Final Review

CSE 486: Distributed Systems

Ethan Blanton

Department of Computer Science and Engineering

University at Buffalo

Logistics

Your final will be in Knox 104 at 11:45 AM on Friday, May 10.

It will be about two hours long.

It is comprehensive.

You may bring and use:

- Your knowledge
- A writing utensil (not red)
- Nothing else

If you are late, you will not be permitted to take the exam.

Gossip Protocols

- Gossip protocols provide probabilistic delivery
- Cost is usually about $c \cdot |G| \log |G|$ per message
- Lightweight Probabilistic Broadcast solves:
  - Changing group membership
  - Process membership knowledge overhead for very large $|G|$

Leader Election

- Centralized authority doesn’t mean permanent authority
- Distributed elections can be held
  - Bully algorithm
  - Ring algorithm
- Global identifiers keep cropping up
- Proof of work can make global IDs safer
- Security guarantees require threat models

Consensus

- Deciding on zero or one is powerful
- Synchronous systems can decide with an arbitrary, predefined number of failures
- Asynchronous systems cannot decide …maybe
  - Failure is indistinguishable from delay

Byzantine Agreement

- Byzantine failures present differently in different circumstances
- Storytelling gets you published
- Consensus can be reached even with Byzantine failure (in a synchronous system)
- More than 2/3 of processes must be honest to achieve this

Mutual Exclusion

We will see mutual exclusion again.

- Mutual exclusion is valuable for distributed systems
- Races occur when ordering is important and not maintained
- Mutexes model mutual exclusion
- Deadlocks can arise when mutexes are used
- Logical clocks can be used to implement distributed mutexes

The Raft Consensus Protocol

- Raft provides consensus through quorum.
- Almost half of the participants can fail without losing consensus.
- Decomposing elections, membership changes, and log manipulation makes Raft easier to understand.

Quorum

- Quorum can solve many problems
- Different quorums have different uses
- Maekawa’s mutual exclusion uses quorum for mutexes
- Mutexes can be solved with relatively few members in a quorum

Consistency and Transactions

- Transactions are multiple actions grouped together into an atomic entity.
- The actions in transactions can be interleaved.
- Some interleavings are inconsistent.
- Consistent interleavings are serializable.
- Two-phase locking preserves serializability.

Locking and Commit Protocols

- Non-exclusive locking can increase concurrency
  - Deadlock and aborts can be triggered!
- Read/Write locks allow multiple readers in parallel
- Two-version locks allow multiple readers and one writer
- Deadlock detection and abort-and-retry can be effective
- Distributed transactions require multi-process atomic commits
- Two-phase commit solves races in a simple commit

Distributed Systems Security

- Distributed security is very hard, and approaches depend on the application.
- The principle of least authority can be used to separate concerns and minimize collateral damage from vulnerabilities.
- Cryptography is important when infrastructure is untrusted.
- TLS is used to protect socket communications.
- Kerberos is a distributed authentication and key exchange protocol that requires minimal trust between entities.