ISSN No. 0976-5697

Volume 4, No. 11, Nov-Dec 2013

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Enhanced Speed Processing of Data using in-Memory Analytics

T.Selva Divya
Assistant Professor, Department of IT
Kalaignar Karunanidhi Institute of technology,India
divyathangavel@gmail.com

M.Jayanthi
Assistant Professor, Department of IT
Kalaignar Karunanidhi Institute of technology, India
jayanthi222@gmail.com

V.Vinoth Kumar
Assistant Professor, Department of IT
Kalaignar Karunanidhi Institute of Technology, Coimbatore
vinothillam@gmail.com

Abstract: Database is used to store large amount of data. When the data size is growing, it is difficult to process using traditional data processing application or tools. Today many organizations's information is growing and they need a large data tools to store a huge amount of data. So, big data tools arrive because of the drawback for storage and processing. Hadoop is an open source software and support many applications which support petabyte sized analytics. This paper deals with working of Hadoop. HDFS is used for data storage in Hadoop. It distributes the work to nodes and communicates with a single named server node and if the name server goes offline. HDFS must restart where it left out and it causes some latency or delay in work of the system. Spark solves the problem of HDFS. It is a column oriented distributed database and has a fault tolerant than HDFS. Spark is In-memory database where the queries of data are retrieved from RAM instead of physical disk the processing speed of spark is much faster than Hadoop system. Map reduce is used for data processing and it splits the work and reduces it into single subset.

Keywords: Big Data, Hadoop System, HDFS, MapReduce, Spark, Cassandra, Databases.

I. INTRODUCTION

Last three decade and even today, Relational database management systems technology plays a major role for online digital media, financial systems, insurance, health care, and transportation etc. The amount of data collected and analyzed in enterprises has increased several folds in volume, variety and velocity. As an outcome, limitation of traditional RDMS occurs. So a new class of systems had to be designed and implemented, giving rise to the new phenomenon of "Big Data". "Bigness" of Big Data, volume of data is only one factor in the requirements of modern data processing platforms. Big Data is high volume, high velocity, and high variety information assets that demand cost effective, innovative forms of information processing for enhanced insights and decision making. A study predicts that the data is generated in digital form will double every two years. A major factor behind this data growth is connectivity via rapidly growing reach of mobile devices, constantly connected to the networks. A vast amount of data is not created by the users, but by the digital universe and it is stored, managed and analyzed by the enterprises such as Internet service providers and cloud service providers. Big data is a collection of large and complex datasets and it become difficult to process using traditional data processing applications. Big data includes data sets with terabytes to many petabytes of data sizes for capturing, managing and processing the large amount of data within an elapsed time. Big data tools are Hadoop, HPPC (High performance computing clusters), and Storm. Big data are high volume, velocity and variety of information [1].

Multidimensional big data is represented as tensors. Big data usually includes data sets with sizes beyond the ability of commonly used software tools to capture, curate, manage, and process the data within a tolerable elapsed time. Big data sizes are a constantly moving target, as of 2012 ranging from a few dozen terabytes to many petabytes of data in a single data set. With this difficulty, new platforms of "big data" tools are being developed to handle various aspects of large quantities of data.

Examples include Big Science, web logs, RFID, sensor networks, social networks, social data (due to the social data revolution), Internet text and documents, Internet search indexing, call detail records, astronomy, atmospheric science, genomics, biogeochemical, biological, and other complex and often interdisciplinary scientific research, military surveillance, forecasting drive times for new home buyers, medical records, photography archives, video archives, and large-scale e-commerce. The challenges Google faced in crawling the web, storing, indexing, ranking and serving billions of web pages could not be solved with traditional Data management systems. The content of Google's search index is exploded and consists of text, images, video, geo-spatial and even renderings of structured data. In 2003 and 2004, Google published details of Google File system and Map Reduce programming framework. Hadoop is developed and became a de facto standard for big data platforms deployed today. The main value that can be derived from Big Data is by aggregating vast amount of data integrated from various sources.Fig.1 shows various technologies used in Big Data. Most of the Big Data processing technologies include Machine learning algorithms and natural language processing algorithms.

© 2010, IJARCS All Rights Reserved

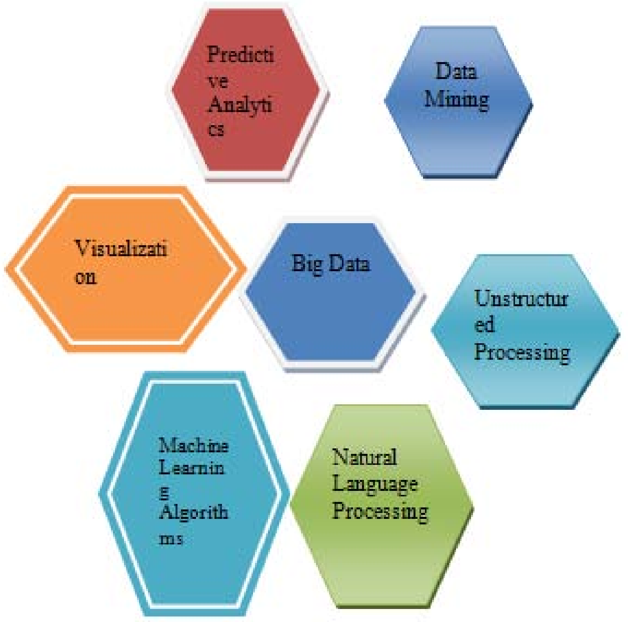

Figure.1: Different technologies in Big Data

II. HADOOP SYSTEM

A. Architecture:

Hadoop is used to run on a large number of machines and does not share any memory is shown in Fig.2. When the data are stored in to Hadoop, it splits into pieces of work and spreads it across different servers. Hadoop monitors every movement of data and it knows where the data resides [2].

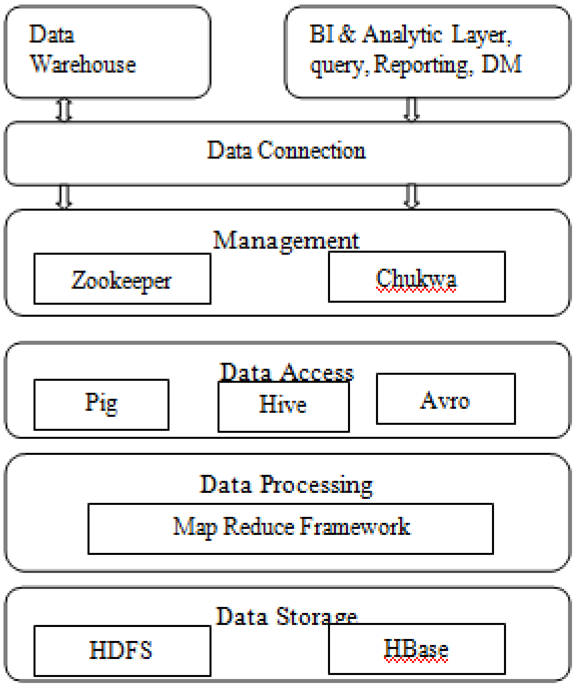

Figure 2. Hadoop System

a. Working of Hadoop:[3]

a) Data is stored in to Hadoop.

b) Hadoop split and distributes the data across multiple machines.

c) Hadoop MapReduce perform the distributed queries on the data is shown in Fig.3. Map reduce is an effective framework for processing data.

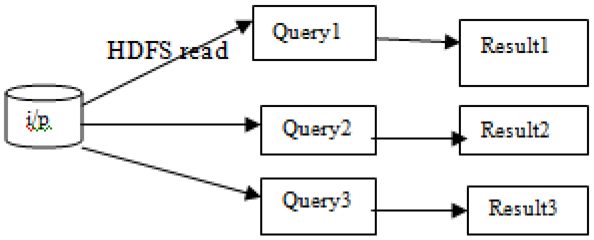

Figure.3:Data Sharing in MapReduce

Moving all the big data to one storage area network or ETL server is not advisable for the big data volumes. Even though, the movements of the data slow down the bandwidth. With Hadoop, the raw data is stored directly to low cost servers one time and only the results or output is passed to the other systems. Hadoop systems sit side by side with data warehouses.

Map and Reduce process distribute the work across the nodes shown in Fig.3 [4]. This allows parallel processing. With Massively Parallel processing, Hadoop can run on different servers with less cost. As every node return their result, reduce function combines the data to deliver a final result.

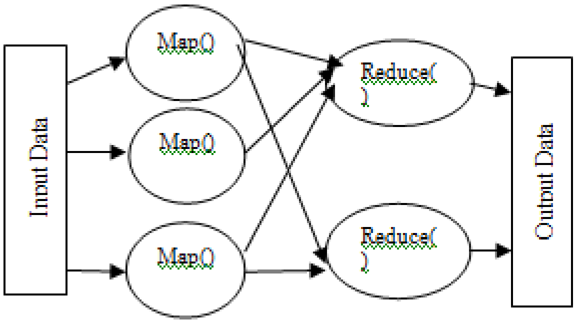

Figure.3:Map/Reduce DAG

B. Data warehouses Vs Hadoop:

Hadoop is a storehouse and refinery for raw data. The first step is to get all the reliable and cost effective data from the sources such as sensors, machines, social media and etc. Even the large and huge amount of data is stored in it, data access is often much faster. Hadoop is a powerful and dynamic archive.

Hadoop is used to maintain the large amount of data without any loss. Hadoop storage grid is useful both for capturing the raw data and long term storage of data. Hadoop sits at both split ends of big scale life cycle. Hadoop and the data warehouse will often work together in a single information supply chain. When it comes to Big Data, Hadoop excels in handling raw, unstructured and complex data with vast programming flexibility.

Data warehouses also manage big structured data, are shown in Fig4, integrating subject areas and providing interactive performance through BI tools[5]. ETL processing runs in parallel across the entire cluster results in much faster operations. MapReduce program issue the SQL statements to data warehouse. Map reduce is just a program.

© 2010, IJARCS All Rights Reserved

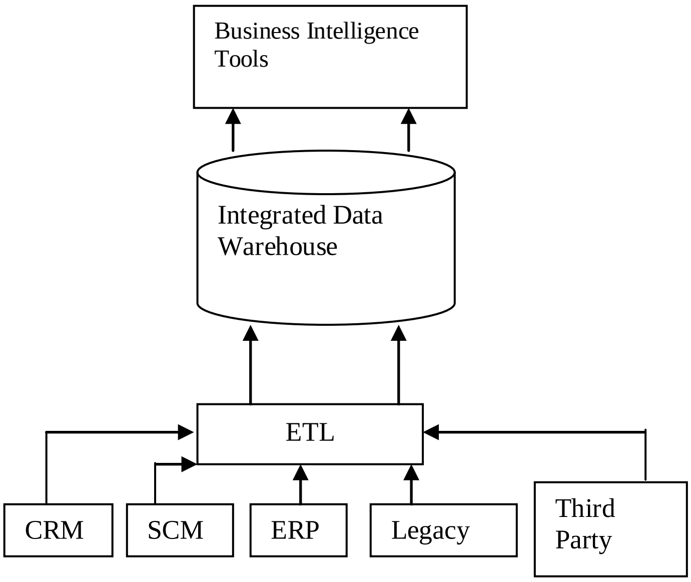

Figure 4. Data Warehouse

C. Hdfs:

It is a distributed file system and it is capable of storing large size of files with the sequential read and writes operation [6]. Each file is divided into chunks and it is stored across different multiple datanodes is shown in Fig 5. HDFS is present at the bottom layer in the Hadoop system. There is a master namenode who monitors overall file directory and where the chunks present. Name node act as a central node and can distribute and redistribute the duplicates as needed. Name node cannot take overall control of every request from the user and so it maintains a data node. It informs all its chunks to the name node at boot up. Each data node contains chunks of data and each chunk has version number for all its updates. If any crash occurs in Data Node it is known to Name node and later chunks will be garbage collected at later time. When the users are request to read a file, Name node will start the process. The namenode will reply which datanode have the copy of the client queries. So that the client or user can contact respective datanode directly without contacting Namenode further.

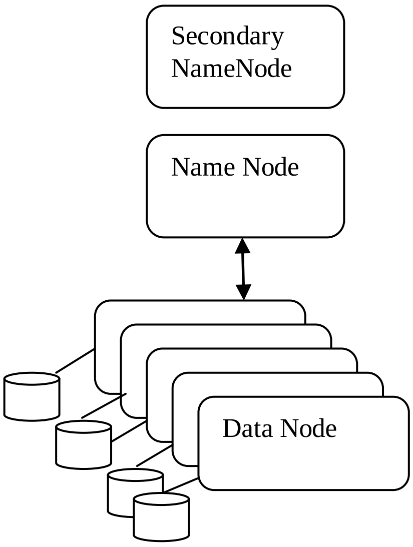

Figure.5: HDFS System

If the user needs to write the file, he must contact namenode and will select the primary and secondary replicas. If the client modification can be done to all datanodes in any order, but the changes will be stored in buffer of each datanode. All changes will be written to an operation log file after the commit request to primary and secondary. This log file maintain list of operation to recover if any crash occurs. Name node also maintains the state of an operation. In case of Namenode crash a new Namenode will take the responsibility for restoring the state from the last check point and replay the log file.

D. HBase:

HBase is a column oriented DBMS and it runs on the top of HDFS [7]. HBase is distributed and not a relational database and structured query language does not work in it. It consists of set of tables and each table contains rows and columns. It is similar to the tradional database. Each table has a primary key for each element and the user can access to the table only by using primary key. It is different from row oriented relational database. It allows attributes are grouped together and get into the column families. HBase predefine the entire table schema and specify the column families. It allows adding new families at any time and able to adapt to the changing requirements. HDFS has masternode and slave node for performing the operations. Mapreduce has job tracker and task traker slaves. Similarly HBase has master node, it manages the cluster and servers store the tables and do the work on the data. HBase is sensitive to loss of master data. It is well suited for real time data analysis and get the input and output for mapreduce jobs run in Hadoop.

E. Data Access:

a. PIG:

Pig is a high level programming language and high level data flow language [8]. It is an execution framework for parallel computation. It runs at top of HDFS and works within Hadoop and Mapreduce frameworks.

b. HIVE:

It is used to access the data and it is a data warehouse infrastructure [11][12].It provides Query and Summarization for Hadoop. It uses a SQL like Query Language to execute map reduce jobs and mainly uses storage and execution modules. Hive support any file system. Hive Queries are first converted to a graph of Hadoop MapReduce jobs that will be executed on Hadoop grid.Hive Query Language is used to access the data and the constructs are SHOW, DESCRIBE, SELECT, USE and JOIN. It not only contains predefined functions or constructs.The other user defined functions and serialization and Deserialization functions are used. UDF allows programmers to write their functions and connect a Hive Query language with functional and scripting language. It contains metadata to store and keep the information that links. The Hive Queries are listed. You want to create a database.

```
hive> CREATE DATABASE demo;
Create a table with a schema
hive> CREATE TABLE demoTable (foo INT, bar
STRING);
hive> describe demoTable
```

© 2010, IJARCS All Rights Reserved

c. *AVRO:*

It moves the data over the network. It is a serialization system that converts data in to fast, compact binary format [10].

F. *Management:*

a. *Zookeeper and Chukwa:*

Zookeeper gives coordination, configuration and group services for distributed applications. Chukwa is used to monitor the system.

III. PROPOSED WORK

A. *Drawback of Hadoop:*

HDFS communicate with a single named server node and creating a single point of failure if the named server goes offline. When this occurs, HDFS instance must restart and pick up where it left out, causing delay in the work processes of the system.

B. *Cassandra:*

It is a column oriented and scalable distributed database [8]. It has no master server and there is high error tolerance than HDFS. Cassandra has some advantages:

a. It has high fault tolerance
b. NOSQL family
c. Read and Write throughput is high
d. Replication is accepted and it is flexible than HDFS

Cassandra gives its best but the processing speed is slow than streaming spark.

C. *SPARK:*

It replaces Cassandra and HDFS. Cassandra is better than HDFS in fault tolerance. Spark is better than Cassandra. Spark processing speed is faster than Hadoop [13].The difference between Row Storage and Column Storage are shown in Fig6a) and 6b).

1	Name1	Age1
2	Name2	Age2
3	Name3	Age3

Figure .6a) Row Storage

1	2	3
Name1	Name2	Name3
Age1	Age2	Age3

Figure.6b) Column Storage

The user's requests are processed and the results are driven in faster manner. MapReduce slow down the tasks due to duplication, serialization and disk I/O.In Business intelligence (BI), in-memory analytics is a methodology used to solve difficult business problems. The data is stored in RAM to maximize the speed, performance and reliability. Mostly the Business Intelligence is disk based that the data or queries are loaded on disks. But In-memory analytics, the data are stored in RAM. It is accepted in 64-bit architecture, so the memory space is large compared to 32-bit. It is designed mainly for the queries which take long time to process the data in large database. It is in-memory data storage is shown in Fig7.

.

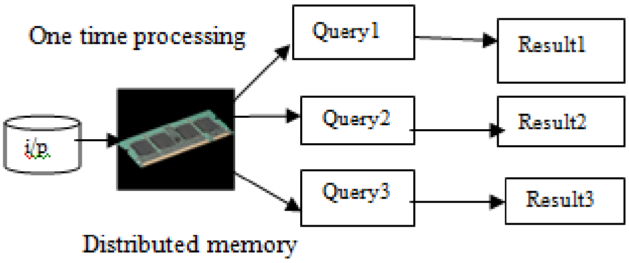

Fig.7: Data sharing in Spark

D. *Streaming SPARK:*

It extends spark to perform streaming computations [14]. The collection of objects or data can be stored in memory across cluster nodes is show in Fig8. It automatically restarts and recovers from failures. To recompute the lost data, the resilient Distributed Dataset follow the series of transformation and it is used to build them [15].

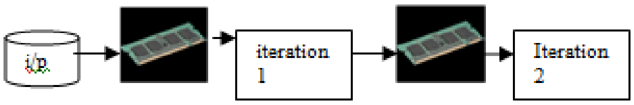

Figure.8: Distributed Memory

Table1. Comparison of different database

Distributed Databases	Iterations	Error Tolerant	Speed
HDFS	127 s/ iteration	Less	Low
CASSANDRA	127 s/iteration	Medium	Medium
SPARK	174s/iteration	High	10-100 times faster
STREAMING SPARK	175 s/iterations	High	10-100 times faster

IV. CONCLUSION

Big Data is emerging today. Many tools are available. Hadoop System deals mainly with HDFS and MapReduce. HDFS is a distributed file system is mainly designed to handle large files with continuous read and write operation. MapReduce splits the tasks across nodes and Reduce function reduce the result into single set. This paper concentrates on problem in Hadoop system and the comparison of different databases. Cassandra is a column oriented distributed database. It has a high error tolerant. Spark is an in-memory database where the information is directly stored in RAM instead of network or disks. so, the processing time in spark is low.

V. REFERENCES

[1]. Andrew Lamb, Matt Fuller, Ramakrishna Varadarajan Nga Tran, Ben Vandiver, Lyric Doshi, Chuck Bear"The Vertica Analytic Database: CStore 7 Years Later" Aug 27, 2012

[2]. Larn C,Warren "J.Hadoop in Action".Manning Publications,2010

[3]. The Hadoop Distributed File System: Architecture and Design," The Apache Software Foundation",2007

© 2010, IJARCS All Rights Reserved

[4]. T.Nykiel,M.Potamias,C.Mishra,G.Kollios, and N. Koudas. MRShare: Sharing Across Multiple Queries in MapReduce. PVLDB, 3(1),2010.

[5]. D.Jiang,B.C.Ooi,L.Shi, and S.Wu. The performance of MapReduce:An In-depth Study.PVLDB,3(1), 2010.

[6]. Jerry Zhao, Jelena Pjesivac-Grbovic," MapReduce The Programming Model and Practice"June 19th 2009.

[7]. Colin White Dr.AmrAwadallah,Founder and CTO,cloudera and Dan Graham,General Manager,Enterprise Systems, Teradata corporation,"Hadoop and the Datawarehouse: When to use which MapReduce and the Data Scientist", 2012

[8]. Comparing the Hadoop Distributed File System (HDFS) with the Cassandra File System (CFS),White Paper BY DATASTAX CORPORATION August 2013, Hadoop HDFS," Hadoop.Apache.org: http://hadoop.apache.org/hdfs/.

[9]. Drew O'Brien, http://vision.cloudera.com/hadoop-selected-for-the-infoworld,2012

[10]. Jimmy Lin,"Bigtable, Hive, and Pig Data-Intensive Information Processing Applications "April 27, 2010

[11]. Ashish Thusoo, Joydeep Sen Sarma, Namit Jain, Zheng Shao, Prasad Chakka, Ning Zhang, Suresh Antony, Hao Liu and Raghotham Murthy ," Hive A Petabyte Scale Data Warehouse Using Hadoop"2008

[12]. Jens Pillgram Larsen, "Hadoop_ fundamentals_an_introduction_to_hive.htm" Dec 3, 2012

[13]. Mateiz Zaharia et al, "Spark and Shark High Speed In-Memory Analytics over Hadoop and Hive Data"2012

[14]. Jesper H. Spring,Jean Privat,Rachid Guerraoui" StreamFlex: High-throughput Stream Programming in Java",2007

[15]. Tathagata Das (TD) UC Berkeley,"Spark Streaming Large Scale near-realtime stream processing"

© 2010, IJARCS All Rights Reserved