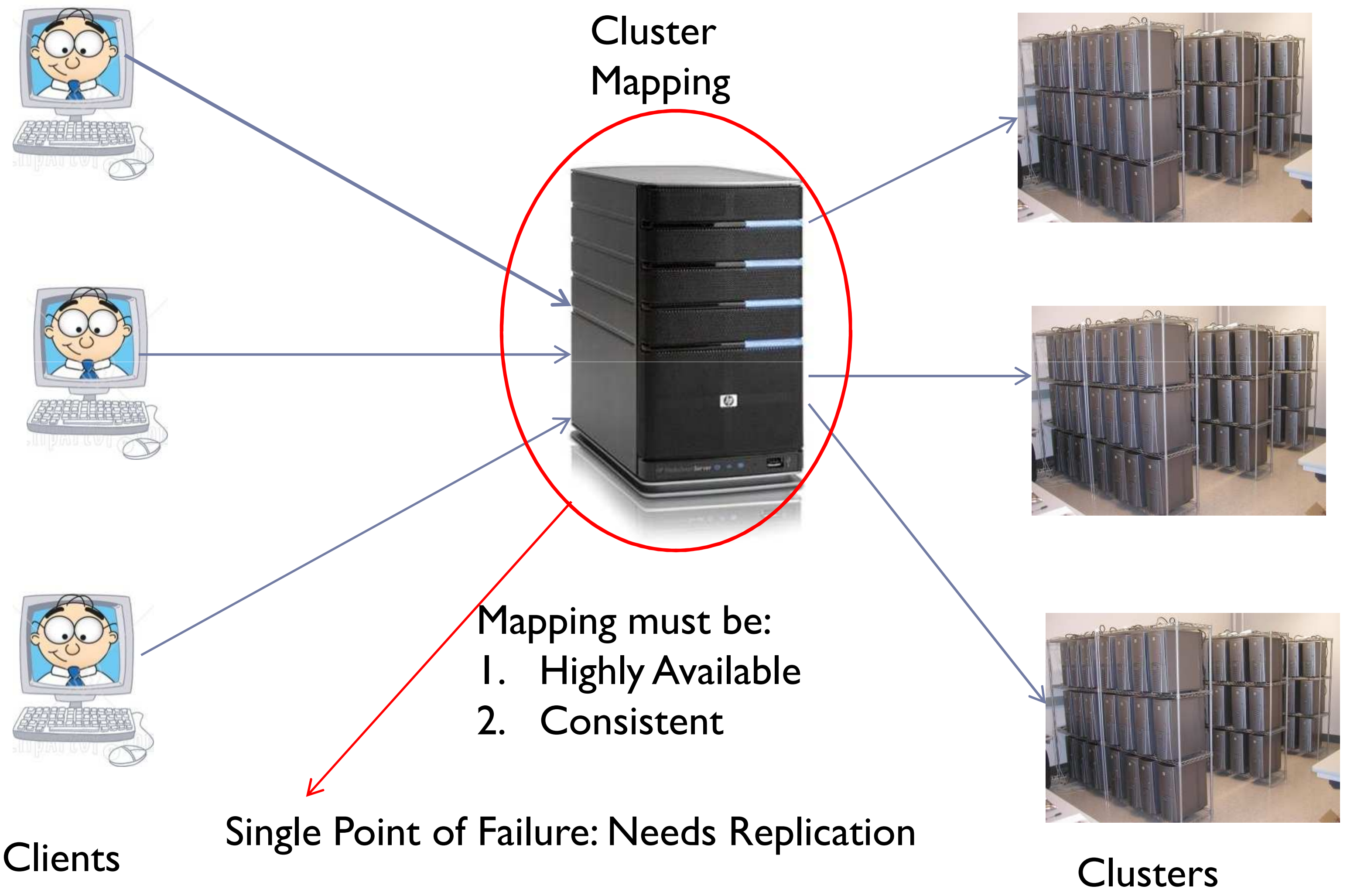
The Chubby Lock Service for loosely-coupled distributed systems

Mike Burrows, Google Inc.

OSDI 2006

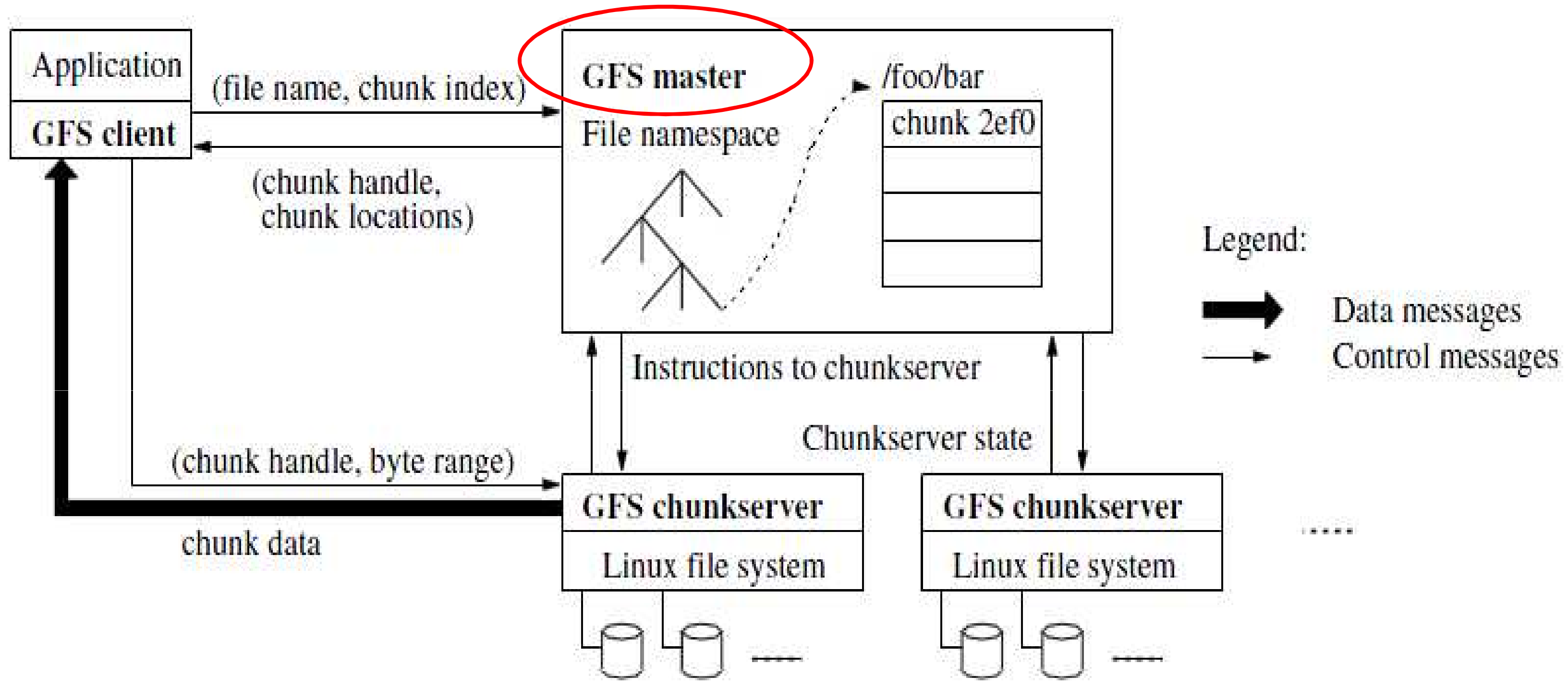
A Typical Scenario



Mapping must be:

1. Highly Available
2. Consistent

Another Scenario : GFS



Needs to appoint a master server

Ref: The Google File System, Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung, Google, 19th ACM Symposium on Operating Systems Principles,2003

What problem are we looking at?

- Distributed Consensus Protocol
- Asynchronous system
- But it is impossible to achieve consensus in an asynchronous system!
- Solution: Paxos Protocol

What is Chubby?

- Distributed locking mechanism
- Stores small files, like metadata information
- Primary goal: Reliability and Availability
- Purpose:
- Developers: Coarse Grained synchronization , Elect leaders
- Actual: As a Name Service

- Typically one chubby instance or "cell" per data center (10,000 4 processor machine on avg.)

A Chubby Cell

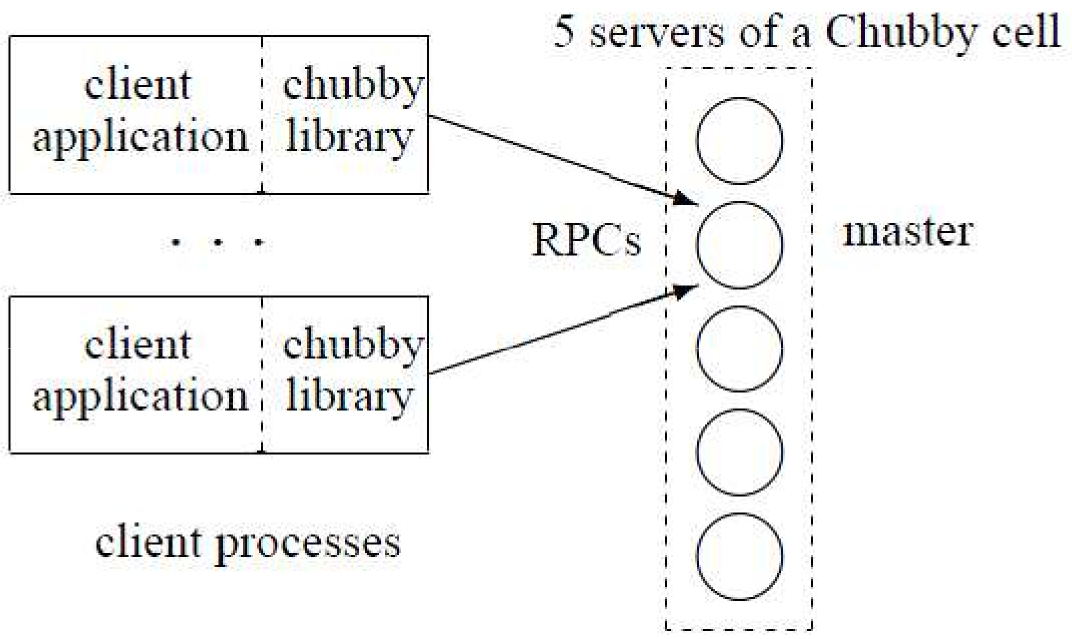


Figure 1: System structure

A Chubby Cell : Optimized for Reads

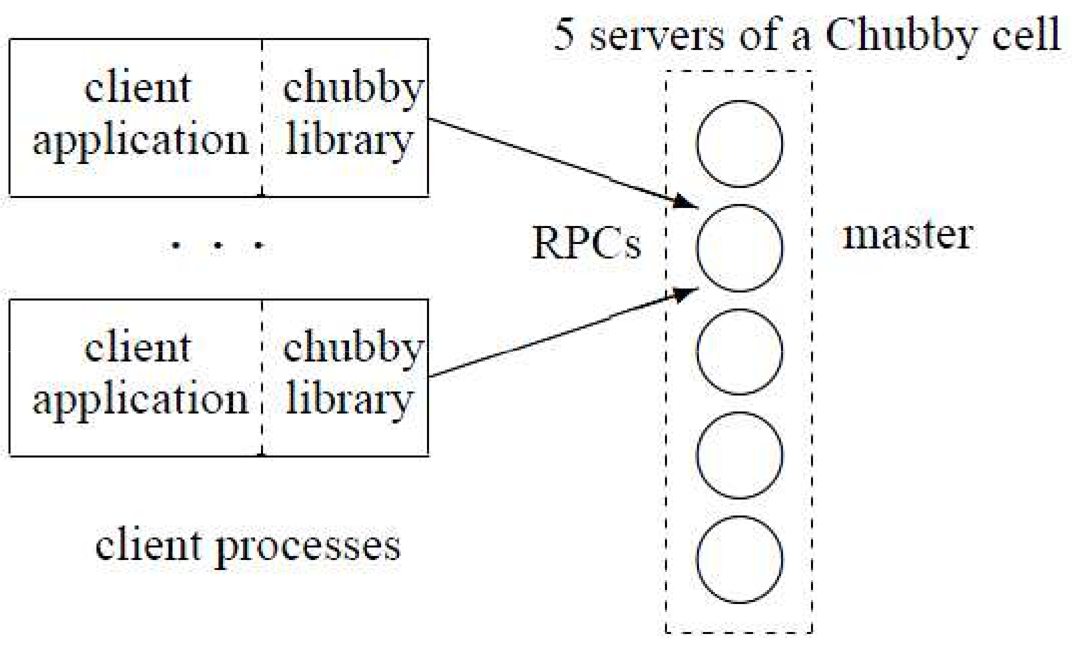


Figure 1: System structure

- Key: Master Lease
- Clients maintain consistent, write-through cache of file data and node metadata

A Chubby Cell : Writes

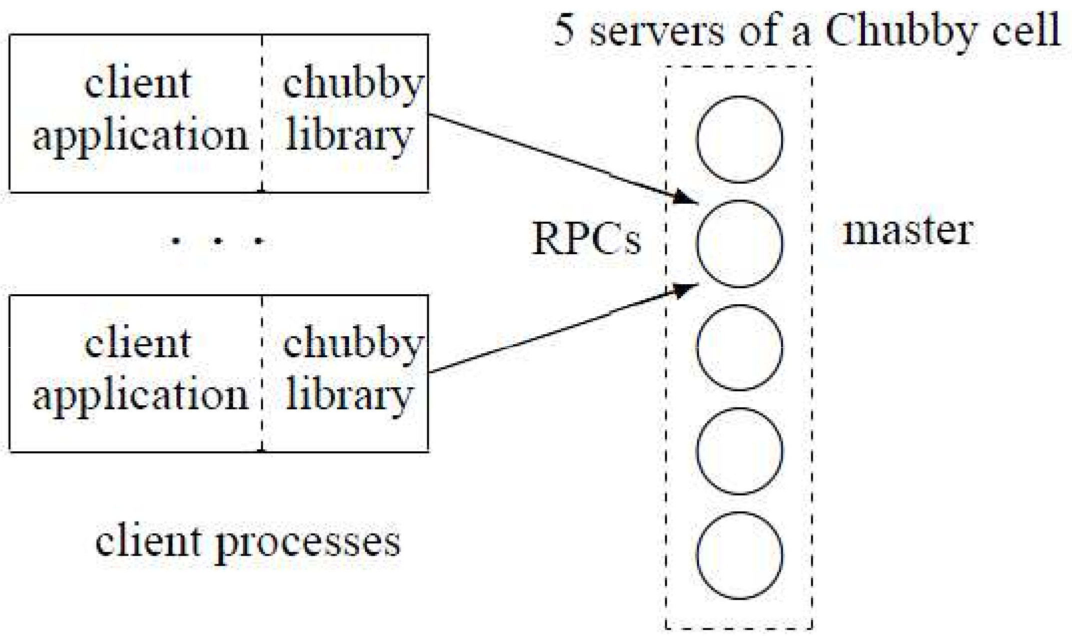


Figure 1: System structure

- What if master fails?
- What if a replica fails?

A Chubby Replica

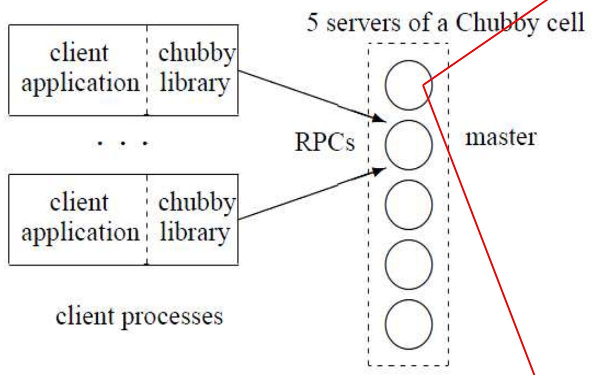


Figure 1: System structure

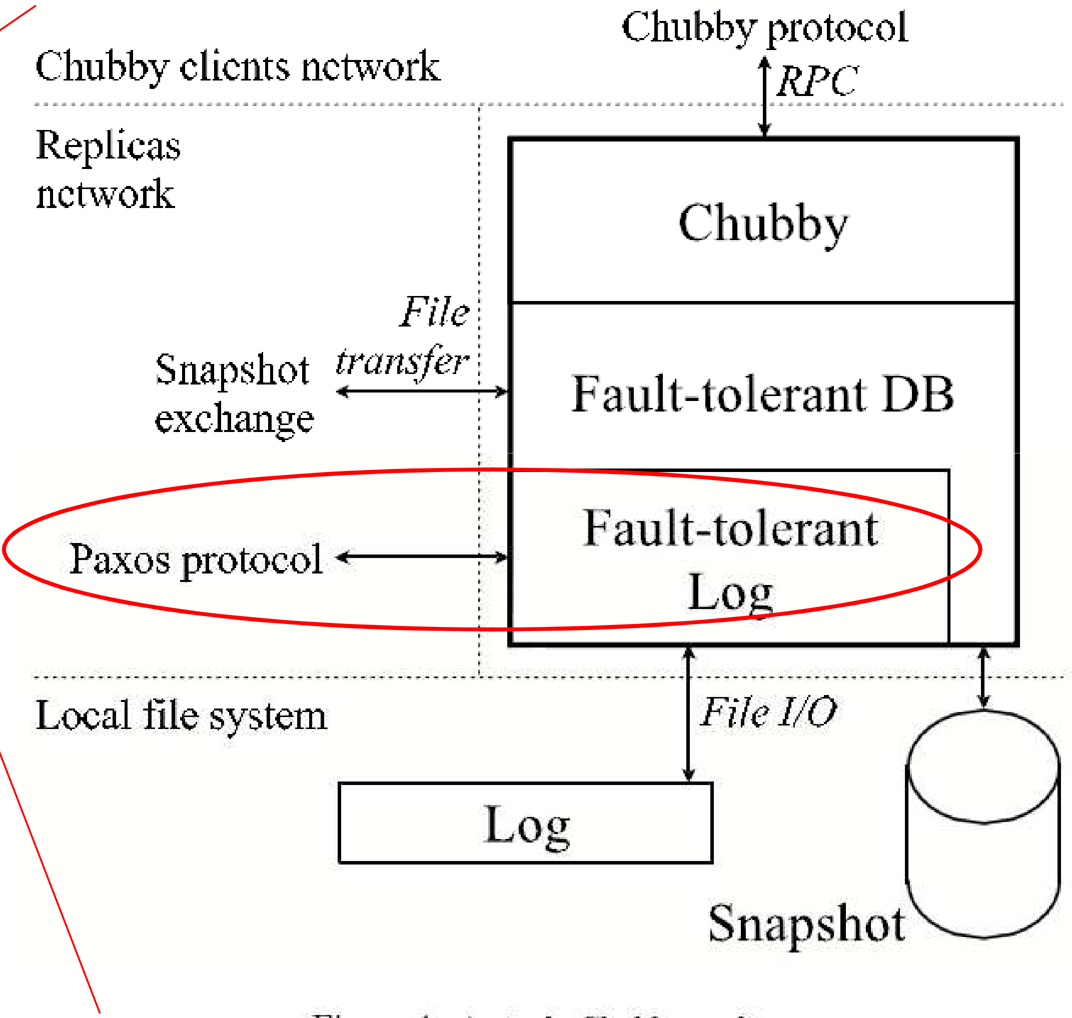
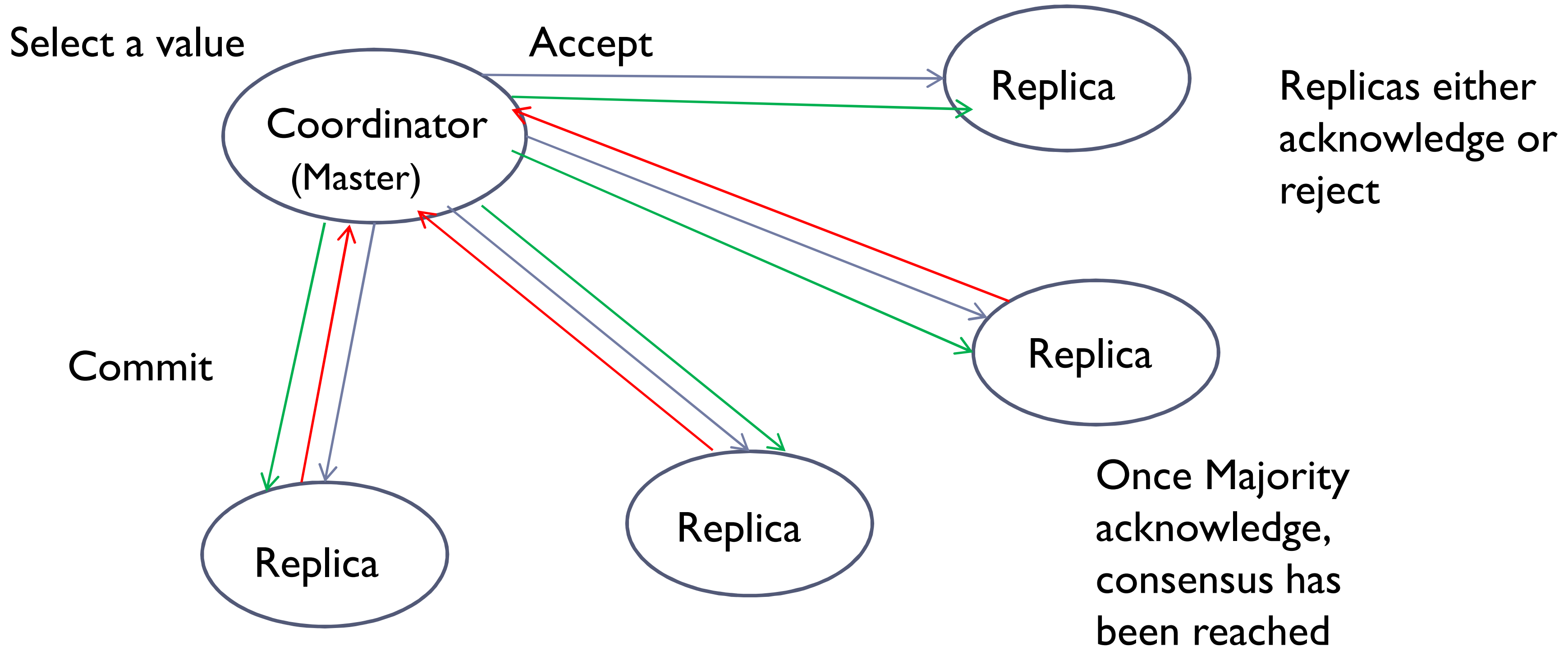


Figure 1: A single Chubby replica.

Ref: Paxos Made Live - An Engineering Perspective; Tushar Chandra, Robert Griesemer, Joshua Redstone ; PODC '07

Paxos Protocol


Select a value
Accept
Coordinator
(Master)
Replica
Replicas either acknowledge or reject
Commit
Replica
Replica
Replica
Once Majority acknowledge, consensus has been reached

Used to agree upon the next entry in replica's log
Originally used by Paxons to run their part-time parliament!

Why not have multiple replicas become coordinator?
Why not have a library implementing Paxos? Why use Chubby at all?

The Chubby Client Interface

- Exports a file system interface
- Files and directories are called nodes
- Typical node name:

  /ls/foo/wombat/pouch

- Ephemeral Nodes: temporary files, indicators that a client is alive
- Node metadata: Three ACL names which are themselves files
- Clients open nodes to obtain handles

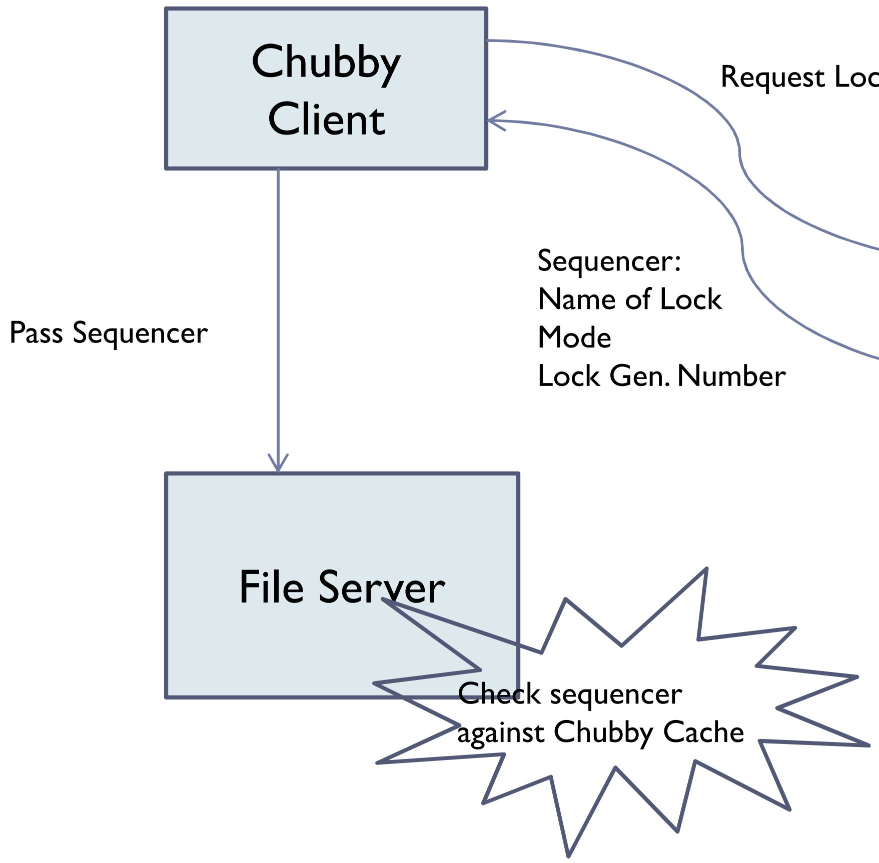
The Chubby Client Interface - API

- Open()
- Close()
- Poison()
- GetContentsAndStat(), GetStat(),ReadDir()
- SetContent()
- Delete
- Acquire(), TryAcquire(), Release()
- GetSequence(), SetSequencer(), CheckSequencer()

Locks

- Locks seen as another node
- An entry into the chubby database
- Either exclusive or shared mode
- Locks are advisory

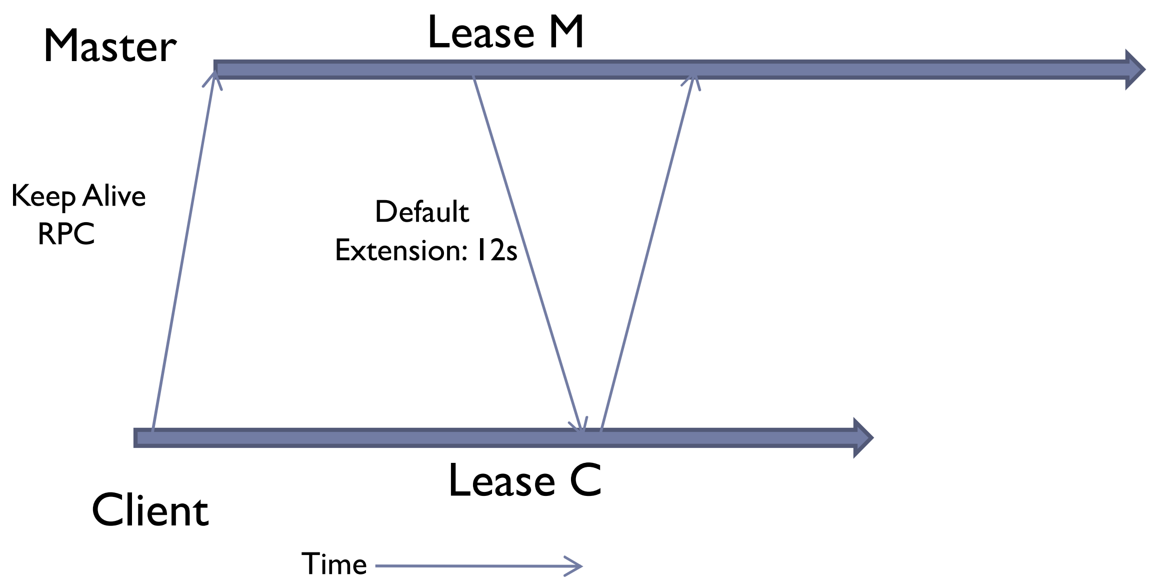
Sequencers and Lock Delay



What if Server does not support sequencer?

Use: Lock Delay

Sessions and Keep Alives



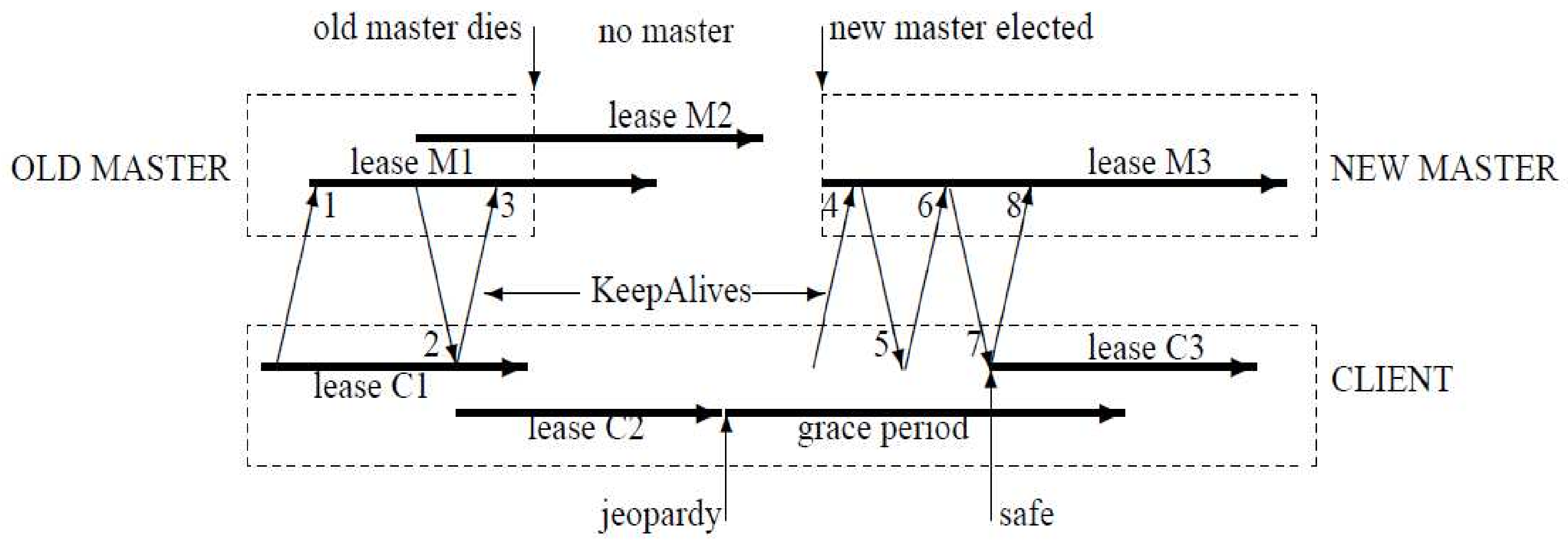
What about time synchronization?
Solution: Grace Period with session in jeopardy

Session Lease : Interval of time during which master guarantees not to end session

Keep Alive: Also used to inform clients of events and cache invalidations

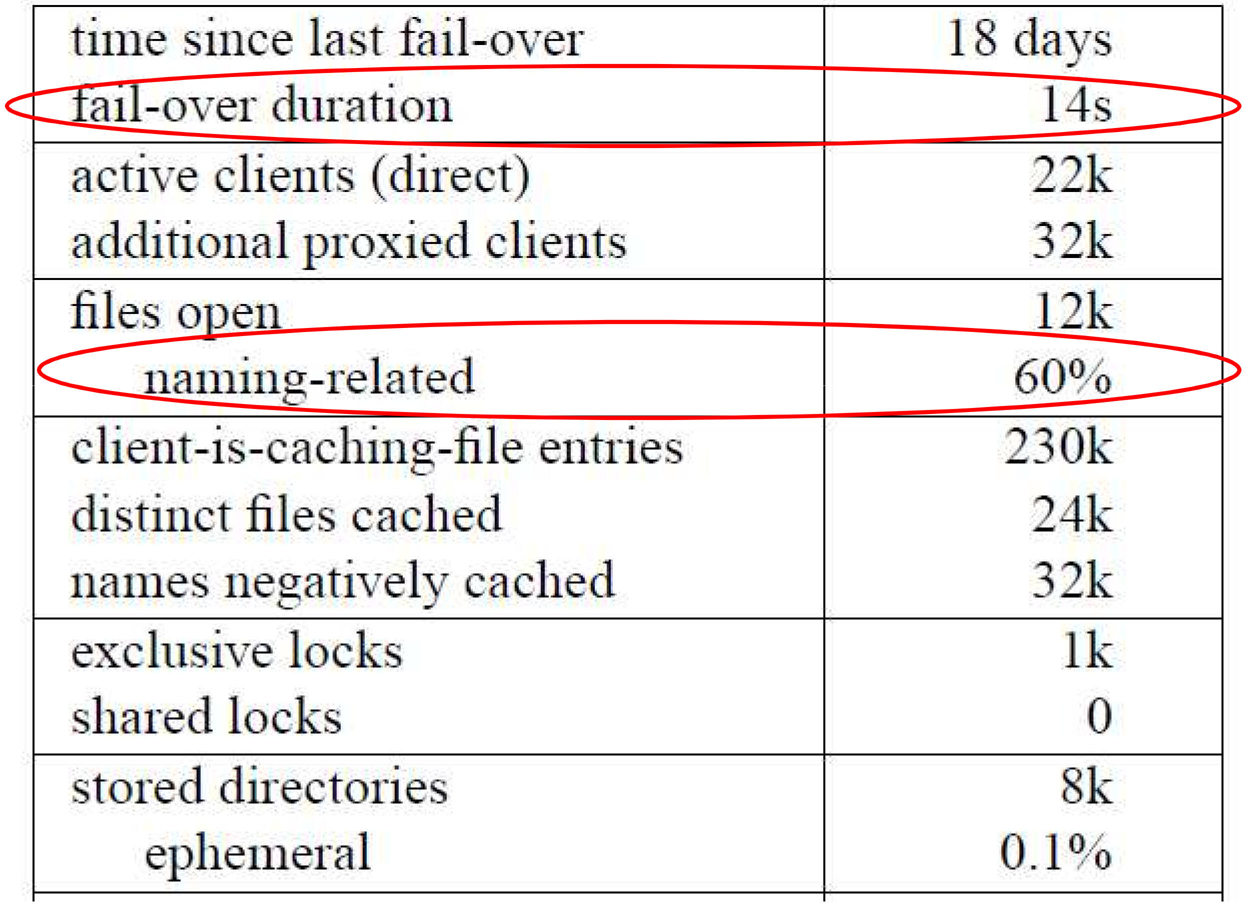
Advantage: All RPC's flow from client to master-helps in overcoming firewalls!
Disadvantage? TCP's back off policy- developers later migrated to UDP

Fail-Overs



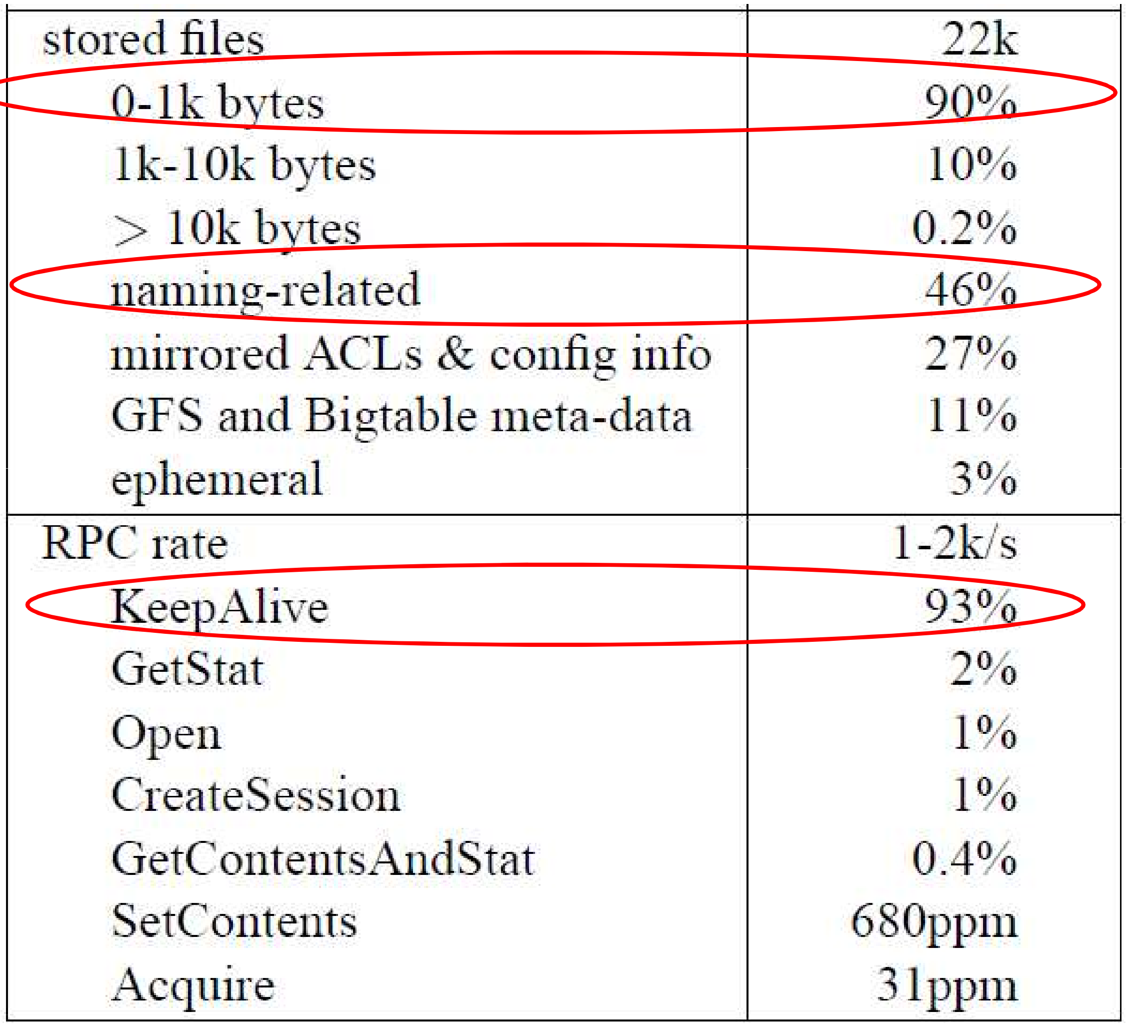
Default Grace Period: 45s (How do you decide what is a good value?)

Some Statistics



time since last fail-over	18 days
fail-over duration	14s
active clients (direct)	22k
additional proxied clients	32k
files open	12k
naming-related	60%
client-is-caching-file entries	230k
distinct files cached	24k
names negatively cached	32k
exclusive locks	1k
shared locks	0
stored directories	8k
ephemeral	0.1%

Some Statistics



stored files	22k
0-1k bytes	90%
1k-10k bytes	10%
> 10k bytes	0.2%
naming-related	46%
mirrored ACLs & config info	27%
GFS and Bigtable meta-data	11%
ephemeral	3%
RPC rate	1-2k/s
KeepAlive	93%
GetStat	2%
Open	1%
CreateSession	1%
GetContentsAndStat	0.4%
SetContents	680ppm
Acquire	31ppm

Use as Name Service

- DNS entries have a TTL for caching – discarded when not refreshed within TTL
- Polling to maintain caches results in heavy traffic.
- Chubby Keep Alives and invalidations provide a solution

Few More Uses

- Used in GFS to elect master server

Whoever obtains lock on a lock file becomes the master- writes its identity on the lock file.

- BigTable
- Elect master
- Allow master to discover servers it controls
- Permit clients to find masters

In addition, both use Chubby to store metadata.

“If Chubby becomes unavailable for an extended period of time, Bigtable becomes unavailable.”

Discussion Points

- No mathematical analysis of the system
- No comparison with other existing system
- Throughput considered secondary
- What if a client is malicious?
- Paxos Protocol assumes that replicas have access to persistent storage that survives crashes. What if that is not the case?
- Only coarse-grained lock provided
- File Size limited to 256 KB

Questions?

Thank You!