

ANALYSIS AND DESIGN OF TECHNIQUE TO INCREASE THE AVAILABILITY IN HADOOP ARCHITECTURE

Seema Rawat¹, Shankar Ramamoorthy²

Abstract—In today's scenario the main problem is data analysis, Apache Hadoop is one of the solutions of Big Data. Hadoop below version 2 lack high availability in NameNode. In the world of cloud computing, huge sets of data are stored, gathered and integrated from everything we are surrounded by all the time. To extract important esteem and pertinent required information from enormous data, we require ideal analysis capabilities and skills. Hadoop distribution file system has a master and slave architecture. It comprises of a master name node i.e. the single server that deals with the record framework and controls access to documents by client's. The name node keeps a reference to every block and file in the file system in memory. I have found the solution of high availability in hadoop version 1. To do this, I used centralized storage that is NFS server and shared that storage with two places. Let's call them one primary and one secondary NameNode according to vocabulary of hadoop version 2 and above. Primary NameNode is working as a master to the DataNodes while secondary NameNode will keep track of live status of primary NameNode. When primaryNameNode goes down then secondary NameNode will replace it's ip with primary NameNode ip and then it becomes the master to the DataNodes. In this way we are trying to obtain high availability in hadoop version less than 2.

Keywords— Big Data; Big Data Tools; Apache Hadoop; HDFS; Hadoop v1 Limitation; High availability in Hadoop v1.

1. INTRODUCTION

Big Data refers to very large structured or semi structured and unstructured data. With traditional data processing application software it is difficult to deal with them. Storing data, searching, analyzing data are included in challenges of big data. Big data characteristics can be described in five V's that are volume, velocity, variety, variability and veracity. Volume means the amount of data generated and stored. Velocity means the rate at which data is generated and analyzed to meet the demand of growth and development. The variety is the type of data generated that can be structured, semi structured and unstructured data. Variability means data inconsistency which may hamper processes to handle and manage it. Veracity means the data vary frequently affects the accurate analysis [1].

Hurdle with big data is when it comes to handle different types of data; we also refer them as heterogeneous data. Heterogeneous data are composed of structured data like data of RDBMS, Data warehouse, etc. Unstructured data like analog data, gps tracking information, audio, video, images, etc. Semi-structured data like xml, email, json, etc. Analysis of structured data is easier than other forms of data because our processing of data is most efficiently if they can store multiple items that are all identical structure and size. Other problems related to big data scaling, timeliness and privacy. Day by day the data is increasing enormously so storage of big data and analysis of big data require large storage as well as fast processing. Scaling can be done in two ways horizontal scaling and vertical scaling. Earlier horizontal scaling was approach to process big data. They tend to increase processing power each time the data is increased. Today vertical scaling approach is best practice to handle big data. Privacy is also major concern, public fear of inapt use of their personal data [2].

Big data is a buzzword among companies and their business nowadays. Big data analysis has revolutionized many businesses. This analysis report helps the executives of the company to take right and fast decisions. Different types of analytics solutions can be classified as descriptive, predictive and diagnostic. Descriptive analytics means to yield the useful information from events and reports from history to identify patterns and create management reports. We can also say modeling past behavior. In very simple way just say what happened. Predictive analytics identifying from past pattern and predict the future. In simple way we can say what could happen in future. Diagnostic analysis means what happened and why it happened. By following above strategies companies take their decisions [3].

2. BIG DATA TOOLS

The Apache Cassandra database is one of the tools which provide effective management of large data. It's property of providing scalability and high availability without compromising performance. Since it works in commodity hardware it is cheaper and efficient. Providing fault-tolerance in commodity hardware or cloud infrastructure make it flawless technology

¹ Research Scholar, Swami Vivekanand University, Sagar, (M.P) India

² Professor, Swami Vivekanand University Sagar, (M.P) , India

[4]. Neo4j is one of tools which provide big data solutions. Neo4j is scalable graph database technology which is developed using java technology. Neo4j is Flexible data model, provides real time insight and incorporate feature scalability and high availability which makes it more reliable [5]. Splunk is one the best analytics tools for analyzing log files. This software for monitoring, searching and analyzing machine generated big data. MongoDB is also one of the tools which is open-source collection and documents-oriented database program for storing as well as analyzing big data .It comes under NoSql Data base technology. MongoDB uses java script documents with schemas

MongoDB is a product developed by MongoDB Inc [6]. One of the commonly used tools for analyzing big data is Tableau. Tableau is an interactive data visualization tool developed by software company Tableau software focused on business intelligence. There are many other tools and technology which are working on the analysis as well as storage of Big data.

Hadoop Architecture- In Hadoop architecture, HDFS is used for storage/reads-writes and MapReduce or YARN is used for processing.

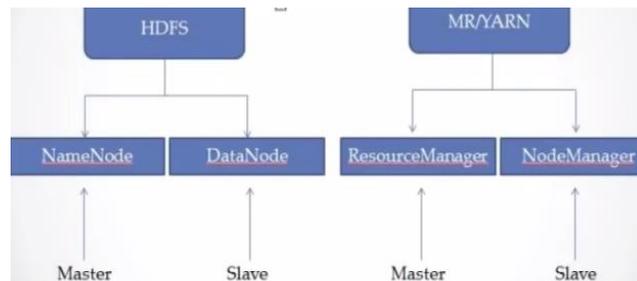


Fig.1 Hadoop Architecture

Storage part is handled by NameNode and DataNode via HDFS. Where NameNode is master and DataNode is slave. Similarly, MP/YARN is handled by ResourceManager and NodeManager. Where ResourceManager is master and NodeManager is slave. The system runs on master system that are known as master daemons and the system runs on slave system that are known as slave daemons. DataNode and ResourceManager will run on master daemon and DataNode and NodeManager will run on slave daemons. Secondary NameNode is run on master daemons. This will only take hourly backup and stores it. It will never take place in case NameNode is crashed. But, it can restart the crashed Hadoop cluster. It is very important daemon for Hadoop1 however, In Hadoop 2 it is not much important.

2.1 HADOOP Framework

The Apache Hadoop is a open-source software for, scalable, reliable, distributed storage and distributed computing. The Apache Hadoop software is a framework that works on the formula of distributed processing of big data and distributed storage of data across different nodes of clusters. It is designed to horizontal scale up. Apache hadoop can scale from single servers to thousands of machines, each performing local computation and storage. It one of the prominent properties is high availability which means the resources are always available. For high availability does not depends on hardware it designed in such a way the high availability is obtained by hadoop software itself. The hadoop software includes these modules [7]:

- Hadoop Common: The utilities that support the other Hadoop modules.
- Hadoop Distributed File System (HDFS): A distributed storage or file system that provides fast access to application data.
- Hadoop YARN: It’s a model that schedules jobs and manages cluster resource.
- Hadoop MapReduce: Hadoop mapReduce provides parallel processing. It works in two stage mapper and reducer. Because of parallel processing of data makes the hadoop one the best solution of bigdata.

2.2 Basic comparison between Hadoop 1 and Hadoop 2:

There are two components of hadoop i.e. hadoop 1.0 components and other is hadoop 2.0 components.

Table 1:. Comparative study of Haddop 1.0 and Hadoop 2.0 components.

SNo	Parameter	HADOOP 1	HADDOP 2
.			

1.	Components	HDFS (hadoop distributed file system) which is used for storage and MapReduce which is used for programming framework.	HDFS (hadoop distributed file system) which is also used for storage and YARN/MRv2(yet another resource negotiator/MapReduce version 2) which is used for cluster management.
2.	Daemons	NameNode, DataNode, Secondary NameNode, Job Tracker, Task Traker.	NameNode, DataNode, Secondary NameNode, Resource Manager, Node Manager.
3.	Working	In this, all storage is done onto HDFS and resource management and data processing is done under MapReduce.	In this, All storage is done onto HDFS but here resource management is done by YARN and MapReduce and other types of jobs are processed by its upper level.
4.	Single point of failure	In this, there is single master and multiple slaves are present so, if in case master crashed then whole system will not work.	Whereas, in this there is one master which is active an another is on standby so in case active Namenode crashed then standby master will take place.

MapReduce is a kind of model which is based on distributed computing and also a processing technique. This algorithm consists two important tasks, Map and Reduce. One can take its advantage in unravel surprising trends in real world data. It is programmed in core java to write business logic for better processing.

In MapReduce, the input and output both are key value pair. Mapper first need to get data i.e input file by default it gets data in text form which can be configurable. For processing data one has to mention business logic in it and finally it writes the output. Mapper's output is further forward to Reducer. Any file defined in Hadoop is divided in blocks.

$$\text{No. of Mapper} = \text{No. of Blocks}$$

After coding and removing the jar file and put that file on execution at the same time processing will start wherever the data is and mapper start running. The number of mapper runs is the same number of blocks is available.

Reducers get data from Mapper and one has to write business logic after its process completion it take out the final summarized output. It's algorithm has two task to be done i.e. Summarization operation and by default one reducer. Once the data received by reducer then its work is to summarize the data and by default one reducer means all the mappers will run but there will be only one reducers which is fully configurable on the basis of number of output required by programmer.

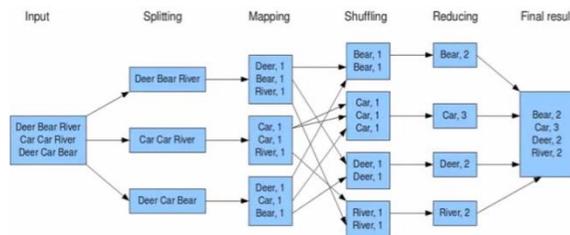


Fig. 2: Working criteria/example of MapReduce

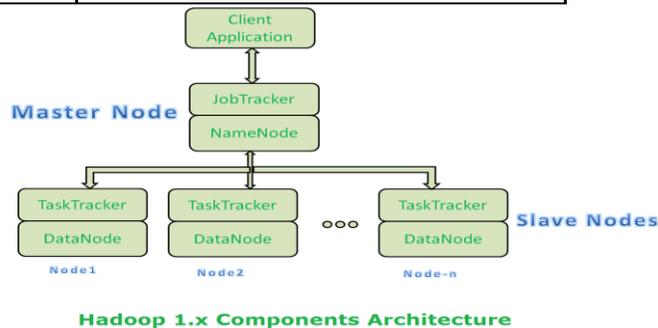
Here, Mapper input is unorganized which is further divided into different blocks after execution of map function it gave the output. Then shuffling process starts it merge similar input afterwards reducer summarized it and provide the final output. After looking onto the major benefit comparative with traditional way so, traditionally big data was split into various machines and each system need to code accordingly and then processing starts. Afterwards in the end all the results again need to aggregated for final result. And if any system is crashed then it forms the data loss. But in this, it make most of its work by itself such as dividing the file, making replication, converting it into file blocks. Secondly, program is written at only once which makes process faster. If any system is crashed so, data can be achieved from another system. Here, competition is taken closer to data which make summarization faster and easy and it is done using clustering technique.

2.3 HDFS

The Hadoop Distributed File System is a distributed file system architect to run on commodity hardware. Commodity hardware means the hardware that is relatively inexpensive and is widely available. The difference between commodity hardware and other distributed file systems are high fault-tolerant and low-cost of hardware devices in HDFS. HDFS provides high throughput access to data and is also suitable for Big Data. HDFS also provide high availability means the data is always available to client. If one of the servers goes down there is replica of same data on different server HDFS automatically redirect the request to the other live server [8]HDFS is stores the application data and its Meta data in two different server. A server in which Application data is stored is called DataNodes. A server in which Meta data is stored is called NameNode. HDFS is works on Master-Slave model. NameNode is master and DataNodes are slave. NameNode is the one which drives all the processing of data because it contains Meta data of all the data. Apache Hadoop is available in from version 0.14.1 to version 3.0.0-alpha4. Mainly Hadoop v1 and Hadoop v2 are used. HDFS architecture of both versions is almost similar but MapReduce has different architecture. Difference between Hadoop version 1.0 and Hadoop version 2 [9].

Table II: Difference between hadoop version 1 and hadoop version 2.

Hadoop version 1.x.x	Hadoop version 2.x.x
Support Map Reduce for Data processing but does not support other computing models.	Support Map Reduce as well as other computing models such as Sparks, Giraph, Pig, Hive, etc.
Map Reduce(MR) is responsible for data processing and cluster management	Yet Another Resource Negotiator (YARN) is responsible for data processing and cluster management
Scalability is limited to 4000 nodes per cluster	Scalability is upto 10000 nodes per cluster
Single NameNode manages the entire mapper and reducer tasks	Multiple NameNode manages the entire mapper and reducer tasks.
Microsoft windows is not supported	Microsoft windows is supported



2.4 Limitation Of Hadoop Version 1.X.X

Hadoop below version 2.0.0 lack the feature of high availability of namenode. If NameNode fails entire cluster fails in hadoop version 1.x.x but in hadoop version 2.0.0 provide high availability in NameNode. I have found the solution to this problem by providing high availability in hadoop version 1.

2.5 High Availabilty Solution In Hadoop

The simple definition of high availability is always available. Since data and applications are very critical on the internet so it very important that services needs to be highly available. High available systems typically attempt to achieve 99.999% (5-nines) availability [10]. High availability can be achieved in cluster system where in each nodes are independent. High availability is engineered in such a way when one node goes down other node goes up. In amazon web services high availability is provided in an entire region. In a region we have more than two data centers. Even if there is disaster in one data center, client’s data will not be lost.

When I say high availability in hadoop version 1.0 I mean high availability of NameNode. Hadoop version version1.0 already has high availability of DataNode but I am going implement high availability of NameNode in hadoop version 1 to do this we need have following [10].

2.5.1 Centralized storage

To provide centralized storage we can use different storage techniques, I used Network File System (NFS) server for that. NFS server is distributed file system protocol. In this server we can share storage. Suppose if storage [12] is shared with read and write permission to two clients then any change in server or any of two clients is going to reflect everywhere because the storage is shared. That is why when data is stored at primary NameNode it is stored in NFS server and same changes are reflected also in secondary NameNode.

2.5.2, Two NameNode

The storage is mounted in both the NameNodes. Both NameNodes must have same configurations in the configuration files that are `core-site.xml` and `hdfs-site.xml`. Let us call one NameNode as primary and another NameNode as secondary NameNode according to vocabulary [13] of Hadoop version (Note: - The secondary NameNode here is taken as an example to explain. The secondary NameNode in Hadoop version 2 and above not only provide high availability but also may have other features). Here Secondary NameNode has two jobs. First it keeps track of the live status of the primary NameNode and second it will change its IP with the IP of the primary NameNode when the primary NameNode fails.

2.5.3 Implementation

The prerequisite to run this Java code is primary NameNode and secondary NameNode will have the same configuration settings, i.e., configuration of `hdfs.site.xml` and `core.site.xml`. This Java code will run at the secondary NameNode. It checks if the primary NameNode is alive or dead by pinging it in an infinite loop. When the secondary NameNode pings successfully, the output of the `getStatusOutput()` function will be 0. If there is any unsuccessful ping, the output of the `getStatusOutput()` function of Java will be non-zero and the loop will break. It will then change the network settings of the secondary NameNode by using Java file handling. In this network, the secondary NameNode changes its IP address with the IP address of the primary NameNode. Then it restarts the network settings. After this, we will mount the NFS shared file to the same location as mounted in the primary NameNode. We need to restart the NameNode daemon. In case the NameNode goes to safe mode, we need to leave the safe mode because in safe mode the NameNode does not work.

Importing commands

```
#checking heartbeat of primary namenode while True:
```

```
check=commands.getStatusOutput("ping -c 1 primary_namenode") if check[0]==0:
```

```
print check[1] pass
```

```
else: break
```

```
#change virtual ip with ip of primary namenode in centos 7.3
```

```
file1=open("/etc/sysconfig/network-scripts/ifcfg-enp0s3:1",'r+') file1.seek(323)
```

```
file1.write("primary_namenode") file1.close()
```

```
# restart network
```

```
commands.getStatusOutput ("systemctl restart network") commands.getStatusOutput ("ifup enp0s3")
```

```
#mount namenode
```

```
commands.getStatusOutput("mount 192.168.43.147:/nfs /namenode")
```

```
#restart hadoop-daemon
```

```
commands.getStatusOutput("hadoop-daemon.sh start namenode") commands.getStatusOutput("hadoop dfsadmin -safemode leave")
```

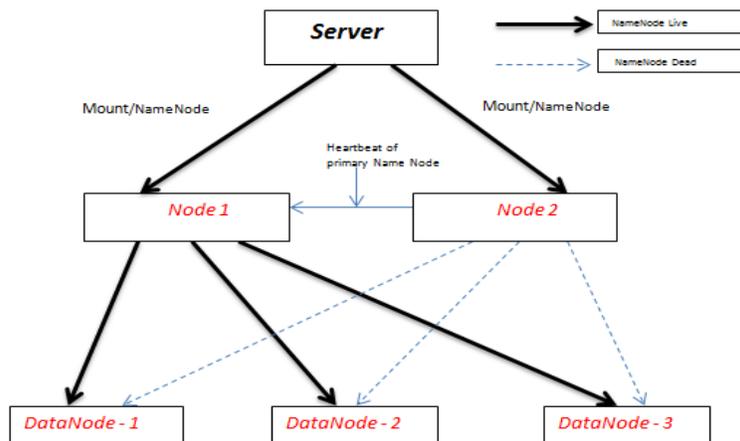


Fig 4. Architecture of implementing high availability of namenode in hadoop version 1.0.0

3. CONCLUSION

We are living in era where data storage and analysis is back bone of every company and data is a crucial entity for company as well as client. Client always want that data should always be available that means they ask for zero down time. High availability feature has become one key aspect in every technology. Many technologies are used for high available solution like Raid redundant array of independent disk, SAN storage area network and now microservices like kubernetes and docker swarm provides high availability solution. We have developed a code which allows us to avail high availability in apache hadoop. Now if any company wants to use hadoop version less than 2.0 then they can use this technique. They don't require migrating from hadoop version 1.0 to Hadoop version 2.0. With this technique we can obtain high availability not only in hadoop but also any other technology with some little configuration and other related processing hardware.

4. References

- [1] Suhail Sami Owais, Nada Sael Hussein, "Extract Five Categories CPIVW from the 9V's Characteristics of the Big Data," (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 7, No. 3, 2016, pp. 254-258.
- [2] Harshawardhan S. Bhosale¹, Prof. Devendra P. Gadekar², "A Review Paper on Big Data and Hadoop" International Journal of Scientific and Research Publications, Volume 4, Issue 10, October 2014 1 ISSN 2250-3153, pp. 1-7.
- [3] Praseeda C K, Dr.B L Shiva kumar "A Review of Trends and Technologis in Business Analytics" International Journal of Advanced Research in Computer Science, Volume 5, No. 8, Nov-Dec 2014, ISSN No. 0976-5697, pp. 225-229.
- [4] Cassandra, <http://cassandra.apache.org/>
- [5] Neo4j, <https://neo4j.com/>
- [6] MongoDB, <https://www.mongodb.com/>
- [7] Apache Hadoop, <http://hadoop.apache.org/>
- [8] M. Dhavapriya, N. Yasodha, "Big Data Analytics: Challenges and Solutions Using Hadoop, Map Reduce and Big Table" International Journal of Computer Science Trends and Technology (IJCTST) – Volume 4 Issue 1, Jan - Feb 2016, ISSN: 2347-8578 pp 5-14
- [9] Sneha D.Borkar, Prof.Chaitali S.Surtakar, "A Review Paper on The Hadoop Distributed File System" International Journal of Research In Science & Engineer Volume:1 Special Issue: 1, e-ISSN: 23948299 Volume:1 Special Issue: 1 p-ISSN:23394-8280, pp. 221-226.
- [10] High availability software, https://en.wikipedia.org/wiki/High_availability_software.
- [11] Saxena, Piyush, Satyajit Padhy "Use of storage as a service for online operating system in Cloud Computing." International Conference on Telecom and Networks (TELNET)(2013).
- [12] Saxena, Piyush, Satyajit Padhy "Eliminating homogeneous cluster setup for efficient parallel data processing." International Journal of Computer Applications 64, no. 17 (2013).