CSE 486/586 Distributed Systems
Introduction

Steve Ko
Computer Sciences and Engineering
University at Buffalo

Welcome to CSE 486/586

- Why do you want to take this course?

- Some positive feedback of this course…
  - *"(CSE 486/586) didn't only helped with understanding the concepts involved, but have also always given me something cool and interesting to talk about in interviews."*
  - *"I am actually learning new things."*

- Some negative feedback of this course…
  - *"Projects are a bit too much on the difficult side."*
  - *"The midterm came almost out of nowhere."*

- Are you ready? ;-)

Building a Distributed System

- *"The number of people who know how to build really solid distributed systems…is about ten"*
  - Scott Shenker, Professor at UC Berkeley

- The point: it's hard to build a solid distributed system.
- So, why is it hard?...but first of all…

What is a Distributed System?

What is a Distributed System?

- *A distributed system is a collection of entities with a common goal, each of which is autonomous, programmable, asynchronous and failure-prone, and which communicate through an unreliable communication medium.*

- This will be a working definition for us.

Why Is It Hard to Build One?

- Scale: hundreds or thousands of machines
  - Google: 4K-machine MapReduce cluster
  - Yahoo!: 4K-machine Hadoop cluster
  - Akamai: 70K machines distributed over the world
  - Facebook: 60K machines providing the service
  - Hard enough to program one machine!
- Dynamism: machines do fail!
  - 50 machine failures out of 20K machine cluster per day (reported by Yahoo!)
  - 1 disk failure out of 16K disks every 6 hours (reported by Google)
- What else?
  - Concurrent execution, consistency, etc.

C

OK; But Who Cares?

- This is where all the actions are!
  - What is the two biggest driving forces in the computing industry for the last 5 years?
  - It's the cloud!
  - And smartphones!
  - They are distributed!
- Now --- it's all about distributed systems!
  - Well…with a bit of exaggeration… ;-)

OK, Cool; How Am I Going to Learn?

- Textbook
  - Main: Distributed Systems: Concepts and Design, 5th Edition (Coulouris, Dollimore, Kindberg, Blair)
  - Optional: Distributed Systems: Principles and Paradigms, 2nd Edition, (Tanenbaum, Van Steen)
- Prerequisites
  - Minimum: CSE 250 Data Structures and Algorithms
  - Ideal: Basic networking concepts (TCP/IP, routing), basic OS concepts (processes, threads, synchronization, file systems), systems programming (pthread, socket)
- Lectures
- (Non-graded) HW assignments
- Programming assignments
- Exams

What Am I Going to Build?

- A "starter" project: project 0
- A distributed key-value storage (based on Amazon Dynamo) on Android in 3 stages: project 1 ~ project 3
- For each project, submit a solution/design document as well as code
- Individual submission

Important Policies

- Late submission: 20% penalty per day
- Regrading
  - If requested, the entire work will be regraded
- No "I"
- No makeup exam
- No grade negotiation

Academic Integrity Policies

- Academic integrity: exams, HW, and code
  - Copying others' code: no
  - Copying from other sources (the Web, books, etc.): get permission
  - Exceptions: http://developer.android.com (copy freely, but mark clearly that you copied)
  - http://stackoverflow.com (generally OK to see how things get done; but **do not copy and paste.**)
  - If found, the incident will be reported to the university
- Will use an automatic similarity checker.
  - When similar submissions are found, both will get F for the entire semester.
- Please be careful when using an online code repository, e.g., GitHub, BitBucket, etc.

How Can I Reach the Teaching Staff?

- Steve: 304 Davis
  - Lectures (MWF 3:00pm-3:50pm)
  - Office hours (MWF 4pm-5pm)
- TAs
  - (Tentative) Recitations: M (10:00 – 10:50) & F (2:00 – 2:50) @ Baldy 106
  - Office hours: generally 3 hours for each and every weekday will be covered.
  - Please do not expect that the TAs will stay more than 3 hours.
- Use Piazza (http://piazza.com/class), instead of email, mailing list, blog, etc.

C

2

What Exactly Am I Going to Learn? Distributed Systems 10 Questions!

- Course goal: answering 10 questions on distributed systems
  - At the end of the semester, if you can answer only 10 questions about distributed systems, you'll probably get an A.
  - Easy enough!
- What are those questions?
  - Organized in 6 themes
  - 1~2 questions in each theme
  - A few (or several) lectures to answer each question

What Exactly Am I Going to Learn?

- Introducing…
- Hydie!
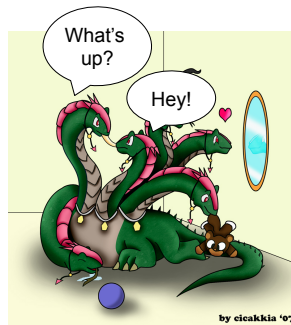

by cicakkia '07

Theme 1: Hint

What's up?

Hey!

by cicakkia '07

Theme 1: Communications

- Q1: how do you talk to another machine?
  - Networking basics
- Q2: how do you talk to multiple machines at once?
  - Multicast
- Q3: can you call a function/method/procedure running in another machine?
  - RPC

Theme 2: Hint

I thought I was doing it…

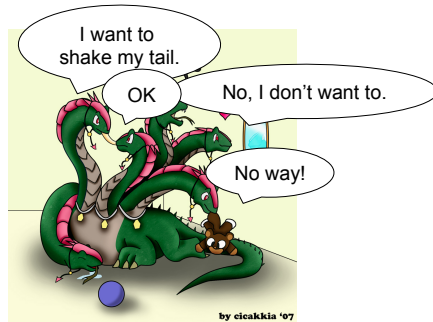I'm shaking my tail.

What? I'm doing it too!

by cicakkia '07

Theme 2: Concurrency

- Q4: how do you control access to shared resources?
  - Distributed mutual exclusion, distributed transactions, 2-phase commit, etc.

Theme 3: Hint

Theme 3: Consensus

- Q5: how do multiple machines reach an agreement?
  - Time & synchronization, global states, snapshots, mutual exclusion, leader election, paxos
- Bad news: it's impossible!
  - The impossibility of consensus

Theme 4: Hint
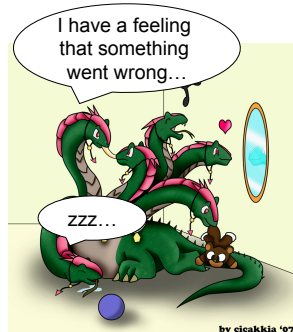
Theme 4: Storage Management

- Q6: how do you locate where things are and access them?
  - DHT, DFS

Theme 5: Hint

Theme 5: Non-Byzantine Failures

- Q7: how do you know if a machine has failed?
  - Failure detection
- Q8: how do you program your system to operate continually even under failures?
  - Replication, gossiping

Theme 6: Hint

We're under attack!

by cicakkia '07

Theme 6: Byzantine Failures

- Q9: how do you deal with attackers?
  - Security
- Q10: what if some machines malfunction?
  - Byzantine fault tolerance

Acknowledgements

- These slides heavily contain material developed and copyrighted by Indranil Gupta at UIUC.
- The material was originally developed for courses CS425/CSE424/ECE428 at UIUC.