HADOOP SKELETON & FAULT TOLERANCE IN HADOOP CLUSTERS

Vishal S Patil[1], Pravin D. Soni[2]
[1]ME (CSE) Scholar, Department of CSE,
P R Patil College of Engg. & Tech. AmravatiAmravati-444605,,

[2]Assitantant Professor, Department of CSE,
P R Patil College of Engg. & Tech. Amravati Amravati-444605,India,

ABSTRACT

In the today's era of information technology and computer science storing and processing a data is very important aspect. Nowadays even a terabytes and petabytes of data is not sufficient for storing large chunks of database. Hence companies today use concept called Hadoop in their application. Even sufficiently large amount of data warehouses are unable to satisfy needs of data storage. Hadoop is designed to store large amount of data sets reliably. Hadoop is popular open source software which supports parallel and distributed data processing. It is highly scalable compute platform. Hadoop enables users to store and process bulk amount which is not possible while using less scalable techniques. Along with reliability and scalability features Hadoop also provide faults tolerance mechanism by which system continues to function correctly even after some components fail's working properly. Faults tolerance is mainly achieved using data duplication and making copies of same data sets in two or more data nodes. In this paper we describe the framework of Hadoop along with how fault tolerance is achieved by means of data duplication.

Keywords: Hadoop, Fault tolerance, HDFS, Name node, Data node.

1. INTRODUCTION

Hadoop is an open source software framework created by *Doug cutting* and *Michael J. Cafarella* [1] [5]. Hadoop was initially inspired by paper publish by Google for their approach of handling data .Hadoop is name after Doug cutting sons toy elephant .It was originally designed to support distributed file processing system .Hadoop was written in java programming language .Hadoop can handle all type of data including audio files communication records e-mails ,multimedia ,picture ,log files etc.[5]. With using Hadoop there is no limit of storing and processing data. Hadoop uses computational technique named MapReduce, in which application is divided into many small fragment each of them is executed on various nodes in cluster .It provide distributed file processing system that store and process bulk of data .Hadoop uses HDFS for storage purpose [2]. HDFS is fault tolerance and provides throughput access to large data set .Hadoop is designed to efficiently process large volumes of information by connecting many computers that can work in parallel. Also one of the most important benefits of Hadoop is to limit the communication between the nodes and makes the system more reliable [12].

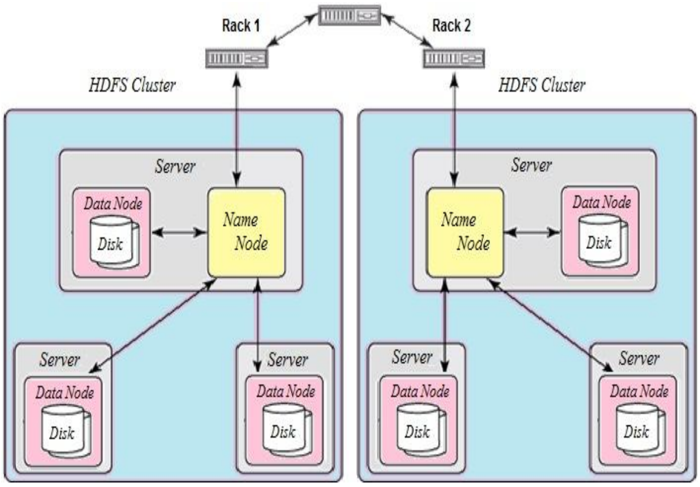
2. ARCHITECTURAL DESIGN OF HADOOP

Each Hadoop cluster contains variety of nodes hence HDFS architecture is broadly divided into following three nodes which are,

2.1 Name Node.
2.2 Data Node.
2.3 HDFS Clients/Edge Node.

2.1 Name Node

It is centrally placed node, which contains information about Hadoop file system [6]. The main task of name node is that it records all the metadata & attributes and specific locations of files & data blocks in the data nodes [9]. Name node acts as the master node as it stores all the 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 free blocks which are to be allocated next. The clients contacts to the name node for locating information within the file system and provides information which is newly added, modified and removed from data nodes[4].



2.2 Data Node

The second type of node in HDFS architecture is data node. It works as slave node. Hadoop environment may contain more than one data nodes based on capacity and performance [6]. A data node performs two main tasks storing a block in HDFS and acts as the platform for running jobs. During the initial startup each data node performs handshakes with name node. It checks for accurate namespaces ID if found then it connects data node to name node, and if not then it simply close the connection [3] [9].

Each data node keeps the current status of the blocks in its node and generates block report. After every hour data node sends the block report to name node hence it always has updated information about the data node. During this handshaking process data node also sends heartbeats to name node after every 10 minutes, due to this action the name node knows which nodes are functioning correctly and which not. If name node doesn’t receive heartbeats from data nodes it just assumes that data nodes are lost and it generates the replica of data node [14].

2.3 HDFS Clients/Edge node

HDFS Clients sometimes also know as Edge node [6]. It acts as linker between name node and data nodes. These are the access points which are used by user application to use Hadoop environment [9]. In the typical Hadoop cluster there is only one client but there are also many depending upon performance needs [6]. When any application wants to read a file it first contacts to the name node and then receive list of data nodes which contains the required data , hence after getting that list the clients access the appropriate data node requesting the data node which can hold that file also location of replica’s which is to be written. Then the name node allocates the appropriate location for that file.

3. FAULT TOLERANCE IN HADOOP

When the system continues to functions correctly without any data loss even if some components of system have failed to perform correctly. It is very difficult to achieve 100% tolerance but faults can be tolerated up to some extent. HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware. HDFS provides high throughput access to application data and is suitable for applications that have large data sets [10]. The main purpose of system is to remove common failures, which occurs frequently and stops the normal functioning of system. When a single node causes whole system to crash and fails such node are known as single point failure nodes. In faults tolerance system its primary duty is to remove such nodes which causes malfunctions in the system [8]. Fault tolerance is one of the most important advantages of using Hadoop. There are mainly two main methods which are used to produce fault tolerance in Hadoop namely Data duplication and Checkpoint & recovery.

3.1 Data Duplication

In this method, the same copy of data is placed on several different data nodes so when that data copy is required it is provided by any of the data node which is not busy in communicating with other nodes. One major advantage of this technique is that it provides instant recovery from failures. But to achieve such type of tolerance there is very large amount of memory is consumed in storing data on different nodes i.e. wastage of large amount of memory & resources. As data is duplicated across various nodes there may be possibility of data inconsistency. But as this technique provide instant and quick recovery from failures hence it is frequently used method compared to checkpoint and recovery.

3.2 Checkpoint & Recovery (Rollback)

In the second method, similar concept as that of rollback is used to tolerate faults upto some extent. After a fixed span of time interval the copy report has been saved and stored. If the failure occurs then it just rollback upto the last save point and from there it start performing the transaction again. This method uses concept called rollback that is a rollback operation brings the system to its previous working condition. But this method increases overall execution time of system, 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.

4. ADVANTAGES OF USING HADOOP

Following are the various advantages of using Hadoop and HDFS.

- The main advantage of Hadoop is that it is highly reliable.
- It is simple and robust.
- It is used to store large data set without any limit on storage.
- It is highly scalable storage platform.
- Cost effective as it is 100% open source software.
- Quick recovery from system failures.
- Ability to rapidly process large amounts of data in parallel.
- Once data written in HDFS can be read several times.

Provides Faults tolerance by detecting faults and provide mechanism for overcoming them [11].

5. FUTURE SCOPE

As in the HDFS architecture there is only single node there may be possibility of breakdown of whole system. That is Hadoop cluster is unavailable when Name node is down. Hence the obvious solution for this problem is that using more than one name node so that even one fails another can handle the processing of data. In addition to that HDFS know to have scheduling delays that keeps the system away from reaching to its full potential. Also one major drawback is that cost of managing multiple independent namespaces when there are large namespaces. Another main point of discussion is that Hadoop has been known to affect users is the lack of good high level support for Hadoop. Without proper support we cannot increase the functional ability of Hadoop [13].

6. CONCLUSION

Despite some of the shortcomings like failures and breakdown of name node, Hadoop with HDFS provides a quite good enough way of handling faults tolerance. In this paper we discussed about the architectural framework of Hadoop and also some of the strategies to overcome the faults tolerance in the HDFS that includes data duplication and checkpoint and recovery. This research can be extended by providing mechanism for handling breakdown in name node. Also arrange some alternative and backup recovery for name node failure.

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%20Work%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.

AUTHOR

Mr. Vishal S. Patil, Received Bachelor’s Degree In Computer Science Engineering from SGB Amravati University & Pursuing Master Degree In CSE from PR Patil College of Engineering. Amravati-444602 Maharashtra, India

Mr. Pravin D. Soni, Received the Master Degree in Computer Science from VJTI, Mumbai in 2011.Working as a Assistant Professor In Department of Computer Science and Engineering at PR Patil College of Engineering. Amravati-444602
Maharashtra, India