CSE 486/586 Distributed Systems
Mid-Semester Overview

Steve Ko
Computer Sciences and Engineering
University at Buffalo

We're at a Mid-Point: What We've Discussed So Far

- Main communication infrastructure: the Internet
- Communication between two processes
  - Socket API
- Failure detection
- Concept of time in distributed systems
- Communication between multiple processes
  - Multicast algorithms
- Organization of distributed systems
  - Server-client
  - Peer-to-peer, DHTs
- Impossibility of consensus

The Other Half of the Semester

- Consensus algorithms: mutual exclusion, leader election, paxos
- Distributed storage basics: transactions and consistency
- Distributed storage case studies: Amazon Dynamo, NFS, Facebook Haystack, Facebook f4
- Remote procedure call
- Security
- BFT (Byzantine Fault Tolerance)

CSE 486/586 Administrivia

- Midterm: 3/15 (Wednesday) in class
  - Everything up to the last lecture
  - 1-page cheat sheet is allowed.
  - Blue or black ink pen
- Best way to prepare
  - Read the textbook & go over the slides.
  - Go over the previous exams.
- PA2-B due this Friday
  - Please remember that we'll be running code similarity checkers (automatic F if found too similar).

Data Centers

Data Centers

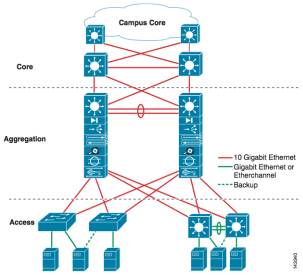
- Hundreds of Locations in the US

Inside

- Servers in racks
  - Usually ~40 blades per rack
  - ToR (Top-of-Rack) switch
- Incredible amounts of engineering efforts
  - Power, cooling, etc.

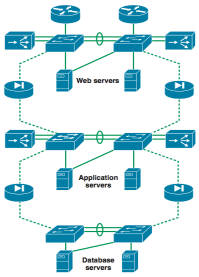
Inside

- Network



Inside

- 3-tier for Web services

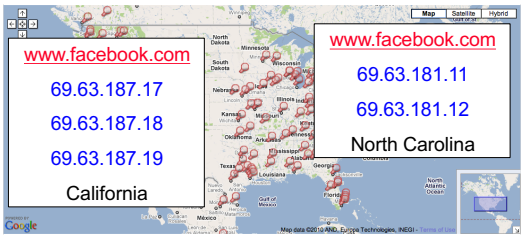


Web Services

- Amazon, Facebook, Google, Twitter, etc.
- World-wide distribution of data centers
  - Load balance, fault tolerance, performance, etc.
- Replicated service & data
  - Each data center might be a complete stand-alone web service. (It depends though.)
- At the bare minimum, you're doing read/write.
- What needs to be done when you issue a read req?
  - Server selection
- What needs to be done when you issue a write req?
  - Server selection
  - Replicated data store management

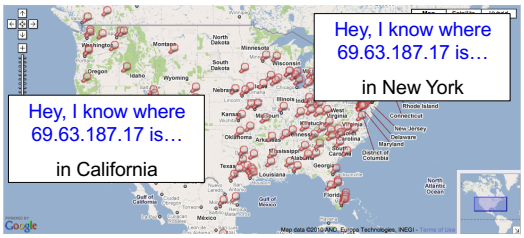
Server Selection Primer

- Can happen at multiple places
- Server resolution process: DNS -> External IP -> Internal IP
- DNS



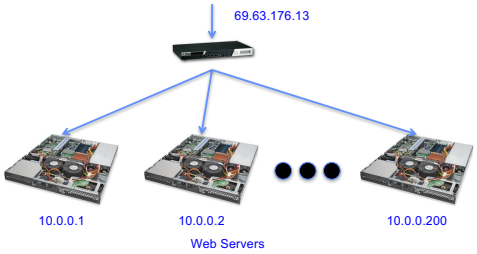
IP Anycast

- BGP (Border Gateway Protocol) level

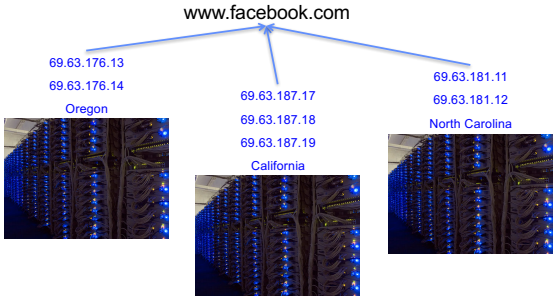


Inside

- Load balancers



Example: Facebook



Example: Facebook Geo-Replication

- (At least in 2008) Lazy primary-backup replication
- All writes go to California, then get propagated.
- Reads can go anywhere (probably to the closest one).
- Ensure (probably sequential) consistency through timestamps
  - Set a browser cookie when there's a write
  - If within the last 20 seconds, reads go to California.
- http://www.facebook.com/note.php?note_id=23844338919

Core Issue: Handling Replication

- Replication is (almost) inevitable.
  - Failures, performance, load balance, etc.
- We will spend most of our time looking at this in the second half of the semester.
- Data replication
  - Read/write can go to any server.
  - How to provide a consistent view? (i.e., what consistency guarantee?) linearizability, sequential consistency, causal consistency, etc.
  - What happens when things go wrong?
- State machine replication
  - How to agree on the instructions to execute?
  - How to handle failures and malicious servers?

Acknowledgements

- These slides contain material developed and copyrighted by Indranil Gupta (UIUC).