Final Exam

- Tuesday December 11th, 5:40pm – 7:40pm
  - http://bannertools.binghamton.edu/exams/index.php?view=2&dept=CS
- Location: FA 258
- Closed book, closed notes
- Calculator required
- No cellphone, smartphone, tablet, laptop, etc.
- Covers Lecture 1 to Lecture 15
- Preparing: In-class practice problems, self-testing homework assignments, quizzes.

Topics we discussed

- Introduction to distributed systems
- Review of computer networks
- Web content delivery
  - DNS, CDN, Facebook case study
- Remote procedure call
- Distributed architectures
  - C/S, P2P, Hybrid
- Time and ordering in distributed systems
- Multicast
- Global snapshot
- Mutual exclusion and leader election

Topics we discussed

- Distributed Transactions
  - Concurrency control
  - Atomic commit protocol
- Consistency models
  - Implementing consistency models
- CAP theorem
- Distributed key-value store
- Distributed file systems
- MapReduce
- Fault tolerance

Disclaimer

- This review helps you better prepare for the final exam.
- It is by no means comprehensive.
- Study all materials for the exam.

Introduction to distributed systems

- Heterogeneity
- Scalability
  - size, geographical, administrative
  - scalability techniques
- Transparency
  - access, location, relocation, migration, replication, concurrency, failure

Distributed architectures

- Centralized architectures
- Decentralized architectures
  - unstructured P2P: Gnutella, Kazaa
  - structured P2P: Chord
- Hybrid architectures
  - CDN
  - BitTorrent

Communication in distributed systems

- Basics of computer networks
  - How do computers talk to each other? – sockets
  - IP, TCP, UDP, DNS, etc.
- Remote procedure call
  - Call a procedure on a remote machine
  - Functions of client-side and server-side stubs
  - Marshalling/un-marshalling
  - Semantics of RPC
    - Failures of RPCs
    - Invocation semantics
  - Interface definition language – purpose

Communication in distributed systems

- Multicast
  - Talk to multiple machines at once
  - What is ordered multicast? What's its purpose?
  - Can you recall a real-world use of ordered multicast?
  - How to implement ordered multicast?

Lack of global clock – physical clock synchronization

- Cristian's algorithm
- Berkeley algorithm
- Network Time Protocol (NTP)

Lack of global clock – logical clocks

- Lamport's algorithm
  - Happened-before relationship
  - Concurrent events
  - Partial ordering
  - Algorithm
  - Deficiencies
- Vector timestamp
  - Goals
  - Algorithm
  - How to identify concurrent events

Lack of global clock – global state

- Chandy and Lamport's "snapshot" algorithm
- What are its assumptions about the communication channel?

Concurrency – access of shared resources

- Distributed mutual exclusion
  - Central server algorithm
  - Ricart-Agrawala algorithm
  - Ring-based algorithm
  - Meakawa's algorithm

- Distributed leader election
  - Bully algorithm
  - Ring-based election algorithm

Concurrency in distributed transactions

- Serial equivalence
- Concurrency control schemes:
  - Locking
  - Optimistic concurrency control
  - Timestamp-based concurrency control

Atomicity in distributed transactions

- Two phase commit
- Three phase commit

Distributed storage

- **Replication** for fault tolerance
- But may cause **consistency** issues

- Data centric consistency models
  - Strict, linearizability, sequential, causal, FIFO
- Client centric consistency models
  - Monotonic read, monotonic write, read your writes, write follows reads
- Eventual consistency

Distributed storage – implementation of sequential consistency

- Primary-backup
- Active replication
- Quorum-based

Distributed storage – CAP theorem

- **Strong consistency**: all nodes see same data at any time, or reads return latest written value by any client
- **High availability**: the system allows operations all the time, and operations return quickly
- **Partition-tolerance**: the system continues to work in spite of network partitions

Distributed key-value store

- Systems that navigate the CAP theorem

- Dynamo:
  - Virtual nodes, replication, data versioning, sloppy quorum, consistency, replica synchronization, etc.
- Cassandra:
  - Partitioning, replication strategies, memtable, sstable, column-oriented, bloom filter, consistency, etc.
- HBase:
  - Region, memstore, HFile, consistency, etc.

Distributed file system

- NFS:
  - Architecture, access model, file access semantics, etc.
- AFS:
  - Architecture, access model, etc.
- GFS:
  - Architecture, master, chunkserver, read, write, append, etc.
- HDFS

Hadoop and MapReduce

- Transform problems into MapReduce programs
- Hadoop internals

Fault Tolerance

- Fault tolerance in process groups
- Byzantine agreement with faulty process
- Proof-of-work in Bitcoin

Good luck!

- Thanks for joining us for the class.
- Wish you a fruitful final season and enjoy the holidays!