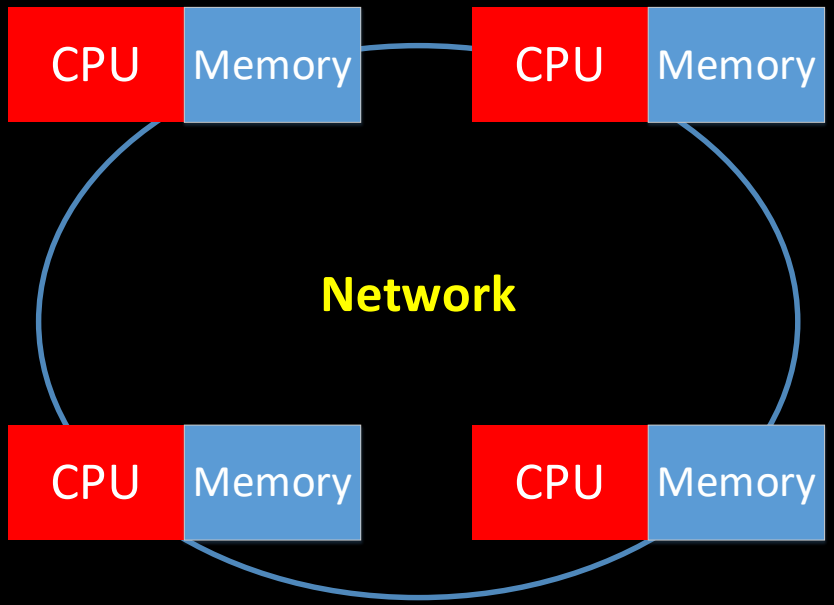
- Distributed memory
  - MPI (Message Passing Interface): CPU Cluster
- Shared memory
  - OpenMP (Open Multi-Processing): CPU
  - CUDA, OpenCL: GPU

Shared Memory



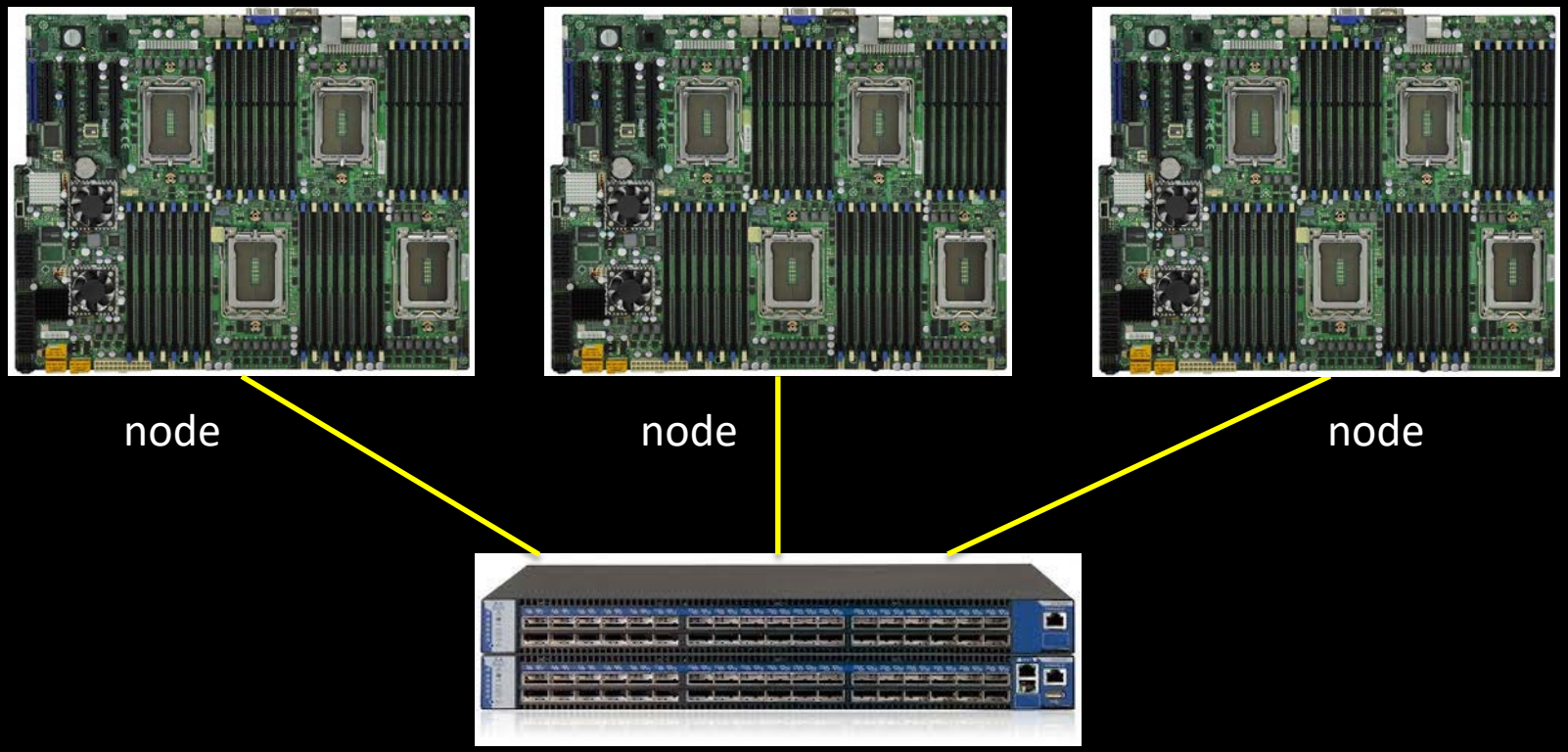
- Advantage
  - Easy to program; Easy to change the existing code
- Disadvantage
  - Memory bandwidth-dependent; Small job size

Distributed Memory



- Advantages
  - Less memory bandwidth dependent (better scaling)
  - Can handle large jobs
- Disadvantages
  - Difficult to change an existing code

Distributed Shared Memory



EXAMPLE PROBLEM

Sum from 1 to 100

- Serial computing

  for(int i=0; i<100; i++) sum += i;

- Parallel computing
  - Shared Memory
  - Distributed Memory
- How to program?

Shared Memory Programming

- No manual domain decomposition.
- Parallelism at the compiler level
- The compiler automatically performs decomposition.

Sample OpenMP code

```
#pragma omp parallel for reduction(+:sum)
for (i=0; i < 100; i++) sum = sum + i;
printf("Total sum=%d\n", total_sum);
```

Distributed Memory Programming

- Domain decomposition
  - 0-10, 11-20, …, 91-100
- Distribute them to each core (MPI function)
  - 0~10: core #1, 11~20: core #2, etc.
- Calculate the partial sum.
- Gather information to get the total sum (MPI function)
- The data are exchanged among many processors through Infiniband cables.

Sample MPI code

```
MPI_Comm_rank (MPI_COMM_WORLD, &rank);
MPI_Comm_size (MPI_COMM_WORLD, &size);
...
my_sum=0;
for (i=n_start; i < n_end; i++) my_sum = my_sum + i;
MPI_Reduce(...)
if(rank==0) printf("Total sum=%d\n", total_sum);
...
```

Distributed Memory Programming

- Requires
  - Good serial programming skill (C, C++)
  - Understanding of computer architecture
  - Understanding of MPI
  - Get familiar with Linux environment

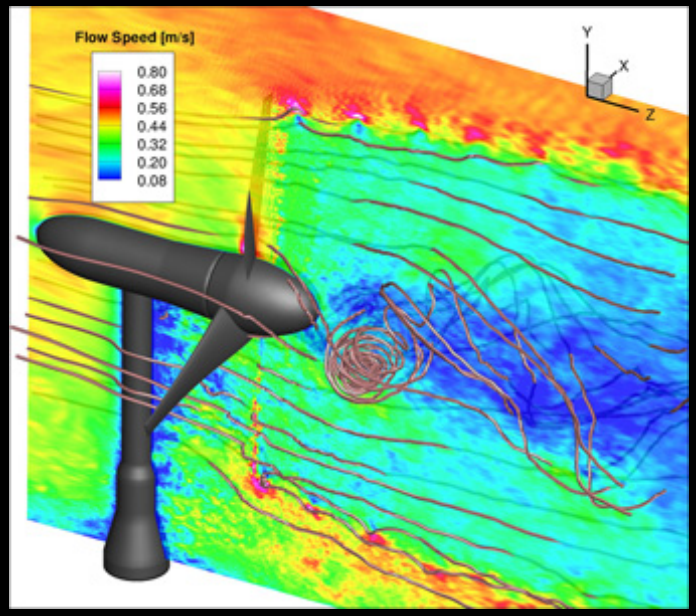
Hybrid Programming

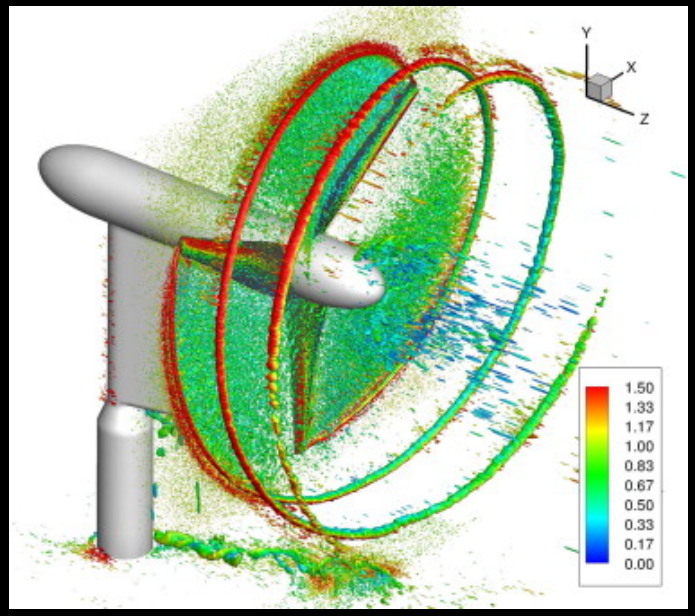
- Scaling issues of a pure MPI code for very large jobs. (core #>10,000)
- Communication overhead > CPU time
- Reduce the overhead by reducing MPI communication.

BlueGene/Q (USA): 1,500,000 cores

HPC Application in Fluid Mechanics

- Flow around a hydro-turbine
- Navier-Stokes equations are solved using 800 cores





- Two weeks of wall clock time
- With a serial computing, it will take 30 years!

Today's Summary

- Parallel computer
  - Shared memory
  - Distributed memory
  - Distributed shared memory
- Parallel programming
  - Message passing interface (MPI)
  - OpenMP

Prerequisites

- Linux
- C Programming
- Compile & Link using GCC

Assignments

- In a group (2 persons) or individually,
  - Install Linux on your own PC
    - CentOS (recommended), Ubuntu, Debian, etc.
  - Install GCC on your own PC
  - Compile and run “hello world” C code
  - Become familiar with Linux
- Prepare a 10-min presentation.
  - Show how you installed Linux.
  - Show how you compiled the “hello world” C code
  - Investigate at least 10 basic commands in Linux

Search “Hello World” in Google

```
#include <stdio.h>


int main ()
{
  printf ("Hello World!\n");
}
```

Class Hour Changes

Mon 9AM-12PM → Mon 10 AM-12PM