CS 3410: Distributed Systems

Spring 2025	Topics	Papers (due Wednesday)
Jan 6-10	Go, RPC	
Jan 13-17	<u>Go examples</u>	GFS, Bigtable
Jan 20-24 (<i>MLK Day</i>)	more Go, CAP, replicated state machines	
Jan 27-31	TCP, sockets, clusters	Paxos, Chubby
Feb 3-7	coherent caching, CAP	
Feb 10-14	transactions, 2-phase commit	Spanner, Calvin
Feb 17-21 (<i>President's Day</i>)	time, clocks, snapshots	
Feb 24-28	peer to peer	Chord, Dynamo
Mar 3-7	concurrency, actors	
Mar 10-14 (Spring Break)	_	_
Mar 17-21	databases	
Mar 24-28	big data	MapReduce, RDDs (Spark)
Mar 31-Apr 4	SOA, microservices	
Apr 7-11	eventual consistency	S3 Node
Apr 14-18		
Apr 21-25 (Thursday last day)		_

Changes to the schedule will be announced in class.

Resources

- Syllabus
- Examples from class
- Effective Go
- Recommended book: The Go Programming Language
- Go package docs
- Screencast on setting up Go and vim-go
- TCP videos
 - o TCP service model (16:27)
 - The end-to-end principle (10:33)
 - o Sliding window (19:25)
 - Retransmission strategies (9:45)
- RPC demo app in Go using Go RPC
 - 1. introduction (8:22)
 - 2. <u>server RPC (3:01)</u>
 - 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)
- Paxos assignment slides
- RPC chat assignment

Papers

- The Google File System
- Bigtable: A Distributed Storage System for Structured Data
- Paxos
 - Skim the original Paxos paper: The Part-Time Parliament
 - Read the simplified version in detail: Paxos Made Simple
 - $\circ~$ We will use this bare-bones protocol description for our assignment: $\underline{\texttt{Paxos}~in~25~lines}$
 - See how Paxos is implemented in modern systems: <u>Paxos vs Raft: Have we reached consensus on distributed consensus?</u>
- The Chubby lock service for loosely-coupled distributed systems
- Spanner: Google's Globally-Distributed Database

- Calvin: Fast Distributed Transactions for Partitioned Database Systems
 - Recommended: skim this paper first: The Case for Determinism in Database Systems
- Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications
- Dynamo: Amazon's Highly-available Key-value Store
- MapReduce: Simplified Data Processing on Large Clusters
- Resilient Distributed Datasets: A Fault-Tolerant Abstration for In-Memory Cluster Computing
- Using Lightweight Formal Methods to Validate a Key-Value Storage Node in Amazon S3

Optional reading

- Managing Update Conflicts in Bayou, a Weakly Connected Replicated Storage System
- Practical Byzantine Fault Tolerance
- Impossibility of Distributed Consensus with One Faulty Process
- The Byzantine Generals Problem
- Session Guarantees for Weakly Consistent Replicated Data
- CAP Twelve Years Later: How the "Rules" Have Changed
- Distributed Snapshots: Determining Global States of Distributed Systems
- Life beyond Distributed Transactions: an Apostate's Opinion
- Scale and Performance in a Distributed File System (AFS)
- Petal: Distributed Virtual Disks (Ethan)
- On Designing and Deploying Internet-Scale Services
- Dapper, a Large-Scale Distributed Systems Tracing Infrastructure
- PNUTS: Yahoo!'s hosted data serving platform (Lily, Linda)
- Mesa: Geo-Replicated, Near Real-Time, Scalable Data Warehousing
- · High-Availability at Massive Scale: Building Google's Data Infrastructure for Ads
- Twitter Heron: Stream Processing at Scale
- Large-scale Incremental Processing Using Distributed Transactions and Notifications (Dason, Joe, Luke)
- F1: A Distributed SQL Database That Scales (Christian, Carter)
- Paxos Made Live—An Engineering Perspective
- Flexible Paxos: Quorum intersection revisited
- Large-scale cluster management at Google with Borg (Trenonn, Braden, Sasha)
- Time, Clocks, and the Ordering of Events in a Distributed System
- Exploiting virtual synchrony in distributed systems
- Conflict-free Replicated Data Types
- Foundational distributed systems papers
- <u>Hall of fame awards</u>. These are systems papers that have been recognized as especially important, though note that only some of them are distributed systems papers.