Commit Protocols

February 26, 2004

Normal 2PC

1) prepare
2) Vote (commit/abort)
3) Decision (c/a)
4) Ack Decision

Timeline: 2PC

Coordinator
- Prepare
- Force C/A
- Commit/Abort
- Write end "forget"

Subordinate
- Prepare
- Commit/Abort
- Force C/A
- Ack

Coordination Subordinate
- Prepare
- Vote C or A
- Force prepare abort
- Force C/A
- Commit/Abort
- Ack

Subordinate Crashes

Coordinator
- Prepare
- Force C/A
- Commit/Point
- Commit/Abort
- Ack

Subordinate
- Prepare
- Vote C or A
- Force prepare abort
- Force C/A
- Commit/Abort
- Ack

Write end "forget"
Coordinator Crashes

- Coordinator
  - Prepare
  - Crash
  - Force C/A
  - Crash
  - Write end
  - "forget"

- Subordinate
  - Force prepare/abort
  - Vote C or A
  - Commit/Abort
  - Force C/A
  - Ack
  - Commit: 2 force, 2 messages
  - Abort: 2 force, 2 messages

Presumed Abort

- Coordinator
  - Prepare
  - Force C/A
  - Write abort & "forget"
  - Commit Point
  - Commit/Abort
  - Force C or write A

- Subordinate
  - Force prepare/abort
  - Vote C or A
  - Ack only commits
  - Commit Point
  - Commit/Abort
  - Force C or write A
  - Commit => Write end, "forget"
  - Commit: 2 force, 2 messages
  - Abort: 0-1 force, 1 message

Presumed Commit

- Coordinator
  - Force collecting
  - Prepare
  - Force prepare/abort
  - Vote C or A
  - Force C/A
  - Commit Point
  - Commit/Abort
  - Force A or write C
  - Ack only aborts
  - Abort => Write end, "forget"

- Subordinate
  - Force prepare/abort
  - Vote C or A
  - Ack only commits
  - Commit: 2 force, 2 messages
  - Commit: 1 force, 1 message
  - Abort: 2 force, 2 messages

Presumed Commit

- Coordinator
  - Force collecting
  - Prepare
  - Force prepare/abort
  - Vote C or A
  - Force C/A
  - Commit Point
  - Commit/Abort
  - Force A or write C
  - Ack only aborts
  - Abort => Write end, "forget"

- Subordinate
  - Force prepare/abort
  - Vote C or A
  - Ack only commits
  - Commit: 2 force, 2 messages
  - Commit: 1 force, 1 message
  - Abort: 2 force, 2 messages
Ninja 2PC Optimizations

- Focus on availability
- Sub dies in phase 2:
  - Complete without the ack
  - Recovery will get the new version from another replica
- Coordinator dies:
  - Replicas contact each other to determine the result
  - Faster than waiting for coordinator to recover

Ninja Apply function

- Use 2PC to apply delta to all replicas
  - Phase 1: prepare
  - Phase 2: commit
  - Ensure serializable updates
- Livelock
  - Contention may cause aborts
  - Replicas that are out of order may abort
  - No guarantee of progress
  - Real problem under high update rates

Weak Apply

- Phase 1: Commit (!)
  - Order may vary, but fine for some uses
  - Retry commit until it succeeds on all replicas
- Phase 2: ack => stop retrying
  - Much less livelock!