

CS8625-May-28-09
First Day of Class
CS8625 High Performance and
Parallel Computing
Dr. Ken Hoganson

Class

Will

Start

Momentarily...

- Observation:
 - Regardless of how much improvement each generation of processor provides, there is a desire for faster processing.
 - Idea: Use parallelism to improve performance using the current generation of technology.
 - (note – it would be nice to have a parallel architecture that allows new technology processors to replace older ones)

Work consists of executable instructions

- Dividing instructions across multiple processing elements → issues
 - How much communication between instructions?
 - Sharing data or results of computation?
 - Independence of blocks of instructions?
 - Data dependencies between instructions.
 - Not every program of instructions can be subdivided for parallel execution.

Parallel Architectures:

- Add parallel HW to take advantage of parallelism at what level?

SW	HW
1. Intra-instruction	Pipeline
2. Inter-instruction	Superscalar
3. Algorithm	Processors
4. Process/Thread	Processors
5. Distributed and Tiered SW	
	Distributed/client-server/Grid

What architecture problems must be solved?

- Interconnections – to share access to data and to pass results between processing elements
 - Performance
 - Contention
 - Scalability
- Topology of the interconnection network or architecture
- My term: Platform Control and Communication Topology (PCCT)

What about SW architecture?

- It determines the pattern of communication between SW components
- Connectivity of components
 - Coupling
 - Cohesion
- The pattern of communication is important – which module talks to which other modules, how often, how much

- My term: Application Communication Topology (ACT)
 - My term for the pattern of SW communication, to which thread/process, how often, how much.
- An ideal parallel computer architecture will provide just as much and no more, connectivity and processing power, as is required by the application, and defined by the ACT.
- Hoganson

- Architectures are rarely created to exactly match one application (though to match a class of applications is valid).
- The closer the match between ACT and PCCT, the better will be the overall performance that is realized.
- So, the process of mapping an application to an architecture that is not ideal for that application, is critical in determining performance.

- To work on high performance and parallel computing systems concepts, we must first understand:
- Computer Architecture
 - Processing elements available at each level and performance enhancements
 - Interconnection technologies
 - Interconnection topologies
- Software architectures
 - Topologies (design approaches)
 - Communication technologies

- We must also understand:
- Performance Boundaries
 - Limitations on performance
 - Constraints and scalability
 - Efficiency in realizing performance
- Software Tools available
 - parallel programming languages and/or systems.

- Goal: tap in to computing power as we tap in to electrical power.
- Electrical Grid, early investigators called this concept the Computer Grid, now called Grid Computing
- Goal: user does not need to know where the computing power is coming from. The system handles many issues “behind the scenes”.
- This goal has not been realized. Not even close (unfortunately).

Grid systems are composed of:

- Many clusters of computers
- Distributed geographically
- Use internet to connect and move data and processing
- Each cluster has multiple machines. Each machine has multiple processors.
- 1000s of processors in the complete system.
- Goal: heterogeneous processors and operating systems.
- Goal not fully realized: May require a particular type of processor. More likely requires a specific OS.

Grid systems are more software than hardware.

- Must have hardware obviously.
- But what makes a system a grid is multiple layers of software:
 - May require a specific OS
 - Requires a set of communication tools
 - Requires a set of user and process validation and certification tools.
 - Requires a set of workload distribution and balancing tools.
 - Requires a “language” for users to initiate and distribute processing

Many applications require:

- Installation of specific software on all machines to be used.
- May require specific libraries for managing and accessing data.
- May require segmentation and pre-positioning of data.
- May require that applications are aware of what SW and data are located on which systems.
- Complex, hard to work with.

Commercial Systems

- Often use Grid Computing as a term to describe their system they would like to sell.
- Is a highly parallel clustered system, often with multiple levels of interconnection.
- Generally expandable (within limits)
- Are used commercially to good effect, for database and engineering applications.
- Are really parallel-distributed-clustered-computing systems.
- No attempt to build a “Computing Grid”.

- Posted syllabus online date/lecture schedule subject to change.
- I will supplement the textbook, but also skip some content.
- Conditional Assignment: Read through Chapter 1 when your books arrive

End
Of
Today's
Lecture.

