A Study of Comparatively Analysis for HDFS and Google File System towards to Handle Big Data

Rajesh R Savaliya[1], Dr. Akash Saxena[2]

[1]Research Scholor, Rai University, Vill. Saroda, Tal. Dholka Dist. Ahmedabad, Gujatat-382 260
[2]PHD Guide, Rai University, Vill. Saroda, Tal. Dholka Dist. Ahmedabad, Gujatat-382 260

***

ABSTRACT - BIG-DATA handling and management is the current requirement of software development industry in face of software developments now a day. It is becomes very necessary for software development industry to store large amount of Data and retrieves the only required information from the stored large scale data in the system. This paper presents the comparison of two similar distributed file working and handling parameters towards frameworks which is used to work with storage of Big-Data in hadoop distributed file system and Google file system. This paper also includes the Map Reduse Structure which common model used by both HDFS and GFS to handle the Big Data. These analyses will useful for understanding the frame work and highlight the features those are common and difference between Hadoop DFS and GFS.

KEYWORDS: HDFS, GFS, NameNode, MasterNode, DataNode, ChunkServer, Big-Data.

1. INTRODUCTION

Big-Data is the keyword which is used to describe the large amount of data, produced by electronic transactions as well as social media all over the world now a day. Hadoop Distributed File System and Google File System have been developed to implement and handle large amount of data and provide high throughputs [1]. Big data challenges are complexity as well as velocity, variety, volume of data and are included insight into consideration in the development of HDFS and GFS to store, maintain and retrieve the large amount of Big-Data currently generated in field of IT [2]. First Google was developed and publish in articles distributed file system in the world of IT that is GFS, then after Apache open-source was implement DFS as an Hadoop DFS based on Google's implementations. Differences and similarities in the both type of file system have been made based on so many parameters, levels and different criteria to handle the big-data. The main important aim of HDFS and GFS ware build for to work with large amount of data file coming from different terminals in various formats and large scale data size (in TB or peta byte) distributed around hundreds of storage disks available for commodity hardware. Both HDFS and GFS are developing to handle big-data of different formats [3].

1.1 Hadoop Distributed File System Framework

HDFS is the Hadoop Distributed File system which is an open source file distributed and large scale data file handling framework and it is design by Apache. Currently so many network based application development environment using this concepts such as Whatup, Facebook, Amazon. HDFS and MapReduce are core components of Hadoop system [4]. HDFS is the Distributed File system which is used to handle the storage of large amount of file data in the DataNode[5].

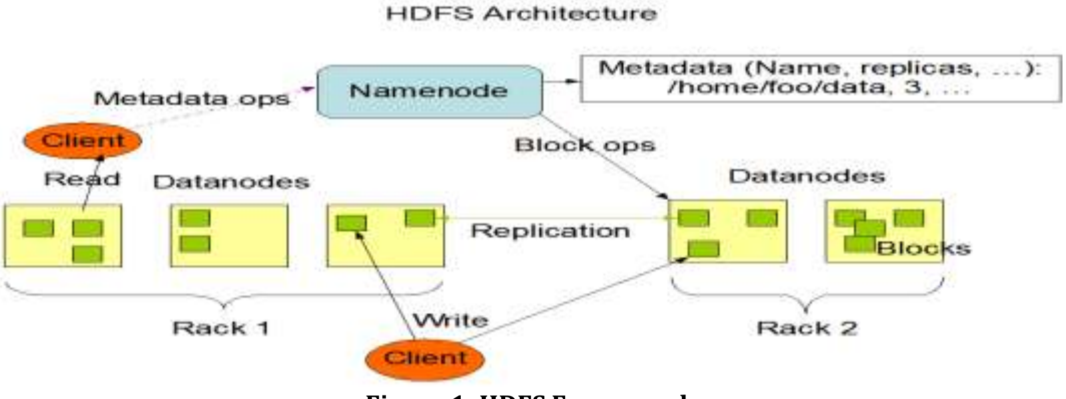

Figure 1: HDFS Framework

Hadoop Distributed File system is a scalable and platform independent system develop in Java.The HDFS is a master-slaver distributed framework specially designed to work with storage of large scale data file in the large cluster which has DataNode and NameNode. The NameNode is work as a master server to handle and store the metadata for large amount of data file in the HDFS. NameNode is also used to manage the data file access through the different clients. The DataNode is used to handle the storage management of large scale data file. Now the main role of the MapReduce is the decomposition of tasks, moniter the task and then integrates the final results. MapReduse programming techniques successfully implemented by the Google to store and process the big amount of data files [6].

1.2 Google File System Framework

Google File System is the network and node based framework. GFS is the based on scalable, reliable, availability, fault tolerance and distributed file system structure design by Google to handle the large amount of data files. Google File System is made to storage system on low cost commodity hardware. GFS is used to optimize the large amount of data storage. GFS develop to handle the big-data stored in hierarchical directories structure. Namespace means metadata, data access control handle by the master, that will deals with and monitors update status of each and every chunk server based on particular time intervals.

Google File System has node cluster with single master and multiple chunk servers which are continuously accessed by different client. In GSF chunk server is used to store data as a Linux files on local disks and that stored data will be divided into (64 MB) size's chunk. Stored data which are minimum three times replicated on the network. The large size chunk is very helpful to reduce network traffic or overhead. GFS has larges clusters more than 1000 nodes of 300 TB size disk storage capacities and it will continuous access by large number of clients [7].

1.2.1 GFS has importance features like,

- Fault tolerance
- Scalability
- Reliability
- High Availability
- Data Replication
- Metadata management
- Automatic data recovery
- High aggregate throughput
- Reduced client and master transaction using of large size chunk server

1. 2. 2 Frameworks of GFS

Google File System is a master/chunk server communication framework. GFS consists of only single master with multiple number of chunk-server. Multiple Clients can easily access both master as well as chunkserver. The by default chunk size is 64MB and data file will be divided into the number chunk of fixed size. Master has 64bit pointer using which master will manage each chunk [7]. Reliability fulfilled using each chunk is replicate on multiples chunkserver. There are three time replicas created by default in GFS.

1.2.2.1 Master

Master is used to handle namespace means all the metadata to maintain the bigdata file. Master will keep the track of location of each replica chunk and periodically provide the information to each chunkserver. Master is also responsible handle to less than 64 byte metadata for each 64MB chunk [10]. It will also responsible to collect the information for each chunkserver and avoid fragmentation using garbage collection technique. Master acknowledges the future request to the client.

1.2.2.2 Client

The working of the client will be responsible to ask the master for which chunkserver to refer for work. Client will create chunk index using name and the byte offset. Client also ensures future request interaction in between master and client.

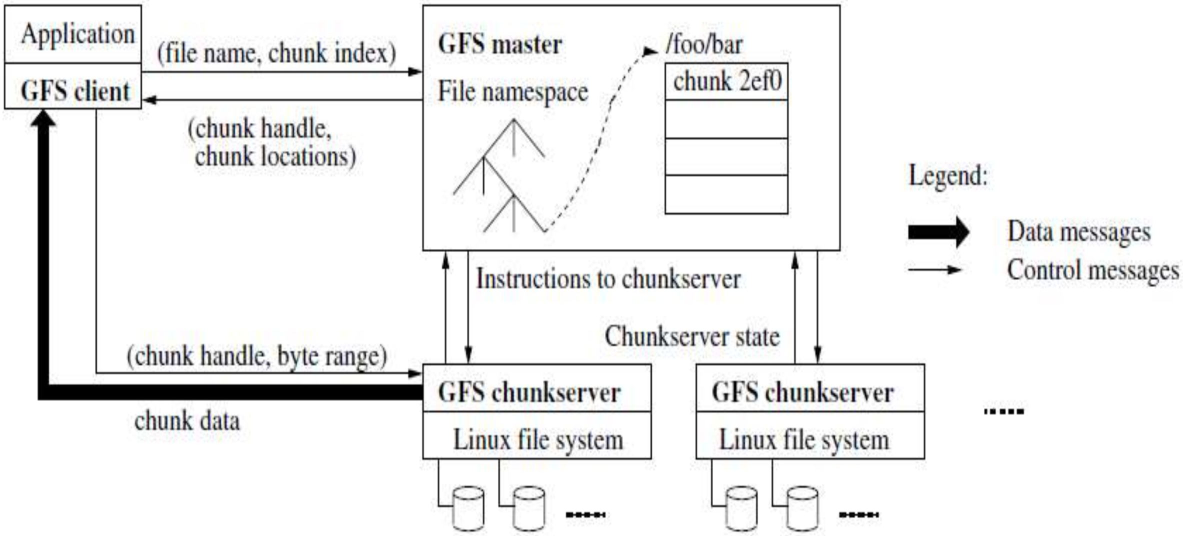

Figure 2: GFS Framework

1.2.2.3 Snapshot

The role of a snapshot is an internal function of Google File System that ensures consistency control and it creates a copy of a directory or file immediately. Snapshot mostly used to create checkpoints of current state for commit so that rollback later.

1.2.2.4 Data Integrity

GFS cluster consists thousands of machines so it will help to avoid the machine failures or loss of data. For avoid this problem each chunkserver maintain its own copy.

1.2.2.5 Garbage collection

Instead of instantly reclaiming or free up the unused physical memory storage space after a file or a chunk is deleted from the system, for that GFS apply a lazy action strategy of Garbage Collection. This approach ensures that system is more reliable and simple.

2. COMPARATIVELY ANALYSIS OF HDFS WITH GFS

Key Point	HDFS Framework	GFS Framework
Objective	Main objective of HDFS to handle the Big-Data	Main objective of HDFS to handle the Big-Data
Language used to Develop	Java Language	C, CPP Language
Implemented by	Open source community, Yahoo, Facebook, IBM	Google
Platform	Work on Cross-platform	Work on Linux
License by	Apache	Proprietary or design by google for its own used.
Files Management	HDFS supports a traditional hierarchical directories data structure [9].	GFS supports a hierarchical directories data structure and access by path names [9].
Types of Nodes used	NameNode and DataNode	Chunk-server and MasterNode

Hardware used	Commodity Hardware or Server	Commodity Hardware or Server
Append Opration	Only supports append operation	supports append operation and we can also append base on offset.
Database Files	Hbase	Bigtable is the database
Delete Opration and Garbage Collection	First, deleted files are renamed and store in particular folder then finally remove using garbage collection method.	GFS has unique garbage collection method in which we cannot reclaiminstantly. It will rename the namespace It will delete after the 3 days during the second scaned.
Default size	HDFS has by default DataNode size 128 MB but it can be change by the user	GFS has by default chunk size 64 MB but it can be change by the user
Snapshots	HDFS allowed upto 65536 snapshots for each directory in HDFS 2.	In GFS Each directories and files can be snapshotted.
Meta-Data	Meta-Data information managed by NameNode.	Meta-Data information managed by MasterNode.
Data Integrity	Data Integrity maintain in between NameNode and DataNode.	Data Integrity maintain in between MasterNode and Chunk-Server.
Replication	There are two time replicas created by default in GFS [10].	There are three time replicas created by default in GFS [10].
Communication	Pipelining is used to data transfer over the TCP protocol.	RPC based protocol used on top of TCP\IP.
Cache management	HDFS provide the distributed cache facility using Mapreduse framework	GFS does not provide the cache facility

3. CONCLUSION

From the above way, it is concluded that this paper describes an insight of comparatively studies towards two most powerful distributed big-data processing Framework which are Hadoop Distributed File System and Google File System. This studies was performed to observe the performance for both HDFS and GFS big-data transactions such as storing as well as retrieving of large scale data file. Finally, this can be concludes that successfully manage network maintenance, power failures, hard drive failures, router failures, misconfiguration, etc. GFS provide the better Garbage collection, Replication and file management as compare as HDFS.

REFERENCES

1) http://hadoop.apache.org.[Accessed: Oct. 11, 2018]

2) http://en.wikipedia.org/wiki/Big_data .[Accessed: Oct. 19, 2018]

3) Gemayel. N, "Analyzing Google File System and Hadoop Distributed File System" Research Journal of Information Technology , PP. 67-74, 15 September 2016.

4) Sager, S. Lad, Naveen Kumar, Dr. S.D. Joshi, "Comparison study on Hadoop's HDFS with Lustre File System", International Journal of Scientific Engineering and Applied Science, Vol. 1, Issue-8, PP. 491-194, November 2015.

5) R.Vijayakumari, R.Kirankumar and K.Gangadhara Rao, "Comparative analysis of Google File System and Hadoop Distributed File System", International Journal of Advanced Trends in Computer Science and Engineering, Vol.1, PP. 553– 558, 24-25 February 2014.

6) Ameya Daphalapuraka, Manali Shimpi and Priya Newalkar, "Mapreduse & Comparison of HDFS And GFS", International Journal of Engineering And Computer Science, Vol. 1, Issue-8, PP. 8321- 8325 September 2014.

7) Giacinto, Donvito, Giovanni Marzulli2 and Domenico Diacono, "Testing of several distributed file-systems (HDFS, Ceph and GlusterFS) for supporting the HEP experiments analysis", International Conference on Computing in High Energy and Nuclear Physics, PP. 1-7, 2014.

8) Dr.A.P Mittal, Dr. Vanita Jain and Tanuj Ahuja, "Google File System and Hadoop Distributed File System- An Analogy ", International Journal of Innovations & Advancement in Computer Science , Vol. 4, PP. 626-636, March 2015.

9) Monali Mavani, "Comparative Analisis of Andrew File System and Hadoop Diatributed File System", Lecture Note on Software Engineerin, Vol. 4 No. 2, PP. 122-125, May 2013.

10) Yuval Carmel, " HDFS Vs. GFS", Topics in Storage System-Spring , PP. 20-31, 2013.

BIOGRAPHIES

Authors' Profile

Mr. Rajeshkumar Rameshbhai Savaliya from Ambaba Commerce College, MIBM & DICA Sabargam and master degree in Master of science and Information Technologies(M.Sc-IT) from Veer Narmad South Gujarat University.Rajesh R Savaliya has teaching as well programming experience and PHD Pursuing from RAI University.

Co-Authors' Profile

Dr. Akash Saxena PhD Guide from Rai University.