Distributed Systems

06r. Assignment 5 review

Paul Krzyzanowski

Rutgers University

Fall 2016

Assignment 5 Review

Question 1

Raft uses a single leader (one server is elected as a leader). Explain how Raft performs leader election.

Short answer: each candidate starts a random timer before proposing itself as a leader & sending election messages to the group.

If you receive a leader proposal and you have not yet proposed yourself, you will acknowledge that candidate and not vote for yourself.

If a candidate gets majority votes, it becomes the leader.

Question 1 – Longer Answer

Raft uses a single leader (one server is elected as a leader). Explain how Raft performs leader election.

To start an election, a candidate votes for itself and sends a "request vote" message to all other servers. Other servers that have not yet voted and receive the request acknowledge the candidate to be the leader. Each server that receives a request will vote for at most one candidate.

If the candidate receives a majority of acknowledgements, it becomes the leader.

If the candidate does not win or lose an election, it times out and starts a new election. Randomized timeouts are used to ensure that split votes happen rarely.

To support recovery and avoid stale state, a "term number" is incremented after each election

If the candidate receives a heartbeat from another server and that leader's term # is at least as large as the candidate's current term, then the candidate recognizes the leader as legitimate and becomes a follower.

Question 2

An elected leader takes client requests. Each request is essentially a log entry that will be replicatedamong the servers. When is a log entry committed in Raft?

A log entry is committed once the leader that created the entry has replicated it on a majority of the servers.

Committed means that the log entry is applied to the state machine.

Question 3

As Dropbox's design evolved,why did Dropbox split the original web server into two web servers? [What was the function of each server?]

Dropbox ran out of capacity at the server because all uploads and downloads went to one server.

One server dealt with metadata. Another dealt with file uploads and downloads.

Question 4

Why were notification servers added?

Notification servers were added to not require clients to continuously poll the server.

This reduces the load on the system since clients that don't have changes don't impose any traffic onto the servers.

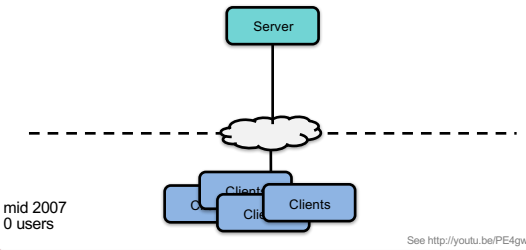
Question 5 (Dropbox)

Why was RPC-based communication added to the blockservers instead of having them talk to the database?

Avoids multiple round-trip calls from the blockserver to the database..
RPCs can contain higher-level directives.

Dropbox: architecture evolution: version 1

– One server: web server, app server, mySQL database, sync server

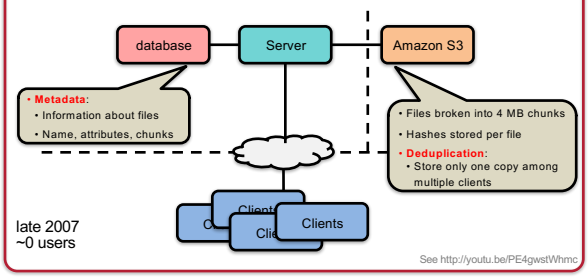


mid 2007
0 users

See http://youtu.be/PE4gwstWhmc

Dropbox: architecture evolution: version 2

– Server ran out of disk space:
moved data to Amazon S3 service (key-value store)
– Servers became overloaded: moved mySQL DB to another machine
– Clients periodically polled server for changes

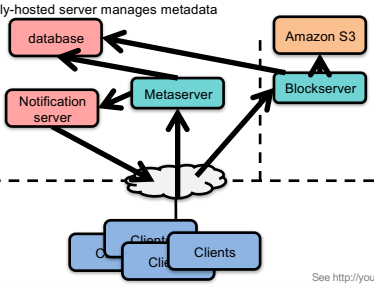


late 2007
~0 users

See http://youtu.be/PE4gwstWhmc

Dropbox: architecture evolution: version 3

– Move from polling to notifications: add **notification server**
– Split web server into two:
• Amazon-hosted server hosts file content and accepts uploads (stored as blocks)
• Locally-hosted server manages metadata

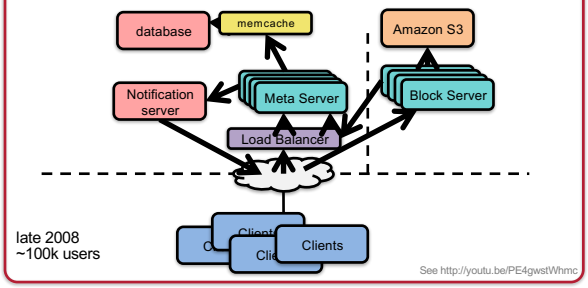


early 2008
50k users

See http://youtu.be/PE4gwstWhmc

Dropbox: architecture evolution: version 4

– Add more metaservers and blockservers
– Blockservers do not access DB directly; they send RPCs to metaservers
– Add a memory cache (memcache) in front of the database to avoid scaling

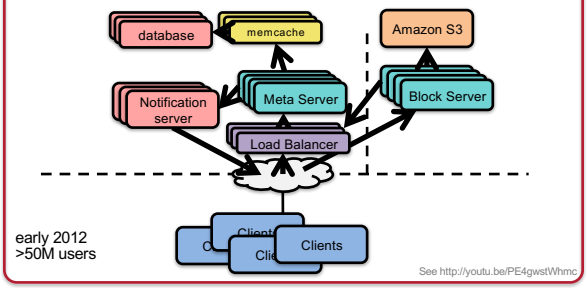


late 2008
~100k users

See http://youtu.be/PE4gwstWhmc

Dropbox: architecture evolution: version 5

- 10s of millions of clients – Clients have to connect before getting notifications
- Add 2-level hierarchy to notification servers: ~1 million connections/server



early 2012
>50M users

See http://youtu.be/PE4gwstWhmc

November 2, 2016 © 2014-2016 Paul Krzyzanowski 13

The End

November 2, 2016 © 2016 Paul Krzyzanowski 14