SUMMARY:
Michael Burrows.
The Chubby lock service for loosely-coupled distributed systems.
In Proc. of the Symp. on Operating System Design and Implementation (OSDI'06), pages 335-350, 2006.

DATE: 18 January 2010

The paper introduces a centralized lock service called "Chubby". Google developers provided this service for synchronizing clients' activities in loosely-coupled distributed systems. The main goal of Chubby is to provide reliability and availability. Performance and storage capacity are considered as secondary objectives. Although Chubby was originally designed as a lock service, it can be used (and is actually used in Google) as a name service as well.

The paper first motivates the need for a centralized lock service. Then it describes the structure of the proposed service. A Chubby instance (a.k.a. a Chubby cell) consists of a small set of servers (usually five), Chubby files and directories. Chubby servers are known as replicas. The replicas in a Chubby cell use a distributed consensus protocol for electing a master. Each Chubby file and directory, called a node, can operate as a lock. Each node contains several meta-data, e.g., access control lists (ACLs) used to control reading and writing for the node. When Clients open nodes, handles are created which are similar to UNIX file descriptors. There are several calls that act on handles. For example, a client handle can hold a lock in an exclusive (writer) mode or a shared (reader) mode. Chubby uses advisory and not mandatory locks and the paper justifies this choice. Chubby clients cache file data, node meta-data, open handles, and locks. The caching protocol invalidates the entries after a modification rather than updating them. Between a Chubby client and a Chubby cell a relationship called a Chubby session is created which is survived by KeepAlive RPC calls. Each session is associated with an interval of time called session lease and can be extended by the master. The client can request an extension of the session lease by sending a KeepAlive call to the master.

The paper describes some techniques such as proxies and partitioning for further scaling of Chubby. Furthermore, it provides a comparison between Chubby and Boxwood's lock server. There are various differences between the two systems since they are designed for different purposes.

SUMMARIZED BY: Somayyeh Zangooei