CS 3410: Distributed Systems

Spring 2017	Paper (due Wednesday)	Topic	Project
Jan 9–13	1. Case study: Google	the internet	Go tour
Jan 17–20 (*MLK Day*)	2. Google File System	Go, RPC	
Jan 23–27	3. Bigtable	peer-to-peer	Go exercises
Jan 30–Feb 3	4. Chord	concurrency	Chat service
Feb 6–10	5. Dynamo		Chord: linked list ring
Feb 13–17	6. Case study: Facebook		Chord: finger tables
Feb 21–24 (*President's Day*)	7. Borg	containers	Chord: fault tolerance
Feb 27–Mar 3	8. MapReduce		1st presentation
Mar 6–10	9. Paxos	consensus	
Mar 13–17 (*Spring Break*)	—	—	—
Mar 20–24	10. Chubby		
Mar 27–31	11. Case study: Twitter		
Apr 3–7	12. Dremel/BigQuery	databases	Paxos
Apr 10–14	13. Megastore		
Apr 17–21	14. Spanner		
Apr 24–26	—		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
- RPC Chapter from *Network Programming with Go*
- Mandelbrot zoom mapreduce package
- Vocabulary:
  - ACID transactions
  - Serializable transactions
  - CAP theorem
  - At least once, at most once, exactly once
  - Idempotent
  - 2-phase commit
  - Pessimistic transactions, optimistic transactions

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. Case study: Google
    - Web search for a planet: The Google Cluster Architecture
    - Building Software Systems at Google and Lessons Learned (video 1:22:44)
2. The Google File System
3. Bigtable: A Distributed Storage System for Structured Data
4. Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications
5. Dynamo: Amazon's Highly-available Key-value Store
6. 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)
7. Large-scale cluster management at Google with Borg
8. MapReduce: Simplified Data Processing on Large Clusters
9. Paxos
    - The Part-Time Parliament
    - Paxos Made Simple
    - Paxos in 25 lines
10. The Chubby lock service for loosely-coupled distributed systems
11. Case study: Twitter
    - Real-Time Delivery Architecture at Twitter (56 minute video)
    - Decomposing Twitter: Adventures in Service-Oriented Architecture (50 minute video)
12. Dremel and BigQuery
    - Dremel: Interactive Analysis of Web-Scale Datasets
    - An Inside Look at Google BigQuery
    - BigQuery under the hood
13. Megastore: Providing Scalable, Highly Available Storage for Interactive Services
14. Spanner: Google's Globally-Distributed Database

Paper summaries

For each paper that we read, you must write a brief summary and submit it using Canvas. You should write about 1000 words. Here are a few guidelines:

- Be terse and concise. Avoid adjectives and other flowery language. You have very little space to summarize a lot of information.
- Summarize the technical content of the reading. Give equal weight to each section of what you read. If you read 10 pages, you should devote roughly 100 words to each page.
- If you do not understand something, try searching for additional information on Wikipedia or other sources on the web (there is a useful tool called Google…) or come to my office hours. Note that this works best if you do the reading early.

Things to avoid:

- Do not write about your reaction to the reading. Do not waste space telling if you liked it or did not like it or if you found it confusing or you were surprised by it.
- Do not write much about the background material and context. Focus on the specific question posed for each reading.
- Do not ask questions in your summary. The time for questions is before you write it.
- Do not use the first person. You are writing about the technical content of the reading assignment, not about yourself.
- Avoid the passive voice.

If you are unsure about any of these guidelines, talk to me, or take your answer to the writing center to get help from a writing tutor.

Presentations

- The Byzantine Generals Problem (midterm)
- Scale and Performance in a Distributed File System (AFS) (Lane, Franco, and Justin)
- Large-scale Incremental Processing Using Distributed Transactions and Notifications (midterm)
- SEDA: An Architecture for Well-Conditioned, Scalable Internet Services

- Time, Clocks, and the Ordering of Events in a Distributed System (midterm)
- Paxos Made Live—An Engineering Perspective
- Life beyond Distributed Transactions: an Apostate's Opinion
- There is More Consensus in Egalitarian Parliments
- PNUTS: Yahoo!'s hosted data serving platform (midterm)
- A Note on Distributed Computing (Charles, Lee, and Richard)
- In Search of an Understandable Consensus Algorithm (Brad and Jeff)
- Mesa: Geo-Replicated, Near Real-Time, Scalable Data Warehousing (Mohaned and Cameron)
- High-Availability at Massive Scale: Building Google's Data Infrastructure for Ads (midterm)
- Distributed Snapshots: Determining Global States of Distributed Systems
- Principles of Robust Timing over the Internet (Shawn, Wyatt, and Timothy)
- Twitter Heron: Stream Processing at Scale (midterm)
- CAP Twelve Years Later: How the "Rules" Have Changed (midterm)
- F1: A Distributed SQL Database That Scales
- Flexible Paxos: Quorum intersection revisited (Lena and Jean)