CSCI 251: Concepts of Parallel and Distributed Systems

Alvin Lin

August 28th, 2017

Course Objectives

- Understand parallelism and concurrency
- Recognize inherent parallelism that exists in computational problems.
- Exploit machine parallelism and learn to write efficient and effective parallel programs
- Learn concurrent programming techniques for distributed computing systems
- Learn basics of routing and networking to better utilize parallel and distributed computing resources
- Understand basics of network security and cloud computing

Grading

- Quizzes (20%)
  - 10 quizzes, one every week.
  - The best 8 quizzes will be taken into account with each counting 2.25%.
  - They will be short answer, true/false, or multiple choice.
  - May have negative grading to discourage guessing.
- Projects (40%)

  - 4 projects
  - 3 programming and one problem solving
  - Each project is worth 10%

- Exams (40%)
  - 2 Exams (midterm and final)
  - Each exam is worth 20%

- Homeworks (not graded directly)
  - Impacts performance on quizzes and exams
  - Discussion in MyCourses

Organization

- Topics
  - Main Topics: parallelism, concurrency, parallel computer systems, distributed computer systems, parallel programming, concurrent programming
  - Additional topics: Principles of computer networks, packet routing, TCP/UDP, network security, cloud computing and virtualization

- Teaching
  - Mondays: Lecture
  - Wednesday: Lecture, review, problem solving, questions, homeworks, quizzes, midterm exam

- MyCourses
  - Course material, important dates, schedule, news, events, homework, sample questions, grades, discussions, etc.

General Suggestions and Advice

- Think parallel
- Think concurrent
- Attend classes
- Visit MyCourses
- Read and digest course materials
- Do homeworks
- Solve problems
- Do not postpone
- Do not carry doubts
- Ask questions to the instructor and fellow students
- Make notes
- Write the algorithm first, and then code
- Discuss with fellow students and the instructor
- Write your own solutions and answers

Parallelism

- What is parallelism?
  - The objective is to execute tasks faster
  - Processes are executed simultaneously on parallel computing elements
- Data parallelism
  - Data to be processed exhibits parallelism (matrix operations, image processing)
  - The same task is performed on the same or different set/stream of data
- Instruction parallelism

  - Processors with multiple execution units
  - Execute multiple instructions through pipelining

- Task parallelism
  - Parallel tasks performed, each on a different computing element

- Device parallelism
  - Relates to hardware
  - Multiple cores, GPUs, parallel computers, clusters, grids, etc.

Applications

At any given moment:

- How many Visa credit cards are processed?
- How many aircrafts are in the air?
- How many students are trying to enroll into a class?
- How many phone calls are active?
- How many homes/people are downloading the same movie from Netflix?
- How many smart phones are using the same app?
- How many sensors are collecting sensory information?
- How many apps are active on your smart phone?

Example: Matrix Multiplication

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix} \times \begin{bmatrix} b_{11} & \dots & \dots & \dots \\ b_{21} & \dots & \dots & \dots \\ b_{31} & \dots & \dots & \dots \\ b_{41} & \dots & \dots & \dots \end{bmatrix} = \begin{bmatrix} \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}$$

This operation can be heavily parallelized.

MIPS and FLOPS

- MIPS - million instructions per second, CPU power to execute instructions
- MFLOPS - million floating point operations per second, used for computations, involves the whole system and not just the CPU. Involves memory, DISK, cache, bandwidth, I/O for high precision scientific calculations.

Why parallel computing?

Single processor systems are inadequate to complete certain compute intensive tasks, on time.

CPUs in computers

- Intel 386 DX (1985): 11 MIPS @ 33 MHz
- Intel Pentium Pro (1996): 541 MIPS @ 200 MHz
- Intel Core 2 X6800 (2006): 27079 MIPS @ 2.93 GHz
- Intel i7 5960X 8 Core (2015): 238,310 MIPS @ 3 GHz

Applications demand

- Several Tera/Peta FLOPS
- Sunway TaihLight, National Super Computer Center, China: 10,649,600 cores, 93 Peta FLOPS
- Titan Cray XK7, ORNL: 560,640 cores, 17.5 Peta FLOPS

Concurrency

What is concurrency? Concurrency is the potential for parallelism. It provides resource access to multiple processes and threads. It involves coordination, sharing, and synchronization. **Concurrency control** deals with correct and efficient access by multiple threads to shared resources because concurrently executing programs need to coordinate.

Distributed Computing

- Concurrent components are independent and communicate and coordinate through message passing or shared memory.
- The lack of a global clock causes problems because each component has its own clock and synchronization is a huge issue.
- Components fail independently. They are isolated from one another and redudant, and should not be visible to the end user.

Characteristics

- Heterogeneous
- Transparent
- Secure
- Privacy preserving
- Scalable
- Fault-tolerant
- Concurrent

Examples

- The Internet
- Sensor Systems
- P2P Systems
- Airlines
- Aircraft
- Cars

Reminders

Professor Mohan Kumar (Professor and Chair, CS Department):
`mjkvcs@rit.edu`
`https://cs.rit.edu/~mjk`
Include “CSCI 251” in your message header.

Jennifer Burt (Additional Contact):
`jennifer@cs.rit.edu`

There is no single textbook for this course. Check MyCourses for appropriate texts and references.

Homework

Identify parallelism in different daily activities and read the “Dining Philosophers” problem.

You can find all my notes at `http://omgimanerd.tech/notes`. If you have any questions, comments, or concerns, please contact me at alvin@omgimanerd.tech