CSE 223A
Principles of Distributed Computing

Keith Marzullo

- CSE 2216
- e-mail marzullo@cs.ucsd.edu
- AIM kilioopu

Class: TuTh 2:00-3:20 HSS 2305A

If you’d like to see me, come by; if I'm not free then we'll make an appointment. My calendar is on zimbra.
Text

Web page (by weekend) will have notes and related papers.

☐ Some books you may find useful to browse:
Goal of course

- Cover some fundamentals of distributed computing
  - Leaning towards the more practical results rather than theory for theory’s sake
  - Reasoning about and consistency
  - Fault tolerance
  - Commit
  - Agreement
  - Quorums
  - Wait-freedom
  - ...

- By end, will have a set of tools available when designing your own distributed solution.
Grading

- Three homework assignments.
- One research project, due at the end of the quarter.
  - Details soon...
A distributed system is...

- Independent processors that communicate via a "narrow" interface
  - broadcast bus
  - point-to-point network
  - wide-area network

- True concurrency
- Partial failures
Why build distributed systems?

- Because that's the way they are.
- To capitalize on the true concurrency for better performance.
- To capitalize on partial failures.
Partial failures and availability

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 at DEC SRC in 1980s
Partial failures and availability

Assume $p_d = P(\text{processor down}) \approx 0.0001$ (about four minutes a month)

$p_u = P(\text{processor up}) \approx 0.9999$

Assuming IID failures,

$P(n \text{ processors up}) = (p_u)^n \approx 1 - (n \times p_d)$

... so distribution decreases availability.

e.g. with $n = 5$, availability is 0.9995 (unavailable about 22 minutes a month).
Partial failures and availability

\[ P(\text{at least one of } n \text{ processors up}) = 1 - (p_d)^n \]

... so can *increase* availability through replication.

e.g. with \( n = 5 \), availability is \( 1 - 10^{20} \) (unavailable about 24 femtoseconds a month?!!?).
Some key problems in distributed computing

- Reasoning about the *state* of a distributed system.
- Methods and protocols for making systems *mask* or *detect and recover* from failures.
- Protocols for *agreeing* on state, message delivery, process failures and recovery, etc.

... in the context of different assumptions about the environment (communication, clocks, failures...)

A simple distributed coordination problem

Two armies are camped on separate hills surrounding a valley in which the enemy army is encamped.

The enemy can vanquish each army separately, but if both armies attack at the same time, they can vanquish the enemy.
Two General's Problem
due to Jim Gray

Attack at dawn!
Two General's Problem

due to Jim Gray
Two General's Problem

I know to attack at dawn...

due to Jim Gray
Two Generals' Problem

I don't know that he knows to attack at dawn...

due to Jim Gray
Two General's Problem

due to Jim Gray
Two General's Problem

I know he knows to attack at dawn...

due to Jim Gray
Two General's Problem

I know he knows that I know to attack at dawn...

due to Jim Gray
Two General's Problem

due to Jim Gray
No finite protocol: proof

Assume a protocol exists, and consider the shortest run that leads to coordinated attack.
No finite protocol: proof
No finite protocol: proof
Two Generals in practice

How do banks solve this problem?
Two generals recap

- To attack requires *common knowledge* which can’t be obtained by exchanging messages that can be lost.
- This can be formalized (*epistomological logics*).
  - A process *knows* a fact if that fact holds in all executions indistinguishable from the current one.
Muddy children

☐ Seven school children are playing outside after it rained.

■ These are smart children - they’ve been studying logic and all reason flawlessly.

☐ At one point their teacher says to them “at least one of you has a muddy forehead”.
Muddy children

The teacher asks them “raise your hand if you know you have mud on your forehead”
- There were no mirrors, the students didn’t feel their foreheads, and no one pointed at anyone; a student could only determine it by logic.

When no one raised their hand, the teacher asked again.

When no one raised their hand, the teacher asked again. Some students raised their hands.
- How many did so?
- How did they know?