**Chubby Lock Service for loosely coupled Distributed Systems**

**Introduction:**

Chubby is a lock service that allows clients to synchronize to their activities and stay informed about the basic environment. Reliability and availability are the key areas focused in this service. It’s client interface is similar to basic file system that performs whole read/write with advisory locks, sending notification for file modification. Google’s BigTable is most benefited out of the chubby service as availability of systems were managed efficiently.

**Design**:

* **Rationale:**

1. Choosing a master using a lock service is simple and easy to do when compared to the conventional way in which we use a consensus protocol, particularly solving compatibility issues.
2. Chubby could be used in place of a name server for client to server data read/write. Doing this saves client caching, client dependency on servers as well as a need for choosing a cache timeout (DNS).
3. Programmers are familiar with lock-based interface and so this could be used on distributed systems for building a reliable decision making in distributed system.

* **System Structure:** Chubby uses a set of servers called as chubby cell and a chubby library that mediates between clients and chubby cell. The servers in chubby cell elect a master unanimously for a period of time called a master lease. They carry on operations performed as directed by master. If one of the replicas fail (in the chubby cell), the master updates its table with an active replica to serve information.
* **Files, Directories and handles:** File system interface of chubby is similar but simpler to UNIX.
* This file system interface is used as APIs in other file systems like GFS.
* **Locks and Sequencers:**  Chubby uses advisory locks. Clients can access the file in read/ write mode. If more than one client is accessing the file for write mode, a lock is held preventing the write access. Introducing a sequence numbers is more complex in a distributed system and so chubby uses a set of sequence numbers for those interactions that make use of locks.
* **Events:** Chubby clients could be notified on event changes, if they subscribe for events. The chubby library takes care of sending an up-call to its clients after the action is performed
* **Caching:** To mitigate traffic, chubby clients cache file data, node meta data. When a change has to be performed, the master issues a “invalidate cache” call to all its replicas. After the call is acknowledge, the master proceeds to modify the data.
* **Sessions and KeepAlives:** Chubby clients establish a session; the first time they make contact with the cell. This is kept valid by checking the session status using calls sent to the clients, called as KeepAlives. KeepAlives are also used for passing events and cache invalidations, that makes clients mandatory to acknowledge cache invalidations to refresh session lease.
* **Failover:** When a master fails, the in-memory state of all session, handles and locks are lost. Once a new master is elected, clients contact the new master for information.
* **Backups:** For every few hours, the master provides a snapshot of database to a GFS file server, located at a different building. This ensures impact prevention of building damage/ location related issues.
* **Database Implementation:** Initial version of chubby used Berkeley DB.

**Mechanism of Scaling:**  Chubby uses a number of proxies and partitioning to handle huge number of clients

**Lessons Learned:** Developers rarely consider availability as a concern; The Single fail over has consequence of about hundred times the conventional one both in time as well as number of machines affected. However, observed availability of for a given client is less than that for a client’s local chubby cell. Hence, clients hardly feel a difference between service outage and availability.

Fine grain locking was not a necessity, as avoiding the unnecessary communication that happens in the chubby cells through implementing the library provides a coarse grain locking.

**Related Works:** Chubby is built with ideas from different resources. The concept of sessions and tokens are derived from Echo, VMS uses lock service system; Similar to VMS caching model is chubby’s API, that uses a file system like name space as more convenient to just files, is based on file system model.