

# DATABASE SEMINAR REPORT

Rahul Malviya

## Overview:

This report describes two papers, **Bigtable: A Distributed Storage System for Structured Data** and **The Chubby Lock Service for Loosely-Coupled Distributed Systems**. Bigtable is distributed storage system for managing structured data and Chubby Lock Service provide coarse-grained locking for loosely coupled distributed systems at Google Inc.

The authors of **Bigtable** starts with a brief introduction on how Bigtable is used by various application at Google Inc which have different requirements as compared to latency, throughput, availability and scalability and how Bigtable fulfills all these goals providing wide applicability, high performance and high availability across various different applications. They then further discuss the underlying data model of the Bigtable providing examples of each component and why this particular design is considered. They also took few live examples from Google application and describe how this data model fits into various applications. In the next section they describe the Sawzall based client API and its features that is used by all Google's application using Bigtable. In this section they also provide a sample code in C++ and tell how it uses the API functions. In the next section the authors describes how Bigtable uses the already existing Google File System, SSTable file format and Chubby lock Service. Furthermore in the next section the authors provide the details about the implementation of Bigtable, its basic component and their functionality in achieving the services provided by the Bigtable to the other applications. In this section they provide complete description of all the operations and how they are performed on Bigtable. At the end of this section they provide few techniques to overcome few shortcomings of the Bigtable. In next section the authors proposes few refinements to improve the performance of Bigtable. The authors further tell how Bigtable is used by various applications at Google stating examples of the Google Analytics, Google earth and Google personalized search. They also measures performance in experimental environment and discuss the performance of Bigtable across various benchmarks defined. They finally describe works related to Bigtable and their correlation with the Bigtable. Finally in the end they provide conclusion and suggests few improvements that can be implemented as future work.

The next paper is about **Chubby Lock Service** which is a distributed locking service intended to use with loosely-coupled distributed systems. The authors of this paper also start with a brief introduction and tell that the main aim of the Chubby is to provide reliability and availability to a relatively large number of clients but throughput and storage is secondary and provides examples of few clients at Google who uses this locking service like Bigtable and Google File System. Then in the following section the authors give details about the design of Chubby discussing why it is used as a locking service rather than client library stating various advantages of a lock service in comparison with a client library. This section also discussed the Paxos algorithm, which the core of almost most distributed system requiring consensus across all the nodes. Then they further discuss the Chubby system structure along with its components and provide details about their functionality and features and how they accomplish Chubby services. In the next few sections the authors describes the complete cycle of a chubby cell and how the various components contributes to accomplish clients' request. In this section the authors also discuss the comparison with various other methods that can be used handle the same task but why a particular method is preferred in Chubby. In the next section the authors describe the API that a Chubby Service provides along with various functions in the API and their functionality. In the next part the authors describes few problems in a Chubby system and how these are handled in a Chubby cell providing examples in each case. Further in the next section they provide few

implementations of Chubby Service. In next section they discuss the scaling techniques for a chubby service and also describes few design errors with the existing implementation and their solutions as well. They also provided few real time statistic information of a running Chubby Service and discussed various parameters described in the statistics. At the end the authors compare Chubby Lock Service with related work and end up with providing conclusion and scope of future work in this field.

#### Comments:

Both the paper deeply discussed the design, implementation and implication of the technique and also provided good number of examples at each stage to get a real understanding about the topic. Bigtable underlying uses Chubby Service and paper describes how Chubby is used to accomplish the services provided by the Bigtable.

#### Questions: Bigtable

1. How replicas are maintained in Bigtable?

Ans: The consensus across replicas is maintained in Bigtable by using the reliable service of Chubby Service which in turn uses Paxos algorithm for distributed systems to maintain the replicas.

2. How Bigtable is column oriented?

Ans: In Bigtable data is accessed according to the column in a particular row i.e. all the transactions are pertaining to a single row and data is being read or written of various columns in that row. So a Bigtable can be said as both row-oriented and column-oriented database.

3. What happens when the master dies?

Ans: When a master dies in the Bigtable cell then a new master is elected and slide number 28 discusses the steps performed the new master to come a stable state to handle the Bigtable clients.

#### Questions: Chubby Service

1. How is Chubby Service used in the Bigtable?

Ans: The Chubby Service is used by Bigtable to keep track of tablet servers, master selection, bootstrap location of Bigtable, to store Bigtable schema information and to store access control lists.

2. How a client does know the location of a new master when the old master dies?

Ans: Each client has an embedded library in them, which contains the function to get the DNS mappings of the replicas. So when a master dies the client communicates with one of the replicas and the replica replies the client back with the new master's location.

3. What is Paxos and how does it works?

Ans: Paxos is a protocol in distributed network that solves the consensus problem across all the replicas and all the details are mentioned in slides 03-11.