CS 3410: Distributed Systems

Spring 2018	Paper (due Thursday)	Topic	Project	Presentations
Jan 8–12	1. Google File System	the internet	Go tour	
Jan 15–19 (*MLK Day*)	2. Bigtable	Go, RPC		
Jan 22–26	3. Case study: Google	peer-to-peer	Go exercises	
Jan 29–Feb 2	4. Chord	concurrency	Chat service	
Feb 5–9	5. Dynamo		Chord: linked list ring	
Feb 12–16	6. Borg		Chord: finger tables	
Feb 19–23 (*Presidents' Day*)	7. Case study: Facebook	containers	Chord: fault tolerance	
Feb 26–Mar 2	8. Paxos			
Mar 5–9	9. Chubby	consensus	Paxos: shell, RPC server, etc.	Kademlia, Session Guarantees
Mar 12–16 (*Spring break*)	—	—	—	
Mar 19–23	10. Megastore		Paxos: all but propose	Byzantine Generals, CRUSH
Mar 26–30	11. MapReduce		Paxos	—, Snapshots
Apr 2–6	12. Case study: Twitter	databases		Petal, CAP
Apr 9–13	13. Spanner			Clocks, FLP
Apr 16–20	—			
Apr 23–27 (*Wednesday last day*)	—		MapReduce	

Resources

- Syllabus
- Examples from class
- A Tour of Go (start here to learn Go)
- Effective Go
- Recommended book: The Go Programming Language
- Go package docs
- Vocabulary:
  - ACID transactions
  - Serializable transactions
  - CAP theorem
  - At least once, at most once, exactly once
  - Idempotent
  - 2-phase commit
  - Pessimistic transactions, optimistic transactions
- TCP videos
  - TCP service model (16:27)
  - The end-to-end principle (10:33)
  - Sliding window (19:25)
  - Retransmission strategies (9:45)
- RPC demo app in Go
  1. introduction (8:22)
  2. server RPC (3:01)
  3. client RPC (4:50)
  4. command-line flags (13:58)
  5. call function (6:58)
  6. client shell (14:35)

7. actor (15:23)

Projects

- Project 0: A tour of Go
- Project 1: Go exercises (in CodeGrinder)
- Project 2: RPC chat room [plain] [pdf]
- Project 3: Chord key/value service [plain] [pdf]
- Project 4: Paxos key/value service [plain] [pdf]
- Project 5: MapReduce [plain] [pdf]
- First presentation
- Second presentation

Code to discover your own IP address. This does not work in all cases, but it is a useful starting point:

- getLocalAddress

Papers

1. The Google File System
2. Bigtable: A Distributed Storage System for Structured Data
    - What does data look like from the client's perspective, i.e., what are rows, column families, and columns?
    - How are responsibilities shared between master, tablet servers, and clients? What happens when each of these components fails and/or restarts?
    - How does Bigtable support small, random reads and writes when the data is stored in GFS, which optimizes for large, sequence reads and writes? Consider the interaction of SSTables (with compactions), memtables, and redo logs.
    - How is data stored on disk and in memory for efficient access? Consider the sequence of events for each of these cases:
        - Random data reads
        - Sequential data reads
        - Random data writes
        - Sequential data writes
    - What is: a group commit, a Bloom filter, a commit log
3. Case study: Google
    - Web search for a planet: The Google Cluster Architecture
    - Building Software Systems at Google and Lessons Learned (video 1:22:44)
4. Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications
5. Dynamo: Amazon's Highly-available Key-value Store
6. Large-scale cluster management at Google with Borg
7. Case study: Facebook
    - Scaling Facebook to 500 Million Users and Beyond (high-level overview, lessons learned)
    - Scale at Facebook (video, 1 hour)
    - Needle in a haystack: efficient storage of billions of photos (details about one specific service)
8. Paxos
    - The Part-Time Parliament
    - Paxos Made Simple
    - Paxos in 25 lines
9. The Chubby lock service for loosely-coupled distributed systems
10. Megastore: Providing Scalable, Highly Available Storage for Interactive Services
11. MapReduce: Simplified Data Processing on Large Clusters
12. Case study: Twitter
    - Real-Time Delivery Architecture at Twitter (56 minute video)
    - Decomposing Twitter: Adventures in Service-Oriented Architecture (50 minute video)
13. Spanner: Google's Globally-Distributed Database

Presentations

First presentation:

- A Note on Distributed Computing
- Impossibility of Distributed Consensus with One Faulty Process (Brad, Adam, Jared)
- Time, Clocks, and the Ordering of Events in a Distributed System (Adam, Brock, William)
- The Byzantine Generals Problem (Bredyn, Keith, Dakota)
- Session Guarantees for Weakly Consistent Replicated Data (Jeff, Blake, Anthony)
- CAP Twelve Years Later: How the “Rules” Have Changed (Lisa, Emily, Dylan)
- Distributed Snapshots: Determining Global States of Distributed Systems (Donald, Tyson)
- Life beyond Distributed Transactions: an Apostate’s Opinion
- Scale and Performance in a Distributed File System (AFS)
- Petal: Distributed Virtual Disks (Carl, Cruz, Joshua)
- CRUSH: Controlled, Scalable, Decentralized Placement of Replicated Data (Allison, Robbie, Andrew)
- Kademlia: A Peer-to-peer Information System Based on the XOR Metric (Kaden, Christian, Brandon)

Final presentation:

- On Designing and Deploying Internet-Scale Services (Brad, Adam K)
- Dapper, a Large-Scale Distributed Systems Tracing Infrastructure
- PNUTS: Yahoo!’s hosted data serving platform
- Mesa: Geo-Replicated, Near Real-Time, Scalable Data Warehousing (Dakota, Cruz, William)
- High-Availability at Massive Scale: Building Google’s Data Infrastructure for Ads (Emily, Lisa, Dylan)
- Twitter Heron: Stream Processing at Scale (Carl, Josh, Blake)
- Large-scale Incremental Processing Using Distributed Transactions and Notifications (Adam H)
- F1: A Distributed SQL Database That Scales
- In Search of an Understandable Consensus Algorithm (Allison, Robbie, Andrew)
- Paxos Made Live—An Engineering Perspective
- Flexible Paxos: Quorum intersection revisited (Jared, Keith, Bredyn)
- There is More Consensus in Egalitarian Parliments (Donald, Tyson)
- Scalability! But at what COST? (Kaden, Brandon, Christian)