Network Science

Lecture 11: August 20, 2009 Lecturer: Xizhou Feng

Scribe: Yifei Ma

High performance computing for Epidemics: Introduction to parallel computing

11.1 Overview

11.1.1 What is Parallel Computing

Traditionally, software has been written for serial computation. The computational problem is broken into a discrete series of instructions and these instructions are executed one by one. On a single computer with a single Central Processing Unit(CPU), at each CPU clock, only one instruction is able to run on the CPU.

In the simplest sense, parallel computing is the simultaneous use of multiple compute resources to solve a computational problem. In contrast to serial computing, the computational problem is broken into discrete parts that can be solved concurrently. Each part if a serial instructions and these parts execute simultaneously on different CPUs.

The compute resource in parallel computing include:

- A single computer with multiple processors
- An arbitrary number of computers connected by a network
- a combination of both

The computational problems able to be computed in parallel usually demonstrate characteristics as:

- Broken apart into discrete pieces of work that can be solved simultaneously
- Execute multiple program instructions at any moment in time
- Solved in less time with multiple compute resources than with a single compute resource

11.1.2 Why use Parallel Computing

- Save time and money.

  In theory, throwing more resources at a task will shorten its time to completion, with potential cost savings. Parallel clusters can be built from cheap, commodity components.

- Solve larger problems.

  Many problems are so large and/or complex that it is impractical or impossible to solve them on a single computer, especially given limited computer memory.

- Provide concurrency.

  A single compute resource can only do one thing at a time. Multiple computing resources can be doing many things simultaneously.

11.1.3 Issues in Parallel & Distributed Computing

- Scalability

  The solutions work well with the problem when the size of the problem increases.

- Performance

  The solutions finish in an acceptable time even for large scale problem.

- Productivity

  The solutions produce good results or provide reasonable services.

11.1.4 Distributed Computing examples

A good example of distributed computing example in our real life is playing football. The coach designs the strategies and instructs his players. And the players execute the strategies in the field, and then in the timeout, they come back to the coach for the next strategies. Analogously, in distributed computing, the head machine divides the whole task into a series of small tasks and distributes these tasks to the working machines. The head machine may also schedule and synchronize the processes on each working machines.

11.2 Parallel Computing Models

11.2.1 Flynn's taxonomy

Instruction	Data
Single Instruction	Single Data
Single Instruction	Multiple Data
Multiple Instructions	Single Data
Multiple Instructions	Multiple Data

11.2.2 SPMD

SPMD (Single Process, Multiple Data) or (Single Program, Multiple Data) is a technique employed to achieve parallelism; it is a subcategory of MIMD. Tasks are split up and run simultaneously on multiple processors with different input in order to obtain results faster. SPMD is the most common style of parallel programming

MPI (Message Passing Interface), a powerful communications protocol used in parallel/distributed computing, is an example of SPMD. It is very common to read MPI code like this:

if (my_rank == r1)

Do bala bala

else if (my_rank == r2)

Do bala bala

Such program will be copied to head machine and each working machines. The if and else statement is to check on which machine this program is running and process the assigned tasks and data for this machine.

11.3 Thread of Computer Architecture

- The size of transistors of clips becomes smaller.
- The frequency of CPU has increased from 200M Hz to 2G Hz
- The Maximum number of cores supported on supercomputer in 1990 was around 1000, and now supercomputer is able to support more than 100,000 cores.
- Power wall and thermal wall are primary issues in current computer architecture design.

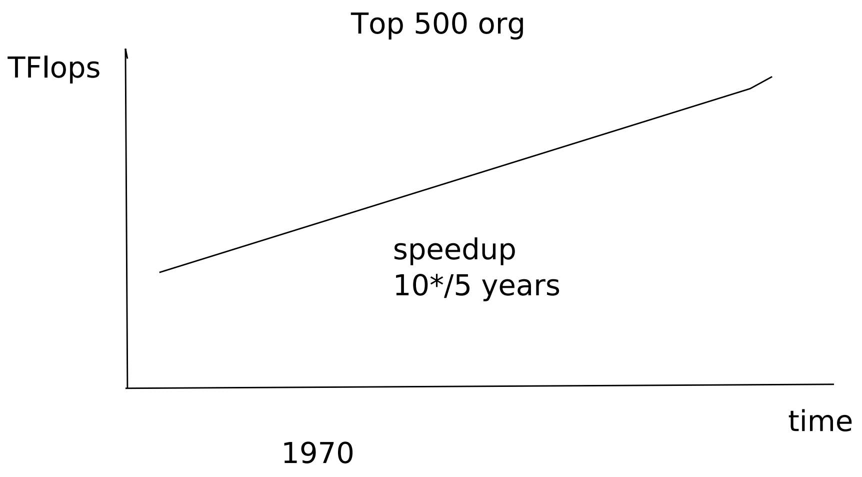

Figure 11.1: Thread of Computer Architecture

11.4 Computer Architecture

Computer Architecture for parallel computing includes multi-computer system (11.2), shared memory system (11.3),message passing system (11.5), PGAS (11.5) and others.

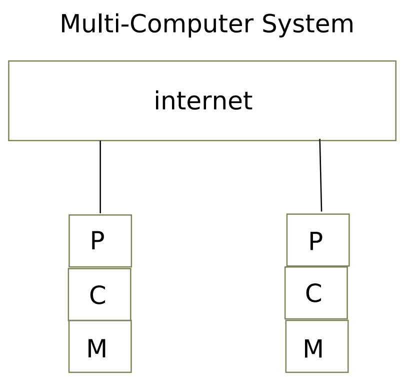

Figure 11.2: Multi-Computer system. P denotes processor, C for cache and M for memory.

11.5 Processor Accelerator

- SIMD (Single Instruction, Multiple Data; colloquially, ”vector instructions”)
- FPGA (Field-Programmable Gate Array)
- GPGPU (General-purpose computing on graphics processing units)

11.6 Parallel Programming Models

- OpenMP (Open Multi-Processing) is a parallel programming interface (API) that supports multi-platform shared memory multiprocessing programming on many architectures.
- Pthread is a POSIX standard for threads. Mostly used for parallel programming in single computer.
- CILK is a programming language designed for multi-threaded parallel computing. The outstanding feature of CILK is it hide thread scheduling and synchronization for the programmer.

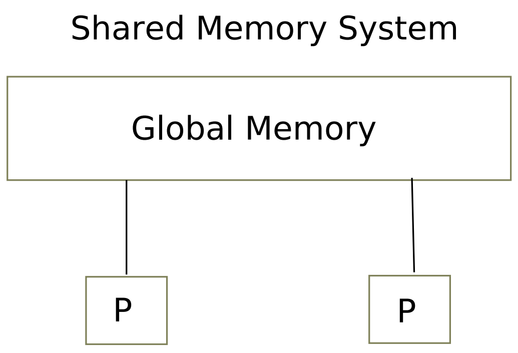

Figure 11.3: Shared Memory system.

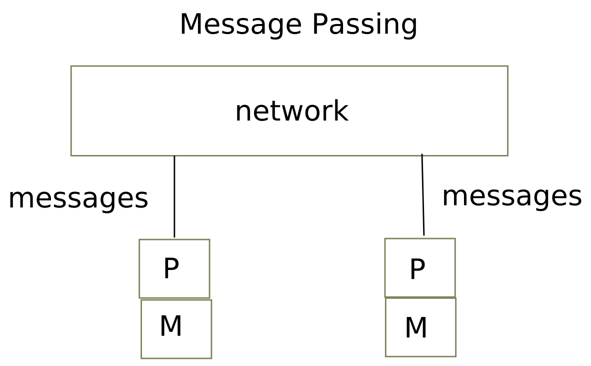

Figure 11.4: Message Passing system.

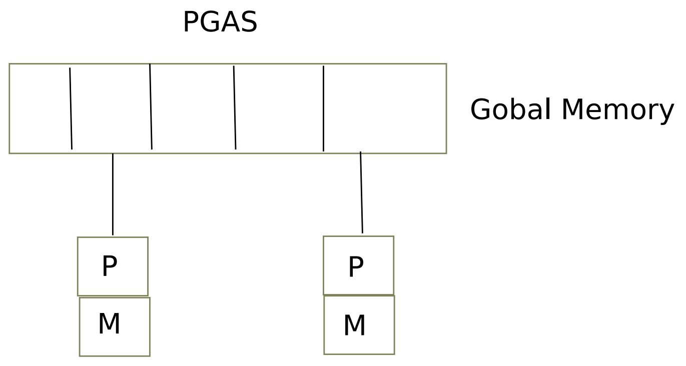

Figure 11.5: Partitioned Global Address Space system. The global memory address space is logically partitioned and a portion of it is local to each processor