Distributed Systems

09r. Assignment 6 review

Paul Krzyzanowski
Rutgers University
Fall 2014
Assignment 6 Review
Two major roles: master election and write ordering.

- Chubby is a set of replicated servers (usually 5)

1. One is elected as a master using Paxos
   - The master lease lasts for several seconds
   - After that, Paxos is used to agree on a new master
     • Usually the same one unless it’s dead.

2. All read/write requests go to the master
   - Master propagates any write data to all replicas
   - Paxos is used to ensure total order
Question 2

Fine-grain locking refers to grabbing locks for individual data structures or fields. A coarse grained lock refers to locking larger structures, such as entire files or tables. Explain the difference between coarse-grained lock use and fine-grained lock use and how that helps Chubby achieve scale.

- **Fine-grained locks**
  - Might be held for a short duration, seconds or less.

- **Coarse grained locks**
  - Might be held for hours or days.
  - They are acquired rarely compared to fine-grained locks.
  - Hence, a server can handle more clients.
Question 3

Chubby clients cache file data and metadata. Explain how consistency is managed.

- The client cache is write-through
- Invalidations are sent by the master, which keeps a list of what each client may be caching

- See section 2.7 of the Chubby paper
Question 4

What turned out to be the most popular use for Chubby?

- Functioning as a name server
  - Look up values for a given name

- See section 4.3 of the Chubby paper; first sentence
Question 5

Explain the role of the GFS master.

• The master maintains all file system metadata

• This includes:
  – the namespace
  – access control information
  – the mapping from files to chunks
  – the current locations of chunks

• It also controls system-wide activities:
  – Chunk lease management
  – Garbage collection of orphaned chunks
  – Chunk migration between chunkservers

• See section 2.3 of the GFS paper
What are three advantages of using a large chunk size in GFS?

1. It reduces clients’ need to interact with the master because reads and writes on the same chunk require only one initial request to the master for chunk location information.

2. Since on a large chunk, a client is more likely to perform many operations on a given chunk, it can reduce network overhead by keeping a persistent TCP connection to the chunkservers over an extended period of time.

3. It reduces the size of the metadata stored on the master.

• See section 2.5 of the GFS paper
What are the three factors the GFS master uses to decide which chunkservers should hold replicated copies of a chunk?

1. Place new replicas on chunkservers with below-average disk space utilization.

2. Limit the number of “recent” creations on each chunkserver. Although creation itself is cheap, it reliably predicts imminent heavy write traffic because chunks are created when demanded by writes.

3. Spread replicas of a chunk across racks.

• See section 4.3 of the GFS paper
Question 8

As Dropbox's design evolved,
a. why did Dropbox split the original web server into two web servers? [What was the function of each]
b. Why were notification servers added?
c. Why was RPC-based communication added to the block servers instead of having them talk to the database?

a. 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.
b. Notification servers were added to not require clients to poll the server.
c. Avoids multiple round-trip calls from the block server to the database.
The End