CS 425/ECE 428
Distributed Systems

Nitin Vaidya
T.A.s

- Persia Aziz
- Frederick Douglas
- Su Du
- Yixiao Lin
• Course handout

  ... textbook
  ... office hours
  ... Piazza
  ... grading policy
  ... late submission policy
Course website

... mid-term exam schedule
... lectures page
... homework

... programming assignments
  (for 4 credit hours only)
What’s this course about?
What this course **is not** about ...
As you can see, I have memorized this utterly useless piece of information long enough to pass a test question. I now intend to forget it forever. You’ve taught me nothing except how to cynically manipulate the system.

- ??????
As you can see, I have memorized this utterly useless piece of information long enough to pass a test question. I now intend to forget it forever. You’ve taught me nothing except how to cynically manipulate the system.

- Calvin
Handout provided for 1st mid-term in Spring 2014 ... something similar this semester too
What is distributed computing?
What is distributed computing?

*Parallel* computing versus *distributed* computing

Example:

To add $N$ numbers where $N$ very large use 4 processors, each adding up $N/4$, then add the 4 partial sums

Parallel or distributed?
What is distributed computing?

• *Parallel* computing versus *distributed* computing

• Role of uncertainty in distributed systems
  – Clock drift
  – Network delays
  – Network losses
  – Asynchrony
  – Failures
A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable.

-- Leslie Lamport
What is distributed computing?

• *Parallel* computing versus *distributed* computing

• Role of uncertainty in distributed systems
  – Clock drift
  – Network delays
  – Network losses
  – Asynchrony
  – Failures
Clocks

• Notion of *time* very useful in real life, and so it is in distributed systems

• Example ...

Submit programming assignment by e-mail by **11:59 pm Monday**

By which clock?
How to synchronize clocks?
How to synchronize clocks?

Role of delay uncertainty
Ordering of Events

• If we can’t have “perfectly” synchronized clocks, can we still determine what happened first?
What is distributed computing?

• *Parallel* computing versus *distributed* computing

• Role of uncertainty in distributed systems
  – Clock drift
  – Network delays
  – **Network losses**
  – Asynchrony
  – Failures
Mutual Exclusion

• We want only one person to speak

• Only the person holding the microphone may speak

• Must acquire microphone before speaking
Mutual Exclusion

• How to implement in a message-passing system?
Mutual Exclusion

• What if messages may be lost?
What is distributed computing?

• *Parallel* computing versus *distributed* computing

• Role of uncertainty in distributed systems
  – Clock drift
  – Network delays
  – Network losses
  – Asynchrony
  – Failures
Agreement

• Where to meet for dinner?
Agreement with Failure

• Non-faulty nodes must agree
Agreement with
Crash Failure & Asynchrony
What if nodes misbehave?

• Crash failures are benign

• Other extreme ... Byzantine failures
Agreement with Byzantine failures (synchronous system)
How to improve system availability?

- Potentially large network delays ... network partition

- Failures
Replication is a common approach

Consider a storage system

- If data stored only in one place, far away user will incur significant access delay

➡️ Store data in multiple replicas,

Clients prefer to access “closest” replica
Replicated Storage

• How to keep replicas “consistent”?

• What does “consistent” really mean?
What’s this course about?
• Learn to “reason” about distributed systems... not just facts, but principles

• Learn important canonical problems, and some solutions

• Programming experience
• In class: we will focus on principles

• Supplemental readings: read about practical aspects, recent industry deployments
Distributed Computing ... our scope

• Communication models:
  – message passing
  – shared memory

• Timing models:
  – synchronous
  – Asynchronous

• Fault models
  – Crash
  – Byzantine
Shared Memory

• Different processes (or threads of execution) can communicate by writing to/reading from (physically) shared memory
Shared Memory
Distributed Shared Memory

- The “shared memory” may be simulated by using local memory of different processors
Distributed Shared Memory
Key-Value Stores
Consistency Model

• Since shared memory may be accessed by different processes concurrently, we need to define how the updates are observed by the processes

• *Consistency model* captures these requirements
Consistency #1

Alice: My cat was hit by a car.
Alice: But luckily she is fine.

Bob: That’s great!

What should Calvin observe?
Consistency #1

Alice: My cat was hit by a car.
Alice: But luckily she is fine.

Bob: That’s great!

What should Calvin observe?
Alice: My cat was hit by a car.
Alice: But luckily she is fine.  
Bob: That’s terrible!

What should Calvin observe?
Consistency #2

Alice: My cat was hit by a car.
Alice: But luckily she is fine. Bob: That’s terrible!

What should Calvin observe?