Indian Institute Of Technology, Delhi

COL733: Cloud Computing Technology Fundamentals

Instructor: S. C. Gupta

Reading Assignment 1: Google File System

Report

August 20, 2019

Submitted To:
S. C. Gupta
Professor
Computer Science Department

Submitted By: (Group 4)
Shubham
2016CS10371

1. Hadoop Distributed File System

The Hadoop Distributed File System (HDFS) is the primary data storage system used by Hadoop applications. HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware. The used memory model in HDFS is writing-once-read-many to make execution easier.

1.1. Architecture: A distributed file system offers performance access to information across extremely scalable Hadoop nodes which can be used through the master-slave architecture of HDFS.

- In a master slave design, Hadoop has central **namenodes** with metadata of the entire file, and DataNodes that store the data. This is a central part of the distributed file system.
- NameNode performs all metadata actions, such as **moving** or **rename** files, while DataNodes serve the client with the read / write requests to the data.

1.2. Fault Tolerance: It is constructed of fault-tolerant commodity hardware. The essence of commodity hardware is that the faulty device can be exchanged if anything is failing.

- Each piece of the data is replicated or copied multiple times (Default 3 times) by the file system and copied to each node, placing at least one copy in a different server computer than the others.
- Each Hadoop cluster, during its initial incarnation, consisted of one Name Node which manages the operation of file systems and supports data nodes that manages data storage on single nodes.
- NameNode is the only breakdown point on Hadoop v1. The entire Hadoop cluster stops functioning is NameNode fails. However, in Hadoop v2 covers this scenario with Active and Standby NameNodes (AvatarNode) which is configured on two different machines

2. Google File System

Google File System is a scalable distributed file system (DFS) developed to meet growing Google data processing needs. GFS gives linked nodes fault tolerance and storage schemes composed of low cost hardware. GFS provides resource intensive reading and streaming projects in large numbers of information with rare writings.

2.1. Architecture: Similar to HDFS, GFS also follows master-slave architecture.

- A shared, fault-tolerant file system is the Google File System. There is an essential NameNode that saves file metadata in a look-up table, matching the full path names of the files to the 64-bit meta-data including chunk id along with read-write locks.
- The files will be divided into chunks, saved and copied to chunk servers on local disks. Each file has a corresponding replication factor that determines how many copies it must always be available on the standard 3 file scheme and distributed among different chunk servers.

2.2. Fault Tolerance: GFS has a high level of system-wide fault tolerance using methods like load balance, control overview and many other functionalities.

- A chunk failure can't lower the data chunks anytime, as there are alternate replicas in the network for all the chunks in the server whose position is in the metadata.
- Metadata is tolerated by shadow masters which keeps a replica of the metadata in the file system and provides read only access to keep copies up to date.

3. Google File System vs Hadoop Distributed File System

Features	Google File System	Hadoop DFS
File Size	The GFS is divided into 64 MB chunks, further divided into 64 KB blocks, and a checkSum is provided for each block.	The HDFS is divided into 128 MB blocks. The replica is contained in NameNode in two files, one for data and one for checkSum.
Availability	GFS is developed to meet growing processing needs of google's extensive data.	HDFS is public for use and supports third party file systems
Communication	GFS utilizes TCP / IP-based communication protocol.	HDFS utilizes TCP / IP RPC-based interaction.
Platform Dependence	GFS is built in C, C++ and is centered on Linux.	HDFS is produced in Java and autonomous of the platform.
Recovery	Metadata backed up by shadow masters, but can not change it before retrieval.	Metadata has several copies, however, action is required if NameNode falls - Hadoop v2 provided solution to this.

4. References

HDFS: R01_Hadoop.pdf

GFS: R02_GFS.pdf