A Review of Hadoop Ecosystem for BigData

Ashlesha S. Nagdive
Research Scholar
Sant Gadge Baba Amravati University, India

R. M. Tugnayat, PhD
Principal
Shri Shankarprasad Agnihotri College of Engineering
Wardha, India

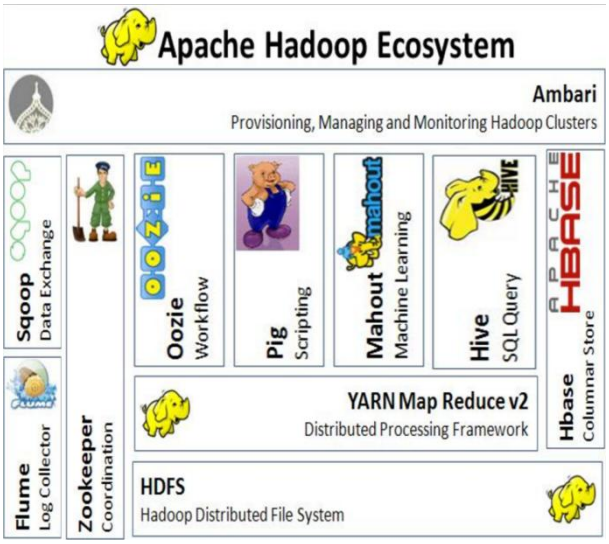
ABSTRACT
This paper, describes Concept of Big Data which is collection of large data set that cannot be proceed by traditional computational techniques. Therefore Hadoop technology designed to process Big Data. Hadoop is the platform in businesses for Big Data processing. Hadoop is an open source, Java-based programming framework which supports the processing and storage of extremely large data sets in a distributed computing environment. It helps Big Data analytics by overcoming the difficulties that are usually faced in handling Big Data. Hadoop can break down large computational problems into smaller tasks as smaller elements can be analyzed economically and quickly[1]. Hadoop is an open-source software framework for storing data and running applications on clusters of commodity hardware. It provides massive storage for various kind of data, enormous processing power and the ability to handle virtually limitless concurrent tasks. All these parts are analyzed in parallel and the results of the analysis are regrouped to produce the final output.

Keywords
Big Data, Hadoop Architecture, Apache Hadoop, Mapreduce, Hadoop Ecosystem, Hadoop Distributed File System (HDFS).

1. INTRODUCTION
Big data is unstructured voluminous data set that traditional data processing application software are inadequate to deal with them. Big data challenges include capturing data, storage, data storage, analysis, data search, sharing, transfer, visualization, querying, updating and information privacy. The three dimensions to big data known as Volume, Variety and Velocity. Big data technologies are important in providing more accurate analysis, which may lead to more concrete decision-making resulting in greater operational efficiencies, cost reductions, and reduced risks for the business. Hadoop is an open-source framework that stores and processes big data in a distributed environment[1]. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage.

Apache Hadoop is born to enhance the usage and solve major issues of big data. The web media is generating huge information per day, and it was becoming very difficult to manage the data of around one billion pages of content. Apache Hadoop is an open source software platform for distributed storage and distributed processing of unstructured data sets on computer clusters built from commodity hardware. Hadoop services provide for data storage, data processing, data access, data governance, security, and operations. Hadoop biggest strength is scalability. It upgrades from working on a single node to thousands of nodes without any inconvenience. The different domains of Big Data means are able to manage the data's are from videos, text medium, transactional data, sensor information, statistical data, social media conversations, search engine queries, ecommerce data, financial information, weather data, news updates, forum discussions, executive reports, and so on[2].Hadoop runs the applications on the basis of Map Reduce where the data is processed in parallel and accomplish the entire statistical analysis on large amount of data.

2. HADOOP ECOSYSTEM
Handling huge volume of data generating from billions on online activities and transactions require continuous up gradation and evolution of Big data. Hadoop ecosystem is a framework of various type of complex and evolving tools and techniques. Mapreduce and Hadoop Distributed File System(HDFS) are two components of Hadoop ecosystem which manages big data.Fig1 depicts element in Big data. All these elements enable users to process large datasets in real time and provide tools to support various Hadoop projects, schedule jobs and manages cluster resources.



Source: http://blog.agro-know.com/?p=3810

Fig 1: Apache Hadoop Ecosystem

Hadoop framework includes following four modules as shown in Figure1:

a)Hadoop Common: These are Java libraries and utilities required by Hadoop modules. These libraries provides file system and OS level abstractions that contains the necessary Java files and scripts required to start Hadoop.

b)Hadoop YARN: It is a framework for job scheduling and cluster resource management.

c)Hadoop Distributed File System (HDFS): A distributed file

system that provides high-throughput access to application data.

d)Hadoop Map Reduce :It is YARN-based system for parallel processing of large data sets.

Hadoop skill set requires thoughtful knowledge of every layer in the Hadoop stack from understanding about the various components in the Hadoop architecture, designing a Hadoop cluster, responsible for data processing.

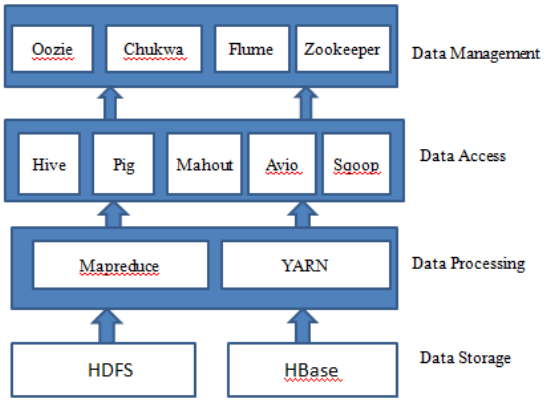


Fig2: Hadoop Ecosystem Elements at various stage of Data Processing

There are other projects included in the Hadoop module which includes:

a)Apache Ambari : It is a tool for managing, monitoring and provisioning of the Hadoop clusters. It supports the HDFS and Map Reduce programs.

b) Cassandra: It is a distributed system to handle extremely huge amount of data which is stored across several commodity servers.

c)HBase: It is a non-relational, distributed database management system that works efficiently on sparse data sets and it is highly scalable.

d)Apache Spark: It is highly agile, scalable and secure the Big Data compute engine, versatile the sufficient work on a wide variety of applications like real-time processing, machine learning, ETL.

e)Hive: It is a data warehouse tool basically used for analyzing, querying and summarizing of analyzed data concepts on top of the Hadoop framework.

f)Pig: Pig is a high-level framework which ensures us to work in coordination either with Apache Spark or Map Reduce to analyze the data.

g) Sqoop: It is framework is used for transferring the data to Hadoop from relational databases. This application is based on a command-line interface.

h)Oozie: It is scheduling system for workflow management, executing workflow routes for successful completion of the task in a Hadoop.

i)Zookeeper: It is an Open source centralized service which is used to provide coordination between distributed applications of Hadoop.

3. HADOOP FRAMEWORK

3.1 Hadoop Distributed File System

File systems like HDFS are designed to manage the challenges of Big Data. Being a core component, Hadoop , MapReduce and HDFS are always being enanced which provide great stability. Hadoop stores and manages petabytes of data using the HDFS. Using HDFS it is possible to connect commodity hardware or system also known as nodes. These nodes are connected over a cluster on which the data files are stored in a distributed way. Using the power of HDFS whole cluster and the nodes can be easily accessed for data storage and processing using the MapReduce process[5]. The Hadoop Distributed File System (HDFS) is based on the Google File System (GFS) and provides a distributed file system that is designed to run on large clusters of small computer machines in a reliable, fault-tolerant way. HDFS is the file system intended for putting away huge documents with streaming information access.

HDFS uses a master/slave architecture where master consists of a single NameNode which manages the file system metadata and one or more slave DataNodes which store the actual data. A file in an HDFS namespace is split into many blocks and those blocks are stored in a set of DataNodes. The NameNode determines the mapping of blocks to the DataNodes. The DataNodes identify read and write operation with the file system. It manages block creation, deletion and replication based on instruction given by NameNode. HDFS provides a shell like other file system and a list of commands available to interact with the file system.

3.2 Hadoop Working

Hadoop follows a master slave architecture design as shown in Fig2 for data storage and distributed data processing using HDFS and Map Reduce. The master node for data storage is Hadoop HDFS is the Name Node and the master node for parallel processing of data using Hadoop Map Reduce is the Job Tracker[3]. The slave nodes in Hadoop architecture are the other machines in the Hadoop cluster which store data and perform complex computations. Every slave node has a Task Tracker and a Data Node that synchronizes the processes with the Job Tracker and Name Node. In Hadoop architectural implementation the master or slave systems can be setup in the cloud[4].

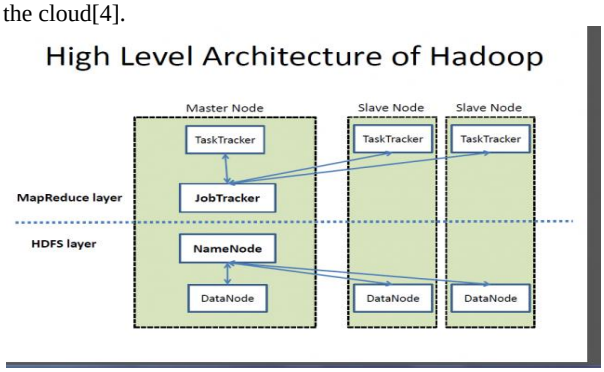


Fig3:High Level Architecture of Hadoop.

Stage1: User or application submits a job to the Hadoop for required process by specifying the following parameters:

1.The location of input and output files in the distributed file system.

2.The java classes in the form of jar file containing the implementation of map and reduce functions.

3.The job configuration by setting different parameters

specific to the job.

Stage2: The Hadoop job client then submits the job (jar/executable) and configuration to the JobTracker which assumes the responsibility of distributing the configuration to the slaves, scheduling tasks and monitoring , which provides status and diagnostic information to the job-client[3].

Stage3: The TaskTrackers on different nodes execute the task as per MapReduce implementation and output of the reduce function is stored into the output files on the file system[3].

3.3 Hadoop Environment Setup

Hadoop is supported by GNU/Linux platform and its flavors. Therefore, we have to install a Linux operating system for setting up Hadoop environment. In case you have an OS other than Linux, you can install a Virtualbox software in it and have Linux inside the Virtualbox.

Hadoop Installation Setup[3]

A. Installation of cloudera CDH 5.8

https://www.cloudera.com/downloads/quickstart_vms/5-8.html

After download tar file of cloudera CDH5.8 vm image file. usingvmware player install it given below video link.

https://www.youtube.com/watch?v=4XBXJpYPkUk

B.Ubuntu Installation:

1.downloadubuntu 16.04 from this link

http://www.ubuntu.com/download/desktop/contribute?version=16.04.1&architecture=amd64

https://www.youtube.com/watch?v=KfOt2As6apQ

Hadoop Installation:

2.InstallHadoop 2.7 using following given steps below:

3.Open command line terminal using ctrl+shift+ T

•Installing Oracle Java 8: run the below command on $ shell

sudo add-apt-repository ppa:webupd8team/java

sudo apt-get update

sudo apt-get install oracle-java8-installer

•Installing SSH

sudo apt-get install openssh-server

•Configuring SSH

ssh-keygen -t rsa -P ""

cat$HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys

4.Download latest Apache Hadoop source from Apache mirrors

First you need to download hadoop 2.7.3 binary file from the give path given below

http://hadoop.apache.org/releases.html

•Copy the Hadoop 2.7.3 folder tar file in home directory-> /home/username/Work

5.User profile :sudonano ~/.bashrc

-- HADOOP ENVIRONMENT VARIABLES START --

export JAVA_HOME=/usr/lib/jvm/java-8-oracle

export HADOOP_HOME=/home/username/Work

export PATH=$PATH:$HADOOP_HOME/bin

export PATH=$PATH:$HADOOP_HOME/sbin

export HADOOP_MAPRED_HOME=$HADOOP_HOME

export HADOOP_COMMON_HOME=$HADOOP_HOME

export HADOOP_HDFS_HOME=$HADOOP_HOME

export YARN_HOME=$HADOOP_HOME

export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native

export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"

-- HADOOP ENVIRONMENT VARIABLES END --

6.Commit the changes of .bashrc

Source ~/.bashrc

7.Configuration file : hadoop-env.sh

To edit file, fire the below given command

hduser@pingax:/home/username/Work/hadoop2.7.3/hadoop/etc/hadoop$ sudogedit hadoop-env.sh

Update JAVA_HOME variable,

JAVA_HOME=/usr/lib/jvm/java-8-oracle

8.Configuration file : core-site.xml

To edit file, fire the below given command

hduser@pingax:/home/username/Work/hadoop2.7.3/hadoop/etc/hadoop$ sudogedit core-site.xml

Paste these lines into <configuration> tag

<property>

<name>fs.default.name</name>

<value>hdfs://localhost:9000</value>

</property>

9.Configuration file : hdfs-site.xml

To edit file, fire the below given command

hduser@pingax:/home/username/Work/hadoop2.7.3/hadoop/etc/hadoop$ sudogedit hdfs-site.xml

Paste these lines into <configuration> tag

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

<name>dfs.namenode.name.dir</name>

<value>file:/home/username/hadoop_tmp/hdfs/namenode</value>

</property>

<property>

<name>dfs.datanode.data.dir</name>

<value>file:/home/username/hadoop_tmp/hdfs/datanode</value>

</property>

10.Configuration file : yarn-site.xml

To edit file, fire the below given command

hduser@pingax:/home/username/Work/hadoop2.7.3/hadoop/etc/hadoop$ sudogedit yarn-site.xml\

Paste these lines into <configuration> tag

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce_shuffle</value>

</property>

<property>

<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name><value>org.apache.hadoop.mapred.ShuffleHandler</value>

</property>

11.Configuration file : mapred-site.xml

Copy template of mapred-site.xml.template file

cp /home/username/Work/hadoop2.7.3/hadoop/etc/hadoop/mapred-site.xml.template:/home/username/Work/hadoop2.7.3/hadoop/etc/hadoop /mapred-site.xml

To edit file, fire the below given command

hduser@pingax:/home/username/Work/hadoop2.7.3/hadoop/etc/hadoop$ sudogedit mapred-site.xml

Paste these lines into <configuration> tag

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

12.FormatNamenode

/home/username/Work/hadoop2.7.3/hadoop/etc/hadoop$hdfsnamenode -format

13.Start all Hadoop daemons

Start hdfs daemons

/home/username/Work/hadoop2.7.3/hadoop/etc/hadoop$ start-dfs.sh

Start MapReduce daemons:

/home/username/Work/hadoop2.7.3/hadoop/etc/hadoop$ start-yarn.sh

14.Track/Monitor/Verify

Verify Hadoop daemons:

/home/username/Work/hadoop2.7.3/hadoop/etc/hadoop$ jps

3.4 Role of Distributed Storage

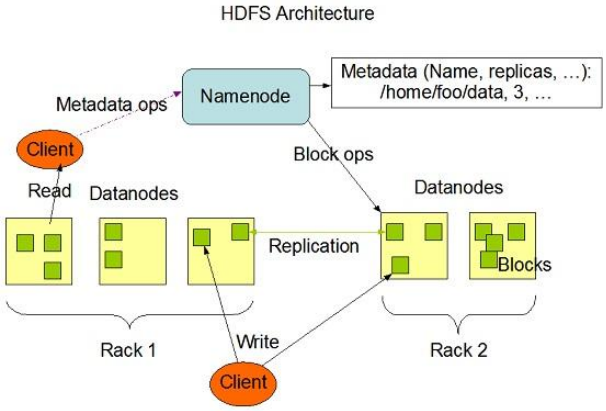


Figure4:HDFS Distributed Storage

A file on HDFS splits into multiple bocks and each is replicated within the Hadoop cluster. A block on HDFS is a block of data within the underlying file system with a default size of 64MB.The size of a block can be extended up to 256 MB based on the specific requirements. Hadoop Distributed File System (HDFS) stores the application data and file system metadata separately on dedicated servers[4]. NameNode and DataNode are the two essential components of the Hadoop HDFS architecture. Application data is stored on servers referred to as DataNodes and file system metadata is stored on servers referred to as NameNode. HDFS replicates the file content on multiple DataNodes based on the replication factor to ensure data reliability. For the Hadoop architecture to be performance efficient, HDFS must satisfy certain pre-requisites.

- All the hard drives must be having a high throughput.
- Good network speed to manage data transfer and block replications.

NameNode
All files and directories in HDFS namespace is represented on the NameNode by Inodes that contain various attributes like permissions, modification timestamp, disk space quota, namespace quota and access times. NameNode maps the entire file system structure into memory. Two files fsimage and edits are used for persistence during restarts.Fsimage file contains the Inodes and the list of blocks which define the metadata. The edits file contains any modifications that have been performed on the content of the fsimage file.When the NameNode starts, fsimage file is loaded and then the contents of the edits file are applied to recover the latest state of the file system[1].

If the hadoop cluster has not been restarted for months together then there will be a huge downtime as the size of the edits file will be increase. This is when Secondary NameNode comes to the rescue. Secondary NameNode gets the fsimage and edits log from the primary NameNode at regular intervals and loads both the fsimage and edit logs file to the main memory by applying each operation from edits log file to fsimage. Secondary NameNode copies the new fsimage file to the primary NameNode and also will update the modified time of the fsimage file to fstime file to track when then fsimage file has been updated[2].

DataNode
DataNode manages the state of an HDFS node and interacts with the blocks.A DataNode can perform CPU intensive

jobs.Jobs like semantic and language analysis, statistics and machine learning tasks, and I/O intensive jobs like clustering, data import, data export, search, decompression, and indexing. A DataNode needs many I/O for data processing and transfer[3].

On startup every DataNode connects to the NameNode and performs a handshake which verifies the namespace ID and the software version of the DataNode. If either of them does not match then ultimately the DataNode shuts down automatically. A DataNode verifies the block replicas in its ownership by sending a block report to the NameNode. As soon as the DataNode registers, the first block report is sent. DataNode sends heartbeat to the NameNode every 3 seconds to confirm that the DataNode is operating and the block replicas it hosts are available[4].

3.5 Working of Hadoop Mapreduce Architecture

MapReduce

Hadoop MapReduce is a software framework for writing applications which process big amounts of data in-parallel on larger clusters of commodity hardware in a reliable, fault-tolerant manner[4]. The term Map Reduce actually refers to the following two different tasks that Hadoop programs perform:

a)The Map Task: This is the first task, which takes input data and converts it into a set of data, in which individual elements are broken down into tuples.

b)The Reduce Task: This task takes the output from a map task as input and combines those data tuples into a smaller set of tuples. The reduce task is always performed after the map task.

Both the input and the output are stored in a file-system. The framework takes care of scheduling tasks, monitoring them and re-executes the failed tasks. The Map Reduce framework consists of a single master Job Tracker and one slave Task Tracker per cluster-node[3]. The master is responsible for resource management, tracking resource availability and scheduling the jobs component tasks on the slaves, monitoring them and re-executing the failed tasks. The slaves Task Tracker execute the tasks as directed by the master and provide task-status information to the master periodically. The Job Tracker is a single point of failure for the Hadoop Map Reduce service which means if Job Tracker goes down, all running jobs are halted[5].

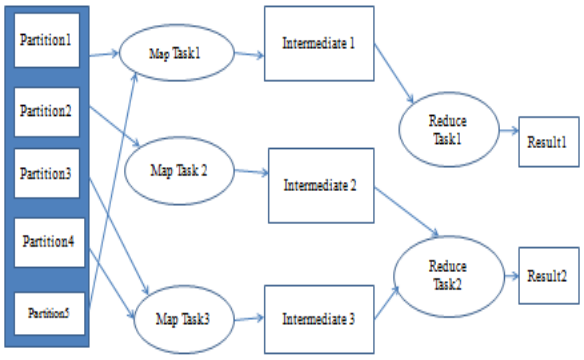


Fig5: Working of Mapeduce

MapReduce is based on the parallel programming framework to process large amount of datadispersed across different systems. The process is initiated when a user request is received to execute the MapReduce program and terminated once the results are written back to HDFS. The execution of a MapReduce job begins when the client submits the job configuration to the Job Tracker which specifies the map, combine and reduce functions along with the location for input and output data as shown in Figure5. On receiving the job configuration, the job tracker identifies the number of splits based on the input path and select Task Trackers based on their network vicinity to the data sources. Job Tracker sends a request to the selected Task Trackers[3].

The processing of the Map phase begins where the Task Tracker extracts the input data from the splits. Map function is invoked for each record parsed by the "InputFormat" which produces key-value pairs in the memory buffer. The memory buffer is then sorted to different reducer nodes by invoking the combine function. On completion of the map task, Task Tracker notifies the Job Tracker[4]. When all Task Trackers are done, the Job Tracker notifies the selected Task Trackers to begin the reduce phase. Task Tracker reads the region files and sorts the key-value pairs for each key. The reduce function is then invoked which collects the aggregated values into output result.

4. ADVANTAGE OF HADOOP ECOSYSTEM

1.It provide easy access to the user to rapidly write and test the distributed systems and automatically distributes the data and works across the machines and in turn utilizes the primary parallelism of the CPU cores.

2.Hadoop library are developed to search and handle failures at the application layer.

3.Servers can be added or removed from the cluster dynamically at any point of time.

Apache Hadoop is the most popular and powerful big data tool, which provides world's best reliable storage layer – HDFS(Hadoop Distributed File System), a batch Processing engine namely MapReduce and a Resource Management Layer like YARN. Open-source – Apache Hadoop is an open source project. It means its code can be modified according to business requirements.The data storage is maintained in a distributed manner in HDFS across the cluster, data is processed in parallel on cluster of nodes. By default the three replicas of each block is stored across the cluster in Hadoop and it's changed only when required. Hadoop's fault tolerant can be examined in such cases when any node goes down, the data on that node can be recovered easily from other nodes. Because of replication of data in the cluster, data can be reliable which is stored on the cluster of machine despite machine failures. Data is available and accessible even there occurs a hardware failure due to multiple copies of data. Hadoop is highly scalable and in a unique way hardware can

be easily added to the nodes. Hadoop is not very expensive as it runs on cluster of commodity hardware. We do not require any specialized machine for it. Hadoop provides huge cost reduction since it is easy to add more nodes on the top here. So if the requirement increases, then there is an increase of nodes, without any downtime and without any much of pre planning.

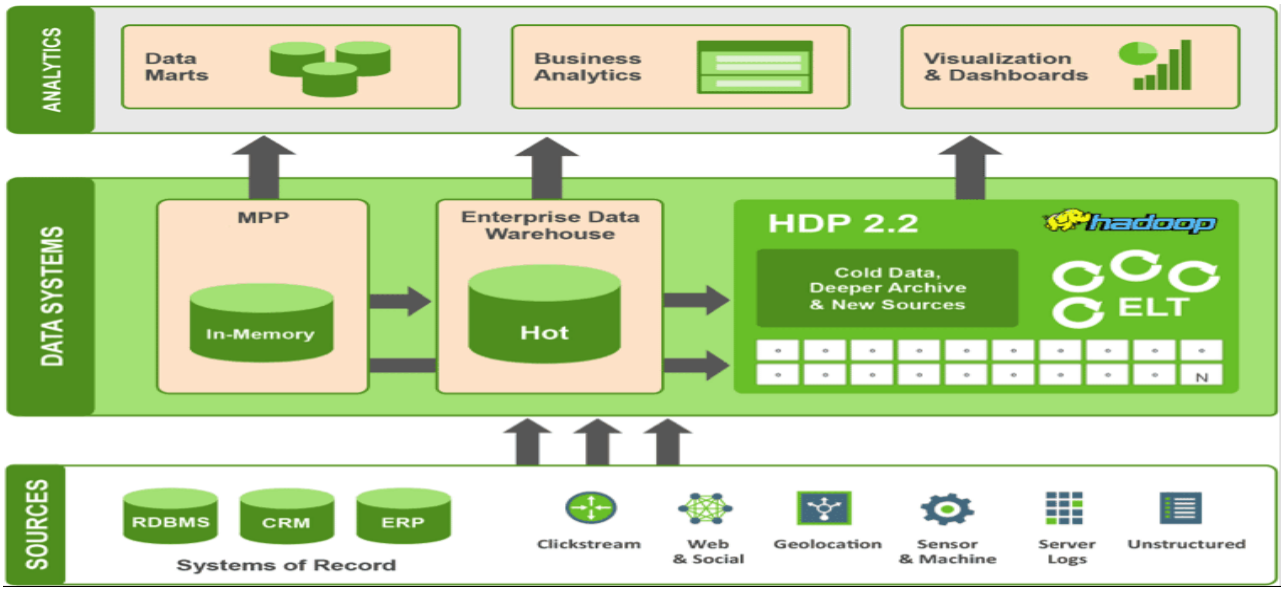


Fig 6: Optimized Data Architecture with Hadoop

5. CONCLUSION AND FUTURE WORK

This paper is about Hadoop ecosystem and has explored its major components as well as Hadoop setup. Various aspects of data storage is focused like HDFS and its architecture. The process of installation of Hadoop setup is analyzed. HDFS ensures data integrity throughout the cluster considering features like maintaining transaction logs. Another feature is validating checksum-an effective error detection technique wherein numerical value is assigned to a transmitted message on the basis of number of bits. HDFS maintains replicated copies of data blocks to avoid corruption of file due to failure of server. This paper also deals with MapReduce framework, which is an integration of different functions to sort,process and analyze bigdata.

The future research includes implementing various technologies for optimizing and improving performance on large data set. The experimental results to be analyzed using various tools and experimental setup.

6. ACKNOWLEDGMENTS

I would like to extend my deepest gratitude to my guide Dr. R M Tugnayat for his valuable guidance, suggestions and motivation. I would also like to thanks my family and friend who constantly supports in my research work.

7. REFERENCES

[1] BaoRong Chang, Yo-Ai Wang, Yun-Da Lee, and Chien-FengHuang, "Development of Multiple Big Data Analysis Platforms for Business Intelligence", Proceedings of the 2017 IEEE International Conference on Applied System Innovation

[2] Chu-Hsing Lin, Jung-Chun Liu, Tsung-Chi Peng, "Performance Evaluation of Cluster Algorithms for Big Data Analysis on Cloud", Proceedings of the 2017 IEEE International Conference on Applied System Innovation

[3] https://intellipaat.com/tutorial/hadooptutorial/introductio n- hadoop/

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

[5] Ms.Preeti Narooka, Dr.Sunita Choudhary, "Optimization of the Search Graph Using Hadoop andLinux Operating System", 2017 International Conference on Nascent Technologies in the Engineering Field (ICNTE-2017) IEEE-ICASI 2017.

[6] Yu-Sheng Su1, Ting-Jou Ding2, Jiann-Hwa Lue3, Chin-Feng Lai4, Chiu-Nan Su5,"Applying Big Data Analysis Technique to Students' Learning Behavior and Learning", Proceedings of the 2017 IEEE International Conference on Applied System Innovation IEEE-ICASI 2017.