Storage management in Big Data Using Hadoop

[1]Shivanand B Lamani, [2]Ravikumar K, [3]Arshi Jamal
[1]Faculty in Computer Science Dept., [2,3]Asst Professor in Computer Science Dept.,
[1]Akkamahadevi Women's University, Vijayapur,
[2]GFGC, Gangavati, India,
[3]GFGC, Sindhanur, India.

Abstract: *Nowadays the data size is increasing because of the use by both public and private sectors and the main reason for vast quantity of increase in data is moving to electronic world. The size of the data is increasing every year by around double of last year. There are mainly three types of data structured, unstructured and semi-structured data. The existing systems were unable to store the structured, unstructured and semi-structured data in a single database, Big data is used to store such data. As day by need data size in increasing we need storage management in Big data.*

In this Paper we are managing the storage of Big data. Apache Hadoop is a platform used for storing data, processing and transforming Big data. Hadoop platform is an open source and freely available software at "Apache Software Foundation".
The main technique of the Paper is management, storage of the data in master machine and divides the data and store in n number of virtual machines and retrieves the data from virtual machines. Replication factor and other important factors need to be included.

Keywords: **Big data, storage management, Hadoop**

I. INTRODUCTION

Every day nearly 2.5 quintillion bytes of information is made and information creation will be expanding. There are many sources from which data is collected such as information from social networks, every day transaction records, cell and GPS information etc.Big data refers to a combination of data with varying size and increment in the data makes it non manageable to store and manage. Big data is combination of structured, unstructured and semi-structured data. Whose size ranges from terabytes to petabytes. Big data uses Hadoop platform for data storing, processing and managing. It will manage the large quantity of data.

Around 2.5 quintillion bytes of data is created by users in a single day. Every year information creation is increasing in large quantity. The purpose behind information creation is there are many sources of information such as sensors, social networking, cell and GPS information. This part of information is called Big Data. Big Data is a technology to store, capture, distribute manage and analyze large size data of different types at high speed. Data comes from various sources in different formats and arrive at various rates. To process such data in an efficient way parallelism is used.

Hadoop is the commonly used software by Big Data. Hadoop is software that uses the technique of distribution of data across clusters of slave machines. It can range from a single master machine to thousands of machines with fault tolerance. Data may be in any form such as unstructured, semi-structured, structured, and heterogeneous. Hadoop and HDFS are used for storage and management of Big Data. Analysis of Big data accepts large distributed file system and data should be flexible, scalable and fault tolerant. Analysis of the data is performed by Map Reduce. Map Reduce uses different techniques for mapping, reducing, searching, shuffling and sorting from DBMS.

1.1 Advantages of using Hadoop

- **Ability to store and process huge amounts of any kind of data, quickly.** It will store data from many platforms such as social media (facebook, gmail etc), internet. Processing of the data is done easy by distributed storage of data.
- **Computing power.** Hadoop processes data fast because of distributed storage of data. Processing power will increase with increase in number of computing nodes.

- **Fault tolerance.** Data is distributed to number of nodes and if one of the node becomes dead then we can retrieve data from the other nodes, where data is replicated to other nodes using replication factor. Number of copies of data is generated and stored automatically by HDFS.
- **Flexibility.** There is no need to preprocess the data for storing. We can store any kind of data such as text, images, videos, audios etc. Data may be structured, unstructured or combination of both.
- **Cost.** This software is freely available
- **Scalability.** It scales linearly by adding extra nodes. Not much maintenance is needed.
- **Robust:** It is less prone to failure of data.
- **Simple:** It is simple to handle because of the simple programming model.

1.2 Characteristics of Big data

- **Volume:** It indicates the amount of data generated and stored. The capacity of Big data ranges from terabytes to perabytes.
- **Velocity:** It indicates the rate at which data is generated, stored and processed.
- **Variety:** Big data collects a data from various resources such as social media, sensors, web, user interactions and review sites. The data collected is in variety of forms. Data can be structured, unstructured or semi-structured.
- **Complexity:** Data management will become a complex process when large volume of data comes from many sources. Such data needs to be linked and correlated to grasp the information.
- **Variability:** Inconsistency of the data set can hamper processes to handle and manage it.
- **Veracity:** The quality of the data being captured can vary greatly. Accuracy of analysis depends on this.

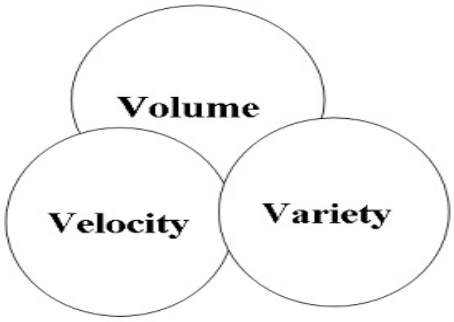

Fig 1. Three V's of Big data

1.3 Classification of Big Data

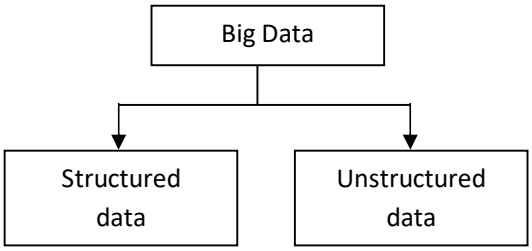

Fig.2 Types of data

Structured Data: These are the numbers and words that can be easily categorized and analysed. It consists of the data generated by network sensors, Smartphone's, GPs.

Unstructured data: It includes more complex information such as data generated from commercial websites. These data cannot be separated and analyzed.

Semi structured data: As the explosive growth of the internet in recent years the amount of data and types of data grow. Most of the data is unstructured data.

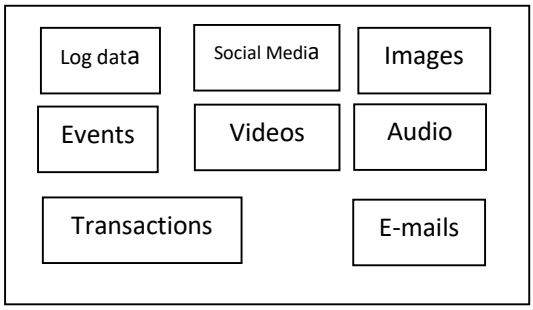

Fig.3 Sources of Big data

1.4 Problem Statement

The objective of the proposed system is to design Big data storage and management system. The data is stored in the masters and slaves are connected to master. Master mode will divide the data and store in a number of slave nodes. Master node will provide access to the data stored in it to the slave nodes.

II. PROPOSED METHODOLOGY AND IMPLEMENTATION

2.1 Proposed System

In centralized environment there is only one master machine that holds all data means centralized storage. If centralized storage fails then whole system fails. But in Hadoop environment the storage done in distributed way. There is only metadata that hold whole information about the data which is stored on different location. Here metadata called Namenode. The name node backup will be taken Secondary Namenode after some interval of time it will store to SSN. Here once we want to store data it will replicated in 3, means it will store in different files in different machine to secure data.

In centralized environment there is only one master machine that holds all data means centralized storage. If centralized storage fails then whole system fails. But in Hadoop environment the storage done in distributed way. There is only metadata that hold whole information about the data which is stored on different location. Here metadata called Name node. The name node backup will be taken Secondary Namenode after some interval of time it will store to SSN. Here once we want to store data it will replicated in 3, means it will store in different files in different machine to secure data.

In centralized environment there is only one master node and multiple slave nodes. The master node holds all data. Any slave machine has storage device. All are connected to the master node to store and retrieve data. Here the storage has a limit if it go in large size like TB, PB, EB. The computing time will be huge. To overcome this problem hadoop system introduced. In hadoop environment one master node and multiple slave nodes present. Here master node contains the Name Node (NN), Job Tracker (JT). Every node contains two part storage part and computation part. The NN is the metadata it hold the address of whole slave node where the actual data is stored.

The JT is computation part here it will take the data from NN means address. And compute using task tracker. The slave node contains Task Tracker (TT) and Data Node (DN). The DD contains actual data or physical data. The TT contains computation on actual data. Here the storage done in different data node. The data is replicated to different slave nodes. So data loss problem will be solved. Any data loss or slave node crash other will be served. There is chance of master node loss. It is solved by Secondary Name Node (SNN). Every time interval the name node value will be copied into SSN so it will be secure.

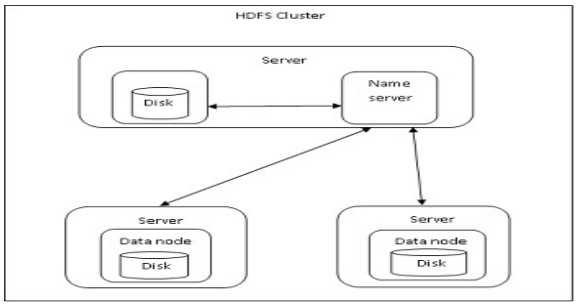

Fig. 4 HDFS Cluster

On each slave machine data node is there. It takes job from master and takes care of it. JobTracker will take of all process of master. JobTracker assigns jobs to TaskTracker in each slave machine using MapReduce Programming model. Processing of the data in each slave machine is handled by TaskTracker.

Now let us explain how data can be stored and processed by hadoop. Suppose we want use Hadoop to store 192MB of data adding a replication factor of 2. Generally the replication factor is more then 3. In Hadoop platform before processing the data, file will be broken down into a size of 64MB or 128MB and that data blocks are moved to different slave nodes. Processing of the blocks will be done by running Hadoop framework. JobTracker will schedule the jobs of each node and TaskTracker processes the data. After completion of storage and processing of data the output is written back.

192 MB of data will be divided into three blocks of size 64 MB (64MB * 3=192MB). In this example we want replicate the data by the factor of 3, to minimize the loss of data. In our example three slaves are available; Hadoop will replicate the block on different slave machines mainly to recover from dataloss by node failure. If one of the node fails we can recover from another machine.
Fig describes the following things,
Block A → stored at slave 1 and 2
Block B → stored at slave 1 and 3
Block C → stored at slave 2 and 3
The storing, managing, analysis and processing will be performed by Hadoop itself and client will simply provide the data to the master node and replication factor.

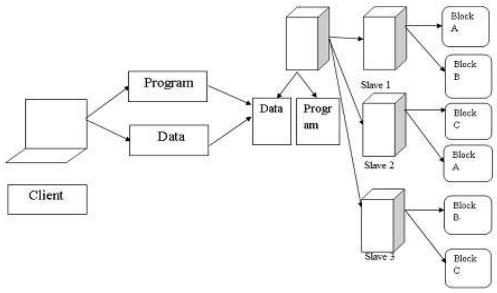

Fig. 5 Storage and processing of data

III. RELATED TECHNOLOGIES

3.1 Hadoop

It is free available software and open source platform. It runs on “Linux environment”. Large quantity of data can be stored and processed. The main advantage of Hadoop is its ability to span around thousands of servers that will not share “memory or disk space”.

For example, if we want to store the whole organizations data into HADOOP, then software will divide the data into pieces and then spread across the different servers. Hadoop keeps track of the locations of all servers where data is stored.Apache Hadoop consists of many components like “Hadoop kernel, MapReduce, HDFS, Apache HIVE, HBase, Zookeep”.

HDFS (“Hadoop Distributed File System “):

It is a “fault tolerant storage system”. It will store vast amount of data, scale up and also recover from loss of data. It creates a group of machines called as cluster and distributes work among the machines. If one of the machines fails, still it will operate without losing any data.it will divide the data into pieces called as “blocks” and stores it across the number of servers. It is having a default replication factor of 3.

How Does Hadoop Work?

It is generally “expensive to build bigger servers” for handling large scale processing, but we can form a cluster by connecting many machines to a master machine.

Main tasks of Hadoop are:

- Data is divided into pieces of uniform size 64 or 128 MB.
- Those pieces forms a blocks and are distributed across cluster.
- HDFS performs processing of the files.
- Hadoop has inbuilt feature of replications and it will replicates to recover from hardware failure.

3.2 HDFS Architecture

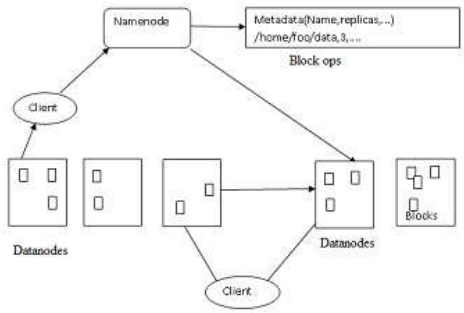

Fig 6. HDFS Architecture

HDFS architecture has a master and slave format. It consists of a single NameNode and a number of DataNodes. NameNode acts as a master and distribute the tasks to DataNodes into fixed blocks of 64MB or 128 MB size. There will be only one DataNode per cluster and it manages the storage. It executes operations like “opening, closing, and renaming”. DataNode will perform read and write operations. DFS follows the master-slave architecture and it has the following elements. It manages system namespace. NameNode will perform the task provided by DataNode.

Goals of HDFS

- **Fault detection and recovery:** It includes commodity hardware and failure of hardware component occurs repeatedly. It includes a mechanism for fault detection and recovery.
- **Large amount of data:** It includes a large number of nodes to store vast amount of data.

Benefits of Hadoop

- It performs two tasks such as storage and distributed processing.
- Cost is less compare to other storage options.
- Hadoop gives protection against hardware failure.
- It can be designed easily and can scale up automatically.

- No processing of data is needed.

3.3 MAPREDUCE

MapReduce is a program, which will divide a single large job into a number of small tasks and distribute to them using Mapping and Reducing functions. Mapping and Reducing together will form a MapReduce system. It will support mainly Java and can also support other languages. The tasks distributed to a number of nodes will run together using Mapping and results are combined together using Reducing.

- **JobTracker:** It keeps track of which node stores which data.
- **TaskTracker:** JobTracker assigns jobs to TaskTracker and inform later after the completion of task.

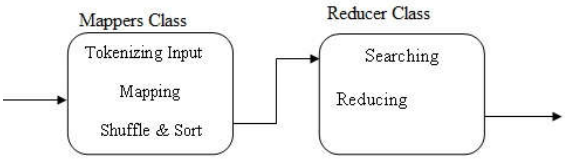

Fig 7. Map Reduce

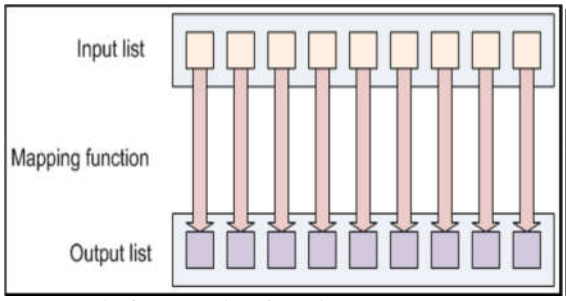

Fig 8. Mapping function

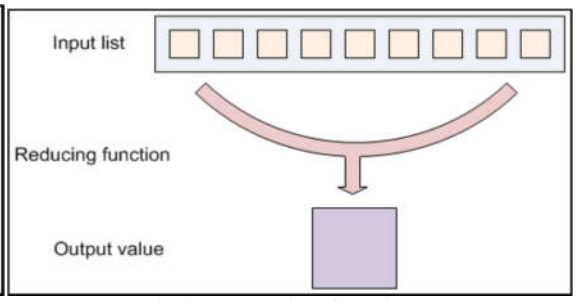

Fig 9. Reducing function

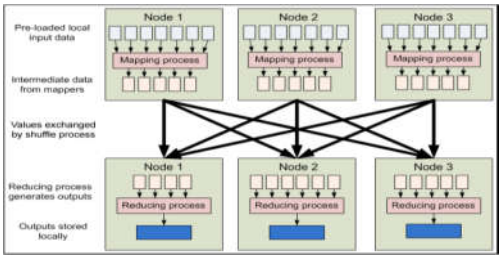

Fig 10. Mapping and Reducing

Here we will understand the working of MapReduce

- **Input**: it will read the data and translate to <key, value> pairs.
- **Map:** It will take <key, value> pairs and processes Mapper.
- **Combiner:** It groups similar data.
- **Shuffle and Sort:** it will take the pairs from Reducers and sort depending on large key.
- **Reducer:** It reduces data by aggregation, combination and filtering.
- **Output:** It translates final <key,value> pairs from reducer and writes to blocks.

IV. RESULTS AND DISCUSSION

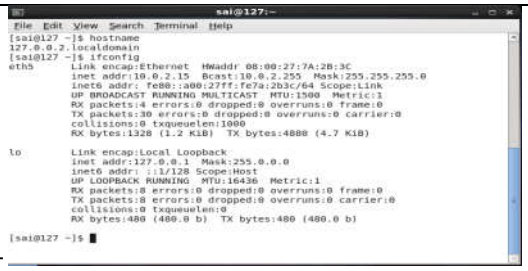

Fig 11. Checking IP address of the system

Changing the Hostname to the IP address so it can access outside machine.

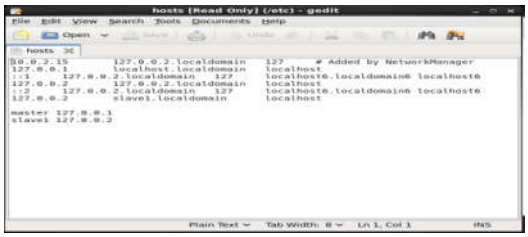

Fig 13. Setting all master node and slave nodes

Here we are setting the task tracker and job tracker for computation of data or retrieving from HDFS.

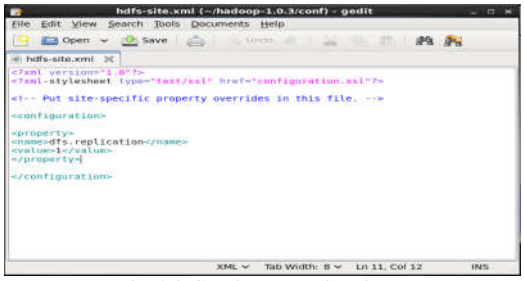

Fig 16. Setting Replications

Here we set replication. Replication means in how many machine it will store.

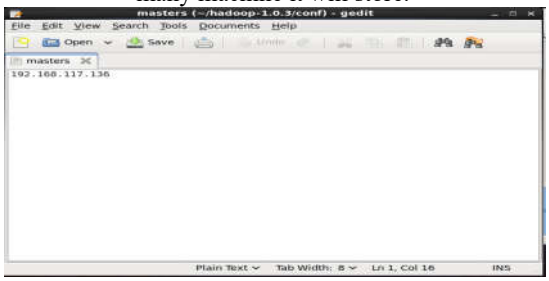

Fig 17. Setting Master node

Set slave node

Here we check for the IP address of the system.

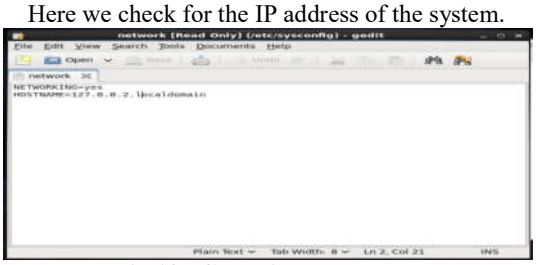

Fig 12. Changing Hostname

Here we are setting all master node and slave nodes using super user

Fig 14. Setting namenode

Here we will set the name node means master node and Data node means where it stores in the location. Lastly garbage collection also done here.

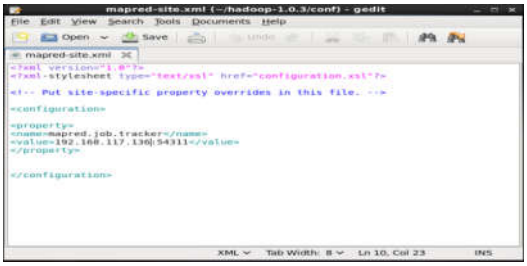

Fig 15. Setting job and task tracker

Setting Master node

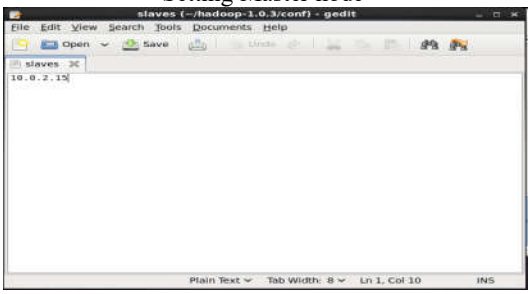

Fig 18. Setting slave node

Lastly we will format name node to refresh HDFS system.

Fig 19. Creating connection between 2 machines

Here we are creating connection between 2 machines using RSA authentication method the public key shared between both client and server.

Fig 20. Format name node to refresh HDFS system

After starting cluster it will show this

Cluster Summary

7 files and directories, 1 blocks = 8 total. Heap Size is 31.57 MB / 966.69 MB (3%)

Configured Capacity	:	35.18 GB
DFS Used	:	28.01 KB
Non DFS Used	:	7.4 GB
DFS Remaining	:	27.78 GB
DFS Used%	:	0 %
DFS Remaining%	:	78.96 %
Live Nodes	:	1
Dead Nodes	:	1
Decommissioning Nodes	:	0
Number of Under-Replicated Blocks	:	2

Fig 21. HDFS cluster summary , This will show the HDFS cluster summary

V. CONCLUSION

The important factor to store and retrieve data is system should be reliable, secure and fast computing. Here loading of data to the system is key factor. There are different type of data can be saved and computed, like structured, unstructured, semi-structured. In earlier system there is problem to store data like unstructured and computing. The proposed system is reliable and secured and fast compared to the existing system. And in HDFS storing of data is very easier compared to the existing system and also our system can save and compute structured, unstructured, semi-structured data formats.

REFERENCES

[1] Tom White, "Hadoop: The Definitive Guide", O'Reilly Media, 2012 Edition.
[2] Intel It Center, "Planning Guide- Getting started with Big Data".
[3] Academia.edu, "Processing Big Data using Hadoop Framework".
[4] Robert D. Schneider," Hadoop for Dummies"
[5] Michael G. Noll. Running Hadoop on Ubuntu Linux (Single Node Cluster) [Online]. Available: Website links: Apache Hadoop.

[6] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," in USENIXSymposium on Operating Systems Design and Implementation, San Francisco, CA, Dec. 2004, pp. 137–150.

[7] A, Katal, Wazid M, and Goudar R.H. "Big data: Issues, challenges, tools and Good practices.". Noida: 2013, pp. 404 – 409, 8-10 Aug. 2013.

[8] Michael Di Stefano, Distributed Data Management for Grid Computing, ISBN 0-471-68719-7, 2005.

[9] Yuhui Deng , Frank Wang, Na Helian, Sining Wu, Chenhan Liao (2008) "Dynamic and scalable storage management architecture for Grid Oriented Storage device" Parallel Computing 34 (2008) 17-31.

[10] Jimmy Lin MapReduce Is Good Enough?, The control project. IEEE Computer 32 (2013).

[11] "Cloud Security Alliance Top Ten Big Data Security And Privacy Challenges "b CSA Big Data Working Group.

[12] Venkata Narasimha Inukollu1 , Sailaja Arsi1 and Srinivasa Rao Ravuri "SECURITY ISSUES ASSOCIATED WITH BIG DATA IN CLOUD COMPUTING" International Journal of Network Security & Its Applications (IJNSA), Vol.6, No.3, May 2014.

[13] Hadoop.apach.org

[14] K.. T. Smith, "Big Data Security: The Evolution of Hadoop's Security Model," InfoQ http://www.infoq.com/articles/HadoopSecurityModel, aug. 2014.

[15] Priya P. Sharma, Chandrakant P. Navdeti, (2014), " Securing Big Data Hadoop: A Review of Security Issues, Threats and Solution", IJCSIT, 5(2), pp2126- 2131

[16] Richa Gupta, Sunny Gupta, Anuradha Singhal, (2014), "Big Data:Overview", IJCTT, 9 (5)