International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 14 Issue 2 –APRIL 2015

BIG DATA STOCK ANALYSIS USING HADOOP

VEERESHETTY DAGADE[#1] ,ABHISHEK AKKATANGERHAL[*2], AMRUTA DESHPANDE[*3] , RUCHA BHANDIWAD[*4]

[#] *B.E | M.Tech | Lecturer,Computer Science and Engineering,Jain College of Engineering of Belagavi,India.*
* *B.E, Computer Science and Engineering,Jain College of Engineering of Belagavi,India.*

***Abstract* — we live in on-demand, on-command Digital universe with data prolifering from various sectors at a very high rate. This data is categoriesed as "Big Data" due to its Volume, Variety, Velocity and Veracity. Most of this data is unstructured or semi structured and it is heterogeneous in nature. Due to the huge volume and the heterogeneity of data, taking into account the speed at which it is generated, makes it tough for the existing computing infrastructure to process Big Data. This paper concentrates on the issue of the data analytics and data in stock market. We present a proper procedure for analyzing raw data received from internet using Big Data technique and Hadoop framework. We also make use of Hadoop tools Hive and Pig to analyze the data. The proposed system would benefit the current application over Data manageability, availability, performance, replication, capacity and as well as query.**

***Keywords* — Apache Hadoop, Big Data, Hive, Map Reduce, Pig, Stock data, Technical Indicators.**

I. INTRODUCTION

Big data exceeds the reach of commonly used hardware environments and software tools to capture, manage, and process it with in a tolerable elapsed time for its user population [1]. Big data refers to data sets whose size is beyond the ability of typical database software tools to capture, store, manage and analyze [2]. Big data is a collection of data sets so large and complex that it becomes difficult to process using on-hand database management tools [3].

Big Data encompasses everything from click stream data from the web to genomic and proteomic data from biological research and medicines. Big Data is a heterogeneous mix of data both structured (traditional datasets –in rows and columns like DBMS tables, CSV's and XLS's) and unstructured data like e-mail attachments, manuals, images, PDF documents, medical records such as x-rays, ECG and MRI images, forms, rich media like graphics, video and audio, contacts, forms and documents. Businesses are primarily concerned with managing unstructured data, because over 80 percent of enterprise data is unstructured and require significant storage space and effort to manage.“Big data” refers to datasets whose size is beyond the ability of typical database software tools to capture, store, manage, and analyze [3].

Big data has the following characteristics:[3]

Volume – The first important characteristics of big data. It is the size of the data which determines whether it can actually be considered Big Data or not. The name ‘Big Data’ itself indicates the data is huge.

Variety - The next aspect of Big Data is its variety. Big-Data can be generated from various fields. This means that the category to which Big Data belongs to is also a very essential fact that needs to be known by the data analysts.

Velocity - The term ‘velocity’ in the context refers to the speed of generation or processing of data. Since the speed of generation of this data is very high it increases the level of difficulty to process it.

Variability – This factor indicates that the Big Data generated may be inconsistent sometimes, making it difficult for data analysts to manage and handle the data

.

Complexity - Data management is a very complex process, especially when large volumes of data come from different sources. These data need to be linked, connected and correlated so that it can grasp the information that is supposed to be conveyed by these data. And hence the complexity of Big-Data

The stock market shows the variation of the market economy, and receives millions of investors’ attention from the time of opening development each day. The stock market is characterized by high-risk, high-yield, hence investors are concerned about the analysis of the stock market and trying to forecast the trend of the stock market. However, stock market is affected by various factors like the politics, economy , along with the complexity of its internal law, such as price changes in the non-linear, and so on therefore the traditional mathematical statistical methods to predict the stock market has not yielded suitable results. Thus, it is very suitable for the analysis of stock data.

II. APACHE HADOOP

Apache Hadoop is an open-source software framework written in Java for distributed storage and distributed processing of very large data sets on computer clusters. All the modules in Hadoop are designed with a fundamental assumption that hardware failures are common place and thus should be automatically handled in software by the framework[6].

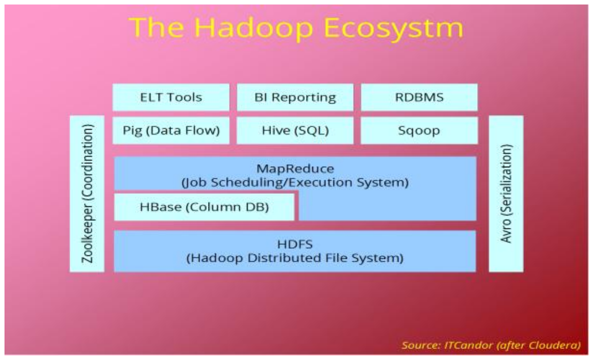
The core of Apache Hadoop consists of a storage part (Hadoop Distributed File System (HDFS)) and a processing part (MapReduce). Hadoop splits files into large blocks and distributes them amongst the nodes in the cluster. To process the data, Hadoop MapReduce transfers packaged code for nodes to process in parallel, based on the data each node needs to process. This approach takes advantage of data locality– nodes manipulating the data that they have on-hand – to allow the data to be processed faster and more efficiently than it would be in a more conventional supercomputer architecture that relies on a parallel file system where computation and data are connected via high-speed networking.

The base Apache Hadoop framework is composed of the following modules:

- Hadoop Common – contains libraries and utilities needed by other Hadoop modules;
- Hadoop Distributed File System (HDFS) – a distributed file-system that stores data on commodity machines, providing very high aggregate bandwidth across the cluster;
- Hadoop YARN – a resource-management platform responsible for managing compute resources in clusters and using them for scheduling of users' applications and
- Hadoop MapReduce – a programming model for large scale data processing.

The term "Hadoop" has come to refer not just to the base modules above, but also to the collection of additional software packages, such as Apache Pig, Apache Hive and others. The fig#1 shows the ecosystem of Hadoop architecture-

The Hadoop Ecosystem

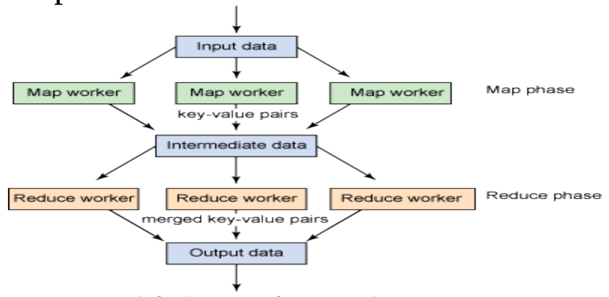


Fig#1: Hadoop ecosystem installed over EXT-4 file system.

III MAPREDUCE

MapReduce is a programming model for expressing distributed calculation on huge amount of data and an execution framework for large-scale data processing on clusters of article of trade servers. It was originally developed by Google and built on well-known principles in parallel and distributed processing. Hadoop implements the open source MapReduce written in java which provides reliable, scalable and fault tolerant distributed computing. Key-value pair forms the basic data structure in MapReduce. Keys and values are primitives such as integers, floating point values, strings, and raw bytes or they may be arbitrary complex structures (lists, tuples, associative array, etc.). Programmers can define their own data types. The map function will take the input data and will generate intermediate key and value pairs. The reduce function then takes an intermediate key and a set of values to form a smaller set of values. Typically the reducer produces only zero or one output value.
MapReduce framework is responsible for automatically splitting the input, distributing each chunk to mappers on multiple machines, grouping and arrangement all intermediate values related with the intermediate key, passing these values to reducers on multiple resources. Master monitors the functioning of the mapper and reducer and re-executes them on failure. The MapReduce jobs have thousands of individual tasks which have to be assigned to nodes in the cluster. When there are large number of jobs, the total number of tasks existing may also exceed the number of tasks that can be run on the clusters concurrently, thus makes it necessary for the scheduler to maintain a task queue in order to track the progress of running tasks, so that waiting tasks can be assigned to nodes as they become available. Fig#2 shows Simplified view of MapReduce

MapReduce Work Flow



Fig#2: Simplified view of MapReduce

IV. PIG

Apache Pig is a platform for analyzing Big-Data that consists of a high-level language for expressing data analysis programs, along with infrastructure for evaluating these programs. Pig's architecture consists of a compiler which produces sequences of Map-Reduce programs, for already existing large-scale parallel implementation. In Pig data workers can write complex data transformations independent of Java knowledge. Pig's simple SQL-like scripting language is called Pig Latin, and is easily understood by developers who are familiar with scripting languages and SQL.

Pig is complete, therefore all required data manipulations can be done in Apache Hadoop with Pig. Using the User Defined Functions (UDF) that are available in Pig, it can invoke code in many languages like JRuby, Jython and Java. Pig scripts can be embedded in other languages. The advantage of Pig is that it can be used as a key to build larger and more complex applications that handle real business problems. Pig works with data from many sources, and stores the results into the HDFS. Important features of Pig are:

• Ease of programming. It is easy to achieve parallel execution of data analysis tasks. Complex tasks consisting of multiple co-related data transformations are explicitly encoded as data flow sequences, thus is easy to write, understand, and maintain.

• Optimization opportunities. The method of encoding the tasks enables the system to optimize their execution, and hence the user can focus on semantics rather than efficiency.

• Extensibility. Users can create their own customised functions to do special-purpose processing.

V. HIVE

The Apache Hive data warehouse infrastructure built on top of Hadoop facilitates querying and managing large datasets residing in distributed storage. Hive provides a mechanism to project structure onto this data and query the data using a SQL-like language called HiveQL. It also allows traditional map/reduce programmers to plug in their custom mappers and reducers when it is inconvenient or inefficient to express this logic in HiveQL. Hadoop was built to organize and store massive amounts of data of various shapes, sizes and formats. Because of its "schema on read" architecture, a Hadoop cluster is a perfect reservoir of heterogeneous data—structured and unstructured—from a multitude of sources. Data analysts use Hive to explore structure and analyze that data, then turn it into business insight. Hive is similar to traditional database code with SQL access. However, since Hive is based on Hadoop and MapReduce operations, there are several key differences. The first is that Hadoop is intended for long sequential scans, and since it is based on Hadoop, queries may have a very high latency (many minutes). Therefore hive cannot be used for applications which need fast response times. Finally, Hive is read-based and therefore not appropriate for application that requires a high percentage of write operations.

How Hive Works

The tables in Hive are similar to tables in a relational database, and data units are organized in taxonomy from larger to more granular units. Databases consist of tables, which are made up of partitions. Data can be accessed via a simple query language and Hive supports overwriting or appending data. Within a particular database, data in the tables is serialized and each table has a corresponding Hadoop Distributed File System (HDFS) directory. Each table can be sub-divided into partitions that determine how data is distributed within sub-directories of the table directory. Data within partitions can be further broken down into buckets. Hive supports all the common primitive data formats such as BIGINT, BINARY, BOOLEAN, CHAR, DECIMAL, DOUBLE, FLOAT, INT, SMALLINT, STRING, TIMESTAMP, and TINYINT. In addition, analysts can combine primitive data types to form complex data types, such as structs, maps and arrays[5].

VI. STOCK DATA ANALYSIS

The stock data is collected by NSE and NYSE .The collected data is completely unstructured hence it becomes difficulty to analyze the data. This data is transformed into understandable format using java programs. This huge amount of data is the loaded onto a Hadoop distributed file system. This file system consists of number of clusters .Once the data is loaded onto the hdfs file system the data is balanced across the clusters. These hdfs system are fault tolerant, as there exists replication of data among the clusters and if any one of the node fails the whole setup doesn't crash, hdfs obtains the replicated file system. Thus once the semi structured data is uploaded on to hdfs file system the data can be used by tools for analysis. We are using Apache hive and apache pig to process the data, which uses mappers and reducers to process the data.

VII. TECHNICAL INDICATORS

A. *Change-*
Change is the difference between closing price and opening price of the day. And it can be calculated by the following formula,
Change = (((close-open)/open) * 100)

B. *Covariance-*
Covariance measures how two variables move together. It measures whether the two move in the same direction or in opposite directions. This can be represented by the following equation:

$$\text{Covariance} = \frac{\sum(\text{Return}_{ABC} - \text{Average}_{ABC}) * (\text{Return}_{XYZ} - \text{Average}_{XYZ})}{(\text{Sample Size}) - 1}$$

C. *Moving Average*
Simple moving average (SMA), is calculated by taking the arithmetic mean of a given set of values. For example, to calculate a basic 5-day moving average you would add up the closing prices from the past 5 days and then divide the result by 5.
Ex. Daily Closing price- 11,12,13,14,15,16,17
To Find MA of day-
1st day- (11+12+13+14+15)/5=13
2nd day- (12+13+14+15+16)/5=14
3rd day- (13+14+15+16+17)/5=15 & so on.

D. *Adjusted Closing Price*
It accounts for all corporate actions such as stock splits, dividends/distributions and rights offerings. It is calculated in following three ways:

i. For example, let's assume that the closing price for one share of XYZ Corp. is $20 on Thursday. After close on Thursday, XYZ Corp. announces a dividend distribution of $1.50 per share. The adjusted closing price for the stock would then be $18.50 ($20-$1.50).

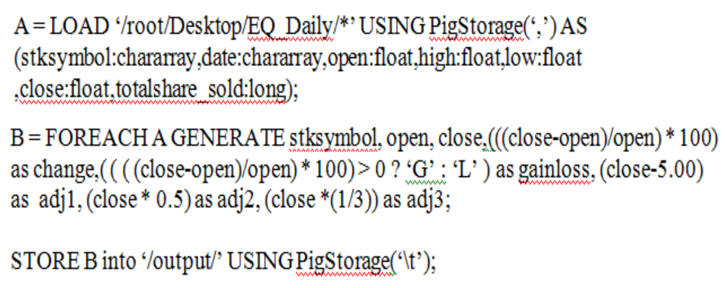
ii. If XYZ Corp. announces a 2:1 stock dividend instead of a cash dividend, the adjusted closing price calculation will change. In this case, the adjusted closing price calculation will be $20*(1/(2+1)). This will give you a price of $6.67.

iii. If XYZ Corp. announces a 2:1 stock split, This time the calculation will be $20*(1/(1x2)),resulting in an adjusted closing price of $10.

E. *On Balance Volume (OBV)-*
1) It is technical analysis indicator used to relate price & volume in stock market.
2) If OBV constantly increases means there are trends.
3) If OBV constantly decreases means there are downward trends.
4) If today's Close > yesterday's close then OBV= OBV (yesterday) + Volume (today)
5) If today's Close< yesterday's close then OBV=OBV (yesterday)-Volume (today).

We use the following sample Pig query on the stock dataset to analyze the data for change , gain-loss, adjusted closing price[4].

Queries in Pig

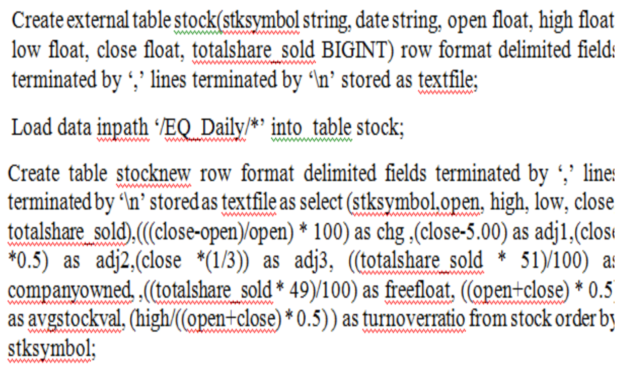


```
A = LOAD '/root/Desktop/EQ_Daily/*' USING PigStorage(',') AS
(stksymbol:chararray,date:chararray,open:float,high:float,low:float
,close:float,totalshare_sold:long);

B = FOREACH A GENERATE stksymbol, open, close,(((close-open)/open) * 100)
as change,( ( ( (close-open)/open) * 100) > 0 ? 'G' : 'L' ) as gainloss, (close-5.00)
as adj1, (close * 0.5) as adj2, (close *(1/3)) as adj3;

STORE B into '/output/' USING PigStorage('\t');
```

Fig#3:sample query using Pig.

This query first loads the data file required for analysing and then generates the specified schema and calculates the given parameters.Futher we store the result into a file over hdfs.Similarly we demonstrate the queries on hive[4]

Queries in Hive

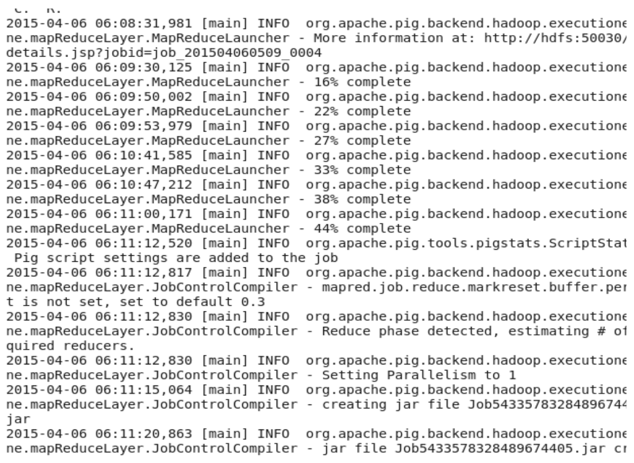


```
Create external table stock(stksymbol string, date string, open float, high float
low float, close float, totalshare_sold BIGINT) row format delimited fields
terminated by ',' lines terminated by '\n' stored as textfile;

Load data inpath '/EQ_Daily/*' into table stock;

Create table stocknew row format delimited fields terminated by ',' lines
terminated by '\n' stored as textfile as select (stksymbol,open, high, low, close
totalshare_sold),(((close-open)/open) * 100) as chg ,(close-5.00) as adj1,(close
*0.5) as adj2,(close *(1/3)) as adj3, ((totalshare_sold * 51)/100) as
companyowned, ,((totalshare_sold * 49)/100) as freefloat, ((open+close) * 0.5
as avgstockval, (high/((open+close) * 0.5)) as turnoverratio from stock order by
stksymbol;
```

Fig#4:sample queries using hive.

Here we first create the external table providing the schema for the huge unstructured data Then we Load data in path provided by the query and thus we perform SELECT query on the data and display results on the terminal .The Fig#4 shows the snapshot of working of query

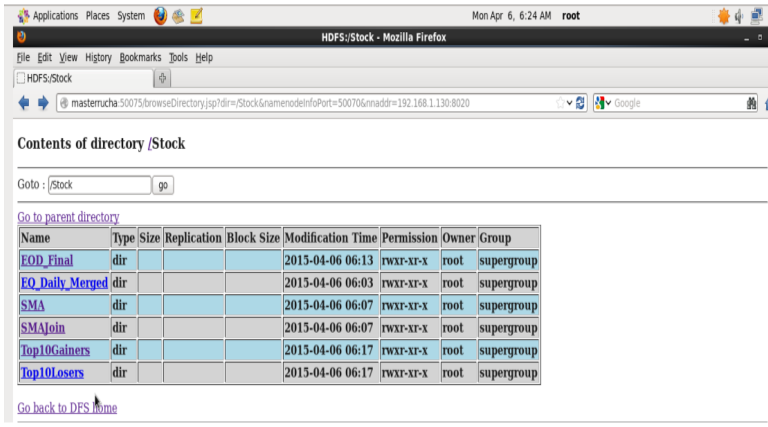
Snapshot of Query Execution



```
2015-04-06 06:08:31,981 [main] INFO  org.apache.pig.backend.hadoop.executione
ne.mapReduceLayer.MapReduceLauncher - More information at: http://hdfs:50030/
details.jsp?jobid=job_201504060509_0004
2015-04-06 06:09:30,125 [main] INFO  org.apache.pig.backend.hadoop.executione
ne.mapReduceLayer.MapReduceLauncher - 16% complete
2015-04-06 06:09:50,002 [main] INFO  org.apache.pig.backend.hadoop.executione
ne.mapReduceLayer.MapReduceLauncher - 22% complete
2015-04-06 06:09:53,979 [main] INFO  org.apache.pig.backend.hadoop.executione
ne.mapReduceLayer.MapReduceLauncher - 27% complete
2015-04-06 06:10:41,585 [main] INFO  org.apache.pig.backend.hadoop.executione
ne.mapReduceLayer.MapReduceLauncher - 33% complete
2015-04-06 06:10:47,212 [main] INFO  org.apache.pig.backend.hadoop.executione
ne.mapReduceLayer.MapReduceLauncher - 38% complete
2015-04-06 06:11:00,171 [main] INFO  org.apache.pig.backend.hadoop.executione
ne.mapReduceLayer.MapReduceLauncher - 44% complete
2015-04-06 06:11:12,520 [main] INFO  org.apache.pig.tools.pigstats.ScriptStat
 Pig script settings are added to the job
2015-04-06 06:11:12,817 [main] INFO  org.apache.pig.backend.hadoop.executione
ne.mapReduceLayer.JobControlCompiler - mapred.job.reduce.markreset.buffer.per
t is not set, set to default 0.3
2015-04-06 06:11:12,830 [main] INFO  org.apache.pig.backend.hadoop.executione
ne.mapReduceLayer.JobControlCompiler - Reduce phase detected, estimating # of
quired reducers.
2015-04-06 06:11:12,830 [main] INFO  org.apache.pig.backend.hadoop.executione
ne.mapReduceLayer.JobControlCompiler - Setting Parallelism to 1
2015-04-06 06:11:15,064 [main] INFO  org.apache.pig.backend.hadoop.executione
ne.mapReduceLayer.JobControlCompiler - creating jar file Job54335783284896744
jar
2015-04-06 06:11:20,863 [main] INFO  org.apache.pig.backend.hadoop.executione
ne.mapReduceLayer.JobControlCompiler - jar file Job5433578328489674405.jar cr
```

Fig#5: Status of Pig query getting executed.

Snaphot of Contents of Direcoty



Applications Places System
Mon Apr 6, 6:24 AM root
HDFS:/Stock - Mozilla Firefox
File Edit View History Bookmarks Tools Help
HDFS:/Stock
masterrucha:50075/browseDirectory.jsp?dir=/Stock&namenodeInfoPort=50070&nnaddr=192.168.1.130:8020
Google

Contents of directory /Stock

Goto : /Stock go

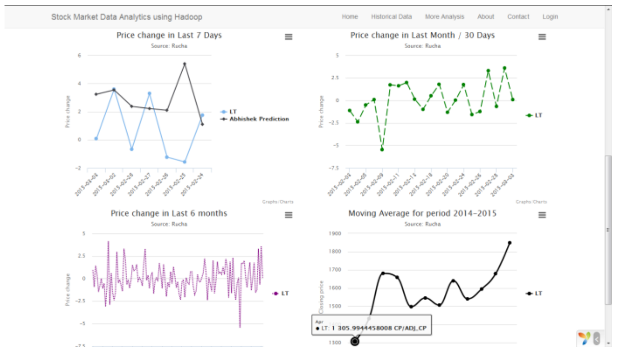
Go to parent directory

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
EOD_Final	dir				2015-04-06 06:13	rwxr-xr-x	root	supergroup
EQ_Daily_Merged	dir				2015-04-06 06:03	rwxr-xr-x	root	supergroup
SMA	dir				2015-04-06 06:07	rwxr-xr-x	root	supergroup
SMAJoin	dir				2015-04-06 06:07	rwxr-xr-x	root	supergroup
Top10Gainers	dir				2015-04-06 06:17	rwxr-xr-x	root	supergroup
Top10Losers	dir				2015-04-06 06:17	rwxr-xr-x	root	supergroup

Go back to DFS home

Fig#6: Snapshot of file system over the HDFS.

Using various such queries the visualization of data is done for various attributes of stock data and one such graph for price change over a period of last 7 days, last 30 days, last 6 months and moving average of the previous year is shown in Fig#7.

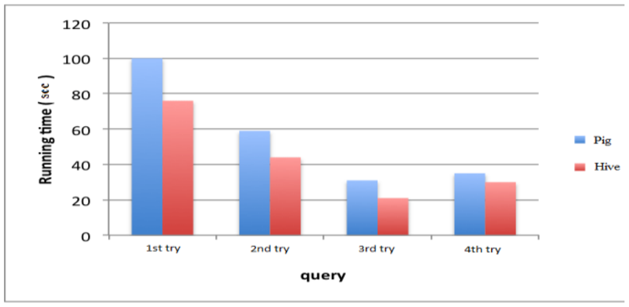


Fig#7: Graph for some calculated attributes.

VIII. CONCLUSION

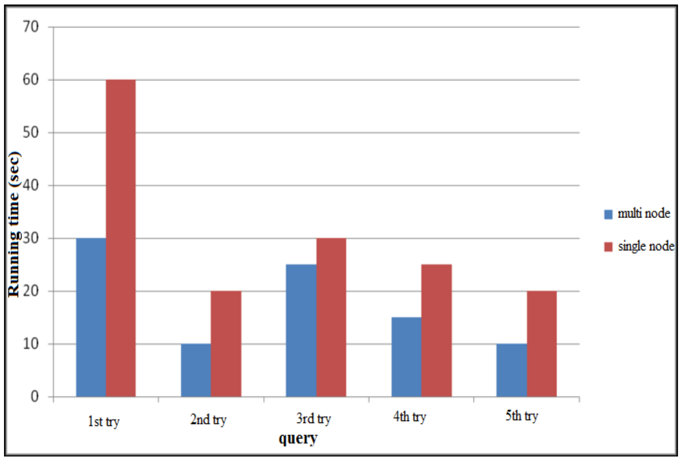
Our project focuses on doing the performance evaluation of the two tools in Hadoop viz. Pig and Hive. And the Fig#8(a) shows the results obtained between the performance of pig and hive queries and according to the analysis Hive proves to be faster and more efficient than Pig.

Comparision Graphs



Fig#8(a): Comparison of pig and hive

Second conclusion from our project is that we also perform the performance evaluation of the queries between pseudo distributed system and fully distributed system .Since in multimode the master node distributes the work to several other slaves that execute in parallel execution speed is much faster in multinode. The Fig#8(b) shows the comparison between pseudo distributed mode and multi node.



Fig#8(b): Comparison between Pseudo distributed node and multi node

REFERENCES

[1]. Teradata Magazine article, 2011

[2]. The McKinsey Global Institute, 2012

[3]. Wikipedia, 2014

[4] www.open-bigdata.com

[5] Hadoop -A Definitive Guide ,O'REILLY

[6] http://hadoop.apache.org/

[7] http://www.core-servlets.com/

[8]Hadoop in Action - Chuk Lam