

Parallel and Distributed Systems – Introduction

Kai Shen

1/16/2014

CSC 258/458 - Spring 2014

1

General Course Information

- Course Web page
 - <http://www.cs.rochester.edu/~kshen/csc258-spring2014>
- Contact through broadcast emails
- Text and references
- Accounts in computer science research lab
- Assignments/project/presentation/participation

1/16/2014

CSC 258/458 - Spring 2014

2

Parallel Computing

- Parallel computing
 - Process sub-tasks simultaneously so that work can be completed faster.
 - Example applications?

1/16/2014

CSC 258/458 - Spring 2014

3

Parallel Application Examples

- Scientific simulations
 - heat propagation on a surface
 - weather system interaction and evolution
- Google PageRank – sparse matrix computation
- SETI@home
- Internet servers
- Big data analytics
 - User preference mining for product advertisement
 - Hedge fund companies analyze financial records to identify profitable trades
 - Health, medical data mining for enhanced health care

1/16/2014

CSC 258/458 - Spring 2014

4



Parallel & Distributed Computing

- Parallel computing
 - Process sub-tasks simultaneously so that work can be completed faster.
- Distributed computing
 - A set of autonomous computers (connected through network) work together to achieve unified goals.
 - Examples: client/server web transaction, networked file systems, a data center with hundreds of thousands of machines, crowdsourcing among many smartphones, ...
- Mutually exclusive?

1/16/2014

CSC 258/458 - Spring 2014

5



Parallel & Distributed Systems

- Hardware and software systems that support parallel & distributed computing
- Hardware?
 - Parallelism at different hardware levels
 - Fast networks/interconnects for distributed computing
- Operating systems?
 - Facilitate parallel/distributed computing with little overhead
 - Protection and security
- Programming systems?
 - Ease of writing parallel/distributed applications **that work well**

1/16/2014

CSC 258/458 - Spring 2014

6



Why Parallel Computing?

The speed of uniprocessor will catch up with the need of computing?

- Moore's law (1965, with later revisions) – circuit complexity doubles every one year (18 months, 2 years)
- Some argue that it is a self-fulfilling prophecy
- Moore said in 2005:

“In terms of size [of transistors] you can see that we're approaching the size of **atoms** which is a fundamental barrier, ...”

1/16/2014

CSC 258/458 - Spring 2014

7



Why Parallel Computing?

Not always desirable to use the fastest hardware

Example:

- A machine with sixteen 3GHz CPU cores, 32GB memory, and hard drives consumes about 200 Watts
- A machine with four 1GHz CPU cores, 2GB memory, and a CompactFlash drive consumes about 2.5 Watts
- It may be more energy-efficient to use multiple low-power machines in parallel

1/16/2014

CSC 258/458 - Spring 2014

8



Why Parallel Computing?

Multiprocessors are increasingly ubiquitous

- Commodity processors contain multiple computing cores because they are cost-effective to produce
- Traditionally sequential software (e.g., desktop applications) should take advantage of it

1/16/2014

CSC 258/458 - Spring 2014

9



Arguments against Parallel Computing?

- Parallel programming is too difficult
- Performance isn't important

1/16/2014

CSC 258/458 - Spring 2014

10



Why Distributed Systems?

- With advances in networking, including local-area network, Internet, smartphones, and wireless sensor network
 - Distributed computers and people are increasingly connected
- Desired services in networked world:
 - data/information exchange and sharing
 - resource sharing
 - collaboration
 - ...
- Better systems characteristics
 - Scalability
 - Availability
 - Data durability

1/16/2014

CSC 258/458 - Spring 2014

11



Problems with Distributed Systems?

- Distributed systems are too complex, unpredictable
- Security vulnerabilities

1/16/2014

CSC 258/458 - Spring 2014

12

How important are these topics?

- Unscientific poll (Google hits) in 2011
 - "Parallel computing" 2.5M
 - "distributed computing" 12.8M
- Google hits today
 - "Parallel computing" 16.6M
 - "distributed computing" 27.1M

1/16/2014

CSC 258/458 - Spring 2014

13

Parallel Programming

- Challenges:
 - Control dependencies
 - Data sharing
- Parallel algorithms:
 - Identification of parallelisms in applications
 - Design control flow and data sharing mechanism
- Parallel programming models:
 - Shared memory
 - Distributed memory

1/16/2014

CSC 258/458 - Spring 2014

14

Shared Memory Parallel Systems

- In shared memory parallel computers, multiple processors can access the memory simultaneously
 - Examples
- Problems with direct data sharing:
 - One processor's cache may contain a copy of the data that was just modified by another processor
 - ⇒ hardware support for cache coherence
 - Two processes' (semantically) atomic operations may be interleaved
 - ⇒ system support for mutual exclusion

1/16/2014

CSC 258/458 - Spring 2014

15

Distributed Memory Parallel Systems

- Parallel systems that do not share memory
 - Examples
- Less requirement on the system support
 - Little or no hardware support
 - Software system support for communications (point-to-point, group)
- More trouble with writing applications
 - Data must be explicitly partitioned and transferred when needed
 - Hard to do dynamic workload management

1/16/2014

CSC 258/458 - Spring 2014

16



MapReduce Parallel Data Processing

- Scalable, distributed memory parallel systems with an easy programming model
 - But restricted parallelism semantics

1/16/2014

CSC 258/458 - Spring 2014

17



Distributed Computing

- Consistency in distributed systems with failures
 - Models: fail-stop, Byzantine
- Problems:
 - Distributed consensus
 - Replication
- Data durability
- Distributed systems without centralized control (sometimes parties are not trusted)
 - Security and trust
 - Distributed reputation

1/16/2014

CSC 258/458 - Spring 2014

18



Scalable Internet Data Centers

- Large Internet data centers (Google, Microsoft, ...)
 - Thousands of machines clustered together to serve online requests
- A form of parallel computing
 - Often embarrassingly parallel workload, but extremely large scale
 - Customized parallel programming model (MapReduce)
- Distributed system
 - Consistency, reliability, ...
- Additional issues
 - Power management, network topology, error monitoring

1/16/2014

CSC 258/458 - Spring 2014

19