International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 04 Issue: 07 | July -2020 ISSN: 2582-3930

Research on Cloud Data Storage Technology and Its Architecture Implementation

Supriyasachan[1st] ,Asst.Prof. Rahul Kumar Chawda

[1,2]Dept. of Computer Science, Kalinga University, Naya Raipur, Raipur, Chhattisgarh 492101, India

Abstract -The idea of distributed computing turns out to be increasingly more famous in most recent years. Information stockpiling is a significant and important exploration field in distributed computing. This paper presents the idea of distributed computing and distributed storage just as the design of distributed storage right off the bat. At that point we break down the cloud information stockpiling innovation - GFS(Google Record System)/HDFS(Hadoop Distributed File System) towards solid undertaking models. In the last part, we outline how to improve the customary record stockpiling strategy dependent on eye OS Web working framework which figures it out document dispersed capacity and shortcoming open minded control however HDFS innovation of Hadoop.

Keywords: Cloud Computing; Cloud Storage; Web Operating System; Distributed File System;

1. Introduction

In most recent years, the idea of distributed computing turns out to be increasingly well known. Distributed computing as a new plan of action is created from disseminated preparing, equal handling and network processing. At present, Google, Amazon, IBM, Microsoft, Sun and other IT mammoths are for the most part looking to create cloud registering advances and items. For instance, Google has been devoted to advancing application motors dependent on the strategies of GFS [1] (Google File System), MapReduce [2], BigTable[3], etc, which give clients strategies and intends to process gigantic information. In this paper, we present the idea of distributed computing and distributed storage just as the engineering of distributed storage initially, break down the cloud information stockpiling innovation—GFS and HDFS (Hadoop Distributed File System) under the particularcases of enterprises, and build the cloud storage architecture through eyeOS Web operating system in our computer.

2. Cloud computing and cloud storage

2.1. Cloud computing definition

Distributed computing emerges from the mix of the customary PC innovation and system innovation, for example, network figuring, dispersed registering, equal processing, utility figuring, virtualization. One of the center idea of distributed computing is decreasing the preparing trouble on client's terminals through consistently improving the mists' dealing with limit. In the long run client's terminals are rearranged into a straightforward info and yield gadgets. Clients can utilize the incredible registering and preparing work on mists and they can arrange their administration from the cloud as per their own needs.

2.2. Cloud storage definition and it's architecture

Distributed storage is a framework that gives capacities, for example, information stockpiling and business get to. It amasses countless various sorts of capacity gadgets through the application programming which are in view of the elements of the bunch applications, matrix procedures, appropriated document frameworks, and so forth. Cloud capacity can be just comprehended as the capacity in distributed computing, and furthermore can be viewed as a distributed computing framework outfitted with huge limit stockpiling. Distributed storage framework design essentially incorporates capacity layer, fundamental administration layer, application interface layer and access layer .

3. Cloud storage technology of enterprises

3.1. GFS [1]

1) System Architecture

A GFS bunch comprises of a solitary ace, mutiplechunkservers and mutiple customers, as appeared in Figure 1(a). Each of these is ordinarily a ware Linux [1]

. • GFS Master: Master deals with all document framework metadata and the records registry structure. GFS employments single ace approach which implies in a similar time just one ace offering types of assistance with the goal that it can stay away from additional expenses for organizing between different bosses simultaneously. A customer associates with the ace just for metadata, and communicates with the chunkservers straightforwardly for every single other datum.

• Chunkserver: GFS documents are partitioned into fixed-size lumps put away on each chunkserver and the default square size is 64M. Each piece is distinguished by a permanent and universally extraordinary 64 piece lump handle allocated by the ace when the lump is made. Each square is imitated on three chunkservers. Clients can set diverse replication levels for every locales of the record namespace. As appeared in Figure 1(a), there are four chunkservers and five pieces as C0-C4. Each lump is saved money on three chunkservers. • Client: GFS customer code connected into every application executes the record framework API and speaks with the ace and chunkservers to peruse or compose the ace for metadata tasks, in any case, all information bearing correspondence goes straightforwardly to the chunk servers[1].

2) Workflow

As appeared in Figure 1(a), flimsy strong lines speak to the control data among customers and ace or on the other hand among ace and chunkservers, thick strong lines speak to the information correspondence between chunkservers and customer, ran lines demonstrate the control data among customers and chunkservers.

Firstly, clients compute chunk index from files structure and chunk size, then send file name and chunk index to master (mark①). Secondly, master sends chunk handle and chunk locations to clients (mark②). Thirdly, clients send chunk handle and byte range to the nearest chunkserver (mark⑤). Finally chunkserver sends data to client (mark⑥).Once clients get chunk locations from master, clients do not interact with master any more. Master does not permanently save the mapping from chunkserver to chunk. Instead, it asks each chunkserver about its chunks at master startup or whenever a chunkserver joins the cluster (③④). The master periodically communicates with each chunkserver in HeartBeat message to give it instructions and collect its state (③④).

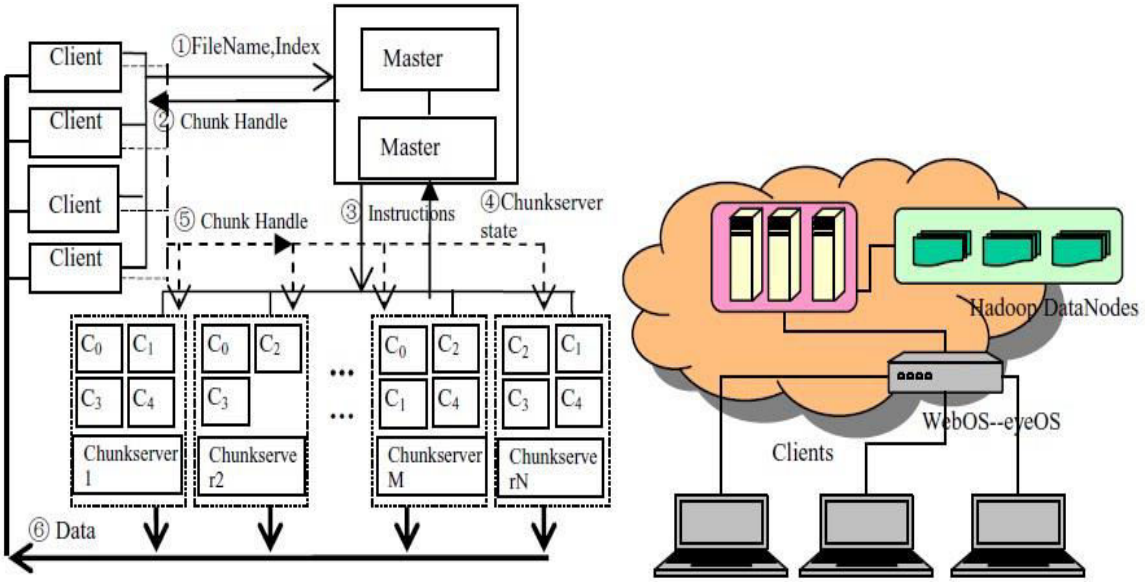

Fig.1.(a) GFS Architecture; (b) System Architecture

3.2. HDFS

Hadoop is facilitated by the Apache Software Foundation, which offers help for a network of open source programming ventures. In spite of the fact that Hadoop is most popular for Map Reduce and its conveyed record framework (HDFS), the different sub projects offer integral types of assistance, or expand on the center to include higher level reflections. The itemized substance allude to record [4]. The complete name of HDFS [5] is Hadoop Distributed File System. HDFS is run on enormous groups of item equipment and resembles GFS of Google. The engineering of HDFS is ace/slave and a HDFS bunch has one namenode and different datanodes. Namenode is the focal server, proportionate to ace in GFS. It is liable for the namespace activity of record frameworks. Data node is like chunkserver of GFS which is liable for overseeing stockpiling on datanodes, making square, erasing square, replicating square and so forth. The records in HDFS are partitioned into one or numerous squares which are put away in data nodes.

Name node and data nodes can be run on the minimal effort Linux PC. HDFS is created by java language.

4. Cloud storage architecture based on Hadoop

4.1. EyeOS

Eye OS is a web work area condition with office programming and individual data the board frameworks, and it empowers the online stockpiling, versatile office. Report the executives in eye OS is basically put away in a solitary server, without deficiency open minded reinforcement highlight and unwavering quality is poor. Getting to documents is asingle string and access execution isn't high. In this paper, we improve the conventional document stockpiling technique and accomplish record appropriated capacity just as issue open minded control utilizing HDFS innovation .

4.2. System implementation

1） Architecture

The capacity framework we structured is appeared in Figure 1(b) which incorporates customers, web working framework eye OS, cloud server (Name Node), distributed storage place (Data Node).

· Clients: Each customer is just pre-introduced with internet browser and clients sign in this distributed storage through internet browser. Customers are the interface among clients and distributed storage framework. · Web Operating System: Web working framework gets clients' entrance demands, confirms the clients' legitimacy, and collaborates legitimately with the customers. It depends on eye OS which offers an enormous number of applications to clients. Clients can download their necessary applications and accomplish a customized framework. Eys OS is additionally the document get to interface for clients and records can be spared in the distributed storage groups by this interface.

· Cloud server (Cloud Name Node): Cloud stockpiling group dependent on Hadoop incorporates cloud server (Name Node) and distributed storage community (Data Node). Cloud server is the name node in Hadoop which oversees record framework namespace, processes the planning from documents to data nodes, dispenses data nodes to spare document squares, and controls outer customers' entrance.

· Cloud Storage focus (Cloud Data Node): Cloud stockpiling focus is data node in Hadoop. It is responsible for sparing records, acknowledging document dispersed capacity, guaranteeing load adjusting, records issue open minded and so on.

2） Operation Process

Clients' tasks dependent on eye OS are composing documents and understanding records. When perusing a document, we download the document to the nearby PC, at that point handle or show the record utilizing the application programming in web working framework. At the point when the documents are changed and spared, web working framework transfers them to distributed storage framework from neighborhood PC.

• Reading documents process: ①Users sign in the web OS from customer through customers' program and double click a document symbol on the web OS. At that point eye OS demands the record from the Hadoop name node.②Name node finds the related data of documents, and processes the record's area. Data nodes which spared the squares of the document send the squares to the clients.③Clients download the record obstructs from the data nodes and combine these squares into a file.④Applications related with the document in the web working framework auto start and show the document.

· Writing documents process: ①Users sign in web OS from customer's internet browser alter and spare records utilizing the chose application. EyeOS demands transferring records to Hadoop namenode. ②Namenode assigns extra room to datanodes as per the document size and the datanodes' stockpiling condition after it gotten the transferring demand. ③Clients transfer record. Namenode separates it into one or various squares furthermore, spared in the designated datanodes.

5. Experiments

These tests utilize five PCs. Three are utilized as customer, eyeOS, namenode individually and the other two are utilized as datanodes. We accept the datanodes are Da and Db. There are documents named FileX what's more, FileY in Da and Db. The analyses are done when Da and Db are consistently typical. As appeared in Table 1, while making File1, this document is spared in Da and Db simultaneously. While erasing FileY, this document in Da and Db are all erased. These methods pieces of information in invalid datanode will be refreshed consequently when the data node recuperates ordinary and information in datanode are consistently most recent.

We do experiments when Db is always normal but Da is invalid and the results are showed in Line 3 of Table 1. Then when Da recovers normal and the results are shown in Line 4 of Table 1.

•Creating Files: When creating File1, we can find the file in Db but can’t find it in Da. If Da recovers normal this moment, we can also find File1 in Da.

•Deleting Files: When deleting FileY, the FileY can’t be found in Db but can be found in Da. If Da recovers normal now, FileY is deleted from Da immediately.

Table 1.Experiments’ Results

Operation	Creating File1	Deleting FileY
Da(Normal) Db(Normal)	File1 can be found in Da and Db.	FileY are deleted from Da and Db.
Da(Invalid) Db(Normal)	File1 is saved in Db, but isn't saved in Da	FileY is deleted from DB, but is not deleted form Da.
Da(recovers normal after invalid) Db(Normal)	File1 is saved as a duplicate file in Da automatically.	FileY is deleted from Da.

6. Conclusions

Distributed computing is the unavoidable item with the improvement of the web, and it likewise brings morerich applications to the web. Cloud information stockpiling innovation is the center region in distributed computing and tackles the information stockpiling method of cloud condition. In this paper, we present the related ideas of distributed computing and distributed storage. At that point we represent a distributed storage engineering dependent on eye OS web working framework in our PCs. Trials confirmed the framework is well.

References

[1]Sanjay Ghemawat, Howard Gobioff,Shun-Tak Leung. The Google file system[C]. Proceedings of the 19th ACM Symposium on Operating Systems Principles. New York: ACM Press, 2003:29-43.
[2]Jeffrey Dean, Sanjay Ghemawat. MapReduce:Simplied data processing on large clusters[C]. Proceedings of the 6th Symposium on Operating System Design and Implementation. New York: ACM Press. 2004:137-150.
[3]Fay Chang, Jeffrey Dean,et al. Bigtable:A Distributed Storage System for Structured Data[J]. ACM Transactions on Computer Systems. 2008,26(2):1-26.
[4]Tom White. Hadoop:The Definitive Guide[M]. United States of America: O' Reilly Media, Inc. 2009.
[5]DhrubaBorthakur. The Hadoop Distributed File System: Architecture and Design [EB/OL]. (2008-09-02) [2010-08-25]. http://hadoop.apache.org/common/docs/r0.16.0/hdfs_design.html.
[6]Hbase Development Team. HBase: Bigtable-like structured storage for Hadoop HDFS[EB/OL]. (2010-08-10) [2010-08-25]. http://wiki.apache.org/hadoop/Hbase.
[7]Mike Burrows. The chubby lock service for looselycoupled distributed systems[C]. Proceedings of the 7th Symposium on Operating Systems Design and Implementation，2006.