HW3 PART1 REPORT

**The Google File System(Summary)**

the Google File System, a scalable distributed file system for large distributed data-intensive applications in order to meet the rapidly growing demands of Google’s data processing needs

This paper presents file system interface (GFS master, chunk/chunkServer, Client) extensions designed to support distributed applications from several aspects including fault tolerance, scalability, data storage, clustered storage. System provides fault tolerance by constant monitoring, replicating crucial data, and fast and automatic recovery. Chunk replication allows us to tolerate chunkserver failures. GFS has successfully met our storage needs and is widely used within Google as the storage platform for research and development as well as production data processing.

**MapReduce: Simplified Data Processing on Large Clusters (Summary)**

This article concludes 3 reasons that MapReduece programming has been successfully used for different purpose. First, the model is easy to use, even for programmers without experience with parallel and distributed systems, since it hides the details of parallelization, fault-tolerance, locality optimization, and load balancing. Second, a large variety of problems are easily expressible as MapReduce computations. Third, the implementation of scales to large clusters of machines makes efficient use of these machine resources and therefore is suitable for use on many of the large computational problems encountered at Google.

Also First, restricting the programming model makes it easy to parallelize and distribute computations and to make such computations fault-tolerant. Second, network bandwidth is a scarce resource. Third, redundant execution can be used to reduce the impact of slow machines, and to handle ma- chine failures and data loss.

**The Chubby lock service for loosely-coupled distributed systems (Summary)**

Chubby is a distributed lock service intended for coarse- grained synchronization of activities within Google’s distributed systems that have meshed well: distributed consensus among a few replicas for fault tolerance, consistent client-side caching to reduce server load while retaining simple semantics, timely notification of updates, and a familiar file system inter- face. We use caching, protocol-conversion servers, and simple load adaptation to allow it scale to tens of thou- sands of client processes per Chubby instance. We expect to scale it further via proxies and partitioning. Chubby has become Google’s primary internal name service; it is a common rendezvous mechanism for systems such as MapReduce; the storage systems GFS and Bigtable use Chubby to elect a primary from redundant replicas; and it is a standard repository for files that require high availability, such as access control lists.

**Bigtable: A Distributed Storage System for Structured Data(summary)**

This report described Bigtable, a distributed system for storing structured data at Google. We are in the process of implementing several additional Bigtable features, such as support for secondary indices and infrastructure for building cross-data-center replicated Bigtables with multiple master replicas. We have also begun deploying Bigtable as a service to product groups, so that individual groups do not need to maintain their own clusters. In addition, our control over Bigtable’s implementation, and the other Google infrastructure upon which Bigtable depends, means that we can remove bottlenecks and inefficiencies as they arise.