VOL. 10, NO. 7, APRIL 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

HADOOP ARCHITECTURE AND FAULT TOLERANCE BASED HADOOP CLUSTERS IN GEOGRAPHICALLY DISTRIBUTED DATA CENTER

T. Cowsalya and S.R. Mugunthan
SVS College of Engineering, Coimbatore, Tamil Nadu, India
E-Mail: kkousalya@gmail.com

ABSTRACT

In today's epoch of computer science storing and computing data is a very important phase. In recent days even a petabyte and exabytes of data is not adequate for storing large number of databases which contains large data sets. Therefore organizations today use concept called Hadoop which is a software framework of big data in their application. Hadoop is designed to store and process large volume of data sets consistently. While using geographically distributed data centers there may be a chance of data loss due to network link failure and node failures. Hadoop provides high reliability and scalability features. Along with it also afford faults tolerance mechanism by which the system functions properly even after a node in the cluster fails. Faults tolerance is mainly achieved using data replication and Heartbeat messages. Heartbeat messages are like acknowledgement messages between two nodes and Data replication is achieved by creating copies of same data sets into more than one computing nodes. In this paper we describe about the Hadoop architecture and its components and also describe how the fault tolerant mechanism is achieved using data replication and heartbeat messages.

Keywords: hadoop, fault tolerance, HDFS, name node, data node.

INTRODUCTION

Hadoop is a software framework used for big data processing. Hadoop is open source software created by Doug Cutting and Michael J. Cafarella. Hadoop was originally motivated by paper published by Google for their approach to lever their data. Hadoop was developed for distributed processing system and parallel processing system. Big data consists of two types of data called structured and unstructured data. Structured data includes normal data types supported by traditional database systems. Unstructured data includes data types like audio, video, emails etc. With the help of Hadoop there is no limit for storing and computation process. To work with the speed of data generation, data computation has also been increasingly moving from traditional database to distributed systems. Scalability and fault tolerance two major challenges are introduced in these systems with the challenges of computing and managing resources.

We are well known that the data appliances are well worse at doing some important factors, that appliances naturally take a extremely parallel approach to compute the data but make a primary problem of building a hard link between the computing power and data storage..

With growing number of cluster, the chance of hardware and software failures also arises in addition, software bugs or hardware defects that never explain up in a testing atmosphere could also suddenly surface in computing. Fundamentally, failures become the standard pretty than the exception at large scale [15]. The variety of failures includes halt failures due to OS crash, network disconnection, and disk hang, etc [9].

GEOGRAPHICALLY DISTRIBUTED DATA CENTER

The data centers distributed at geographical regions are known as geographically distributed data centers. For example totally Google has 13 data centers over 8 countries and 4 continents.

Figure-1. Google's data center location.

ADOOP ARCHITECTURE

Hadoop consists of two main components called Hadoop Distributed File System and MapReduce.

HADOOP DISTRIBUTED FILE SYSTEM

Hadoop Distributed File System is a distributed or parallel file system which is designed to run on commodity hardware. HDFS is highly fault tolerant. The goals and assumptions of HDFS include Hardware failure, Streaming data access, storing large data sets, simple coherence model. HDFS is generally classified into three parts called name node, data node and HDFS client/edge node

NAME NODE

It is centrally placed node, in which it consists of information about file system of Hadoop [6]. The primary job of name node is it records all the attributes and Meta data and correct locations of files and blocks in the data nodes [9]. Name node acts as the server node as it stores all the essential information about the system [14]. As name node acts as the master node it generally knows all information about allocated and replicated blocks in cluster. It also has information about the next free blocks which is to be allocated to compute next. The clients enquires to the name node for placing information in the file system and gives information which is recently added, removed pr modified from the data nodes [4].

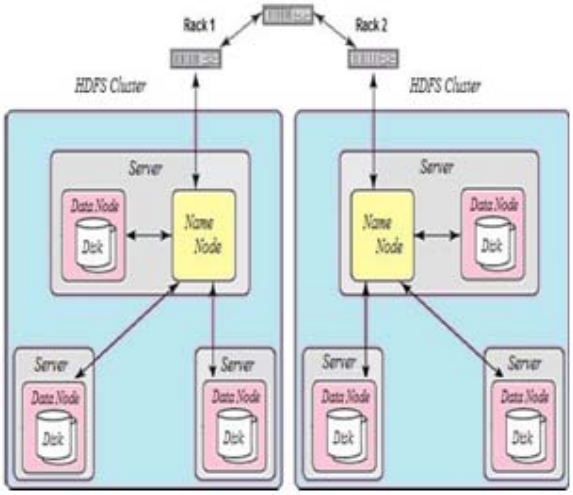

Figure-2. HDFS architecture.

DATA NODE

Another type of node in HDFS architecture is data node which is also called as compute node. It works as client node. Hadoop location may contain multiple data nodes based on capability and performance [6]. A compute node performs two main tasks i) storing adata block in HDFS and also acts as the environment for running jobs. During the early startup each computing node performs handshakes with name node. It checks for correct namespaces ID. If it found then it connects compute node to name node, and if it not found any correct namespace id then it simply close the connection [3] [9].

Each compute node keeps the current status of the blocks in its node and generates block report. After every hour data node sends that block report to server node so it always contains the updated information about the data node. During this process compute node also sends heartbeats to name node every 10 minutes, due to this the name node may identify which nodes are functioning properly and which are not functioning properly. If name node doesnot receive any heartbeat message from the data or computing node, then it just assumes that the compute node fails and it generated the replica of the data node in the same cluster [14].

EDGE NODE

Edge nodes are also known as HDFS clients [6]. It acts as the mediator between name node and the computing node. These edge nodes serves as an access points from which the user application uses the Hadoop environment efficiently [9]. In the cluster there are many clients depending on the performance [6]. When any user application needs to read a file, it contacts to the server node first and then receives the data nodes list that contains the required data. After getting the list, the client user access the data node which contains the file and also the replica location. Then the server locates the appropriate location for the file to be written.

MAPREDUCE

MapReduce is a programming model. It is used for computing large data sets with parallel and distributed algorithms in the cluster. MapReduce is the heart of Hadoop.

MATERIALS AND METHODS

FAULT TOLERANCE

Fault tolerance is defined as, when the system functions properly without any data loss even if some hardware components of the system has failed**.** It is very hard to reach cent percent fault tolerance but faults can be tolerated up to some extent. HDFS provide high throughput to access data application and suitable to have large data sets as their input [10]. The main purpose of this fault tolerance is to remove frequently taking place failures, which occurs commonly and disturbs the ordinary functioning of the system.

Single point failure nodes occur when a single node failure causes the entire system to crashes. The primary duty of fault tolerance is to remove such node which disturbs the entire normal functioning of the system [8]. Fault tolerance is one of the major advantages of using Hadoop. The three main solutions which are used to produce fault tolerance are data replication, heartbeat messages and checkpoint and recovery.

DATA REPLICATION

An application can specify the number of replicas of a file at the time it is created, and this number can be changed any time after that. [6] The name node makes all decisions concerning block replication. HDFS uses an intelligent replica placement model for reliability and performance. Optimizing replica placement makes HDFS unique from most other distributed file systems, and is facilitated by a rack-aware replica placement policy that uses network bandwidth efficiently.

The same copy of data is positioned on several different computing nodes so when that data copy is needed it is provided by any of the data node

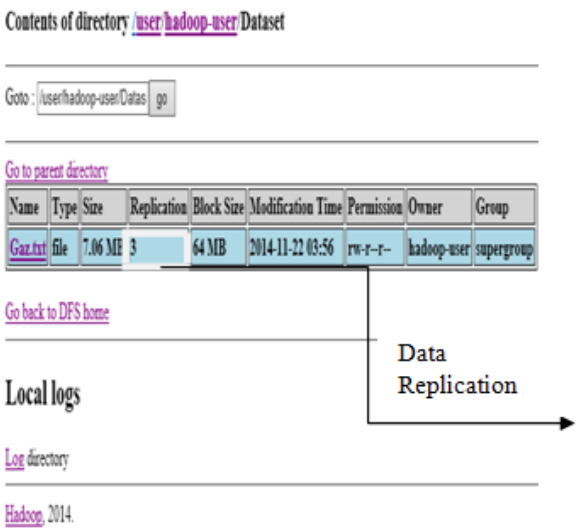

Figure-3. Data replication in name node location.

which is not busy in communicating with other nodes. The major advantage of using this technique is to provide instant recovery from node and data failures. But one main disadvantage is by using this type of fault tolerance mechanism high memory is consumed in storing same data on multiple nodes. There also may be possibilities of data inconsistency. It is frequently used method because this technique provides quick recovery of data from failures. Once the data is stored on the node in the cluster same copy of the data is replicated in two more nodes. i.e. totally three copies of data is located on the same cluster.

REPLICA SELECTION

If Read latency and Global bandwidth has to be reduced, HDFS tries to convince a read request from a replica that is closest to the reader. The replica is preferred to satisfy the read request, if there already exist a replica on the same rack on the reader node. Replica in the local data center is preferred over any remote replica when hadoop clusters spans multiple data centers.

Fair Scheduling algorithm is used for replica selection. A perfect replica node is selected if a node is failed to process or if the data has been loosed due to network link failure.

HERATBEAT MESSAGES

The solution for the above two problems are heartbeat messages. Here heartbeat message is a message sent from an inventor to the endpoint to identify if and when the inventor fails or is no longer available. Heartbeat messages are non-stop on a periodic recurring basis from the inventor's startup until the inventor's shutdown. When the receiver identifies lack of heartbeat messages during an anticipated arrival period, the destination may determine that the inventor has failed, shutdown, or is generally no longer available.

CHECKPOINT AND RECOVERY

In this method, similar concept as that of rollback is used to tolerate faults up to some point. After a fixed length of time interval the copy report has been saved and stored. It just rollbacks to the last save point when the failure occurs and then it start performing transaction again. Overall execution time of system is increased, because the rollback operations need to go back and check for the last saved consistent stages which increase the time. Also there is one major drawback of this method is that it is very time consuming method compared to first method but it requires less additional resources.

FUTURE SCOPE

In HDFS architecture system, failure in single node may breakdown the whole system. That is the Hadoop cluster is damaged when the server node i.e. the name node is down or failed. The obvious solution for this problem is to use more than one name node so that failure of one name node can be managed by another. Cost of managing different independent nodes in geographically distributed data center. So Cost minimization algorithm can be concentrated.

CONCLUSIONS

In spite of some of the failures and breakdown of name node, Hadoop provides better way of handling fault tolerance. We discussed about the basic architecture of Hadoop and its components and also about the fault tolerance mechanisms like data replication, heartbeat messages and checkpoint and recovery methods.

REFERENCES

[1] Apache Hadoop. http://hadoop.apache.org.

[2] Hadoop Distributed File System http://hadoop.apache.org/hdfs.

[3] Borthakur, D. 2007. The Hadoop Distributed File System: Architecture and Design. http://hadoop.apache.org/common/docs/r0.18.0/hdfs_design.pdf.

[4] Shvachko, K., *et al*. 2010. The Hadoop Distributed File System. *IEEE*. http://storageconference.org/2010/Papers/MSST/Shvachko.pdf.

[5] Apache Hadoop. http://en.wikipedia.org/wiki/Apache_Hadoop.

[6] Joey Joblonski. Introduction to Hadoop. A Dell technical white paper. http://i.dell.com/sites/content/business/solutions/whitepapers/en/Documents/hadoop-introduction.pdf

[7] An introduction to the hadoop distributed file Systemhttp://www.ibm.com/developerworks/library/wa-introhdfs.

[8] Selic, B. 2004. Fault tolerance techniques for distributed systems. IBM. http://www.ibm.com/developerworks/rational/library/114.htm.

[9] Jared Evans CSCI B534 Survey Paper. "Fault Tolerance in Hadoop for Work Migration" http://salsahpc.indiana.edu/b534projects/sites/default/files/public/0_Fault%20Tolerance%20in%20Hadoop%20for %20Work%20Migration_Evans, %20Jared%20Matthew.pdf.

[10] The Hadoop Distributed File System: Architecture and Design" by Dhruba Borthakur, http://hadoop.apache.org/docs/r0.18.0/hdfs_design.pdf .

[11] Hadoop - Advantages and Disadvantages http://www.j2eebrain.com/java-J2ee-hadoop-advantages-and-disadvantages.html.

[12] Shivaraman Janakiraman ''Fault Tolerance in Hadoop for Work Migration" http://salsahpc.indiana.edu/b534projects/sites/default/files/public/8_Fault%20Tolerance%20in%20Hadoop%20for%20 Work%20Migration_Janakiraman, % 20Shivaraman.pdf.

[13] Wang, F. *et al*. 2009. Hadoop High Availability through Metadata Replication. ACM.

[14] T. White. Hadoop: The Definitive Guide. O'Reilly, 2009.