Analysis of Big data through Hadoop Ecosystem Components like Flume, MapReduce, Pig and Hive

Dr. E. Laxmi Lydia[1], Dr. M.Ben Swarup[2]

[1]Associate Professor, Department of Computer Science and Engineering, Vignan's Institute Of Information Technology, Visakhapatnam, elaxmi2002@yahoo.com, Andhra Pradesh, India.
[2]Professor, Computer Science and Engineering, Vignan's Institute Of Information Technology, Visakhapatnam, bforben@gmail.com,Andhra Pradesh, India.

Abstract

In gigantic data world, Hadoop Distributed File System (HDFS) is amazingly understood. It gives a framework to securing data in a passed on circumstance besides has set of instruments to recuperate and plan. These data set using aide diminish thought. In this paper, a serious examination has been passed on to discuss that how colossal data examination can be performed on data set away on Hadoop scattered report system using Pig and Hive. Apache Pig and Hive are two endeavors which are layered on top of Hadoop, and give more lifted sum tongue to use Hadoop's MapReduce library. In this paper, as an issue of first significance, the crucial thoughts of MapReduce, Pig and Hive are displayed and their execution correlation.

Keywords: Hadoop Distributed File System, Pig, Hive, mapreduce, Framework

I. INTRODUCTION

We have entered an era of Big Data [1]. Huge information is for the most part accumulation of information sets so extensive and complex that it is exceptionally hard to handle them utilizing close by database administration devices. The principle challenges with Big databases incorporate creation, curation, stockpiling, sharing, inquiry, examination and perception. So to deal with these databases we require, "exceedingly parallel software's". As a matter of first importance, information is procured from diverse sources, for example, online networking, customary undertaking information or sensor information and so forth. Flume can be utilized to secure information from online networking, for example, twitter. At that point, this information can be composed utilizing conveyed document frameworks, for example, Google File System or Hadoop File System. These record frameworks are extremely proficient when number of peruses are high when contrasted with composes. Finally, information is dissected utilizing mapreducer with the goal that inquiries can be keep running on this information effectively and proficiently.Figure 1 showing the hadoop ecosystem. In the ecosystem of Hadoop, there have been several recent research projects exploiting sharing opportunities and eliminating unnecessary data movements, e.g. [2] [4] [6] [3].

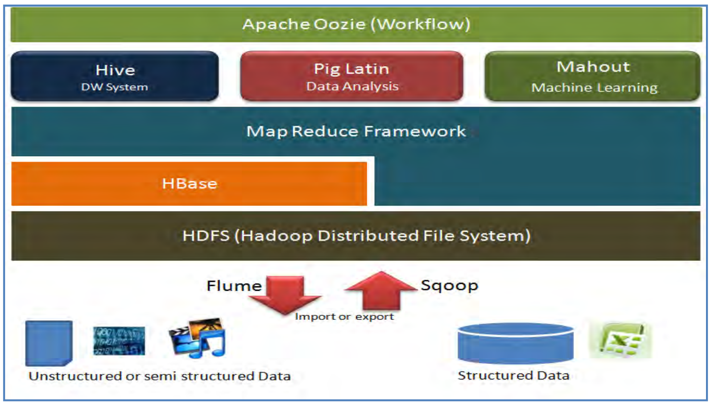


Figure1 Hadoop EcoSystem

II. ACQUIRE DATA

First of all, data has to be acquired from different sources. Main sources of data are:

- ✓ Traditional Organization data – it includes customer info from CRM systems, transactional ERP data or web store transactions and general ledger data.
- ✓ Machine generated or sensor data – it includes Call Detail Records, smart meters, weblogs, sensors, equipment logs and trading systems data.

- ✓ Social data – it includes customer feedback stream and micro blogging sites such as Twitter and social media platforms such as Facebook.

A. Flume

Data from web systems administration is generally gotten using flume. Flume is an open source programming undertaking which is made by cloudera to go about as an organization for gathering and moving enormous measure of data around a Hadoop bundle as data is conveyed or in no time. Crucial use case of flume is to gather log records from all machines in cluster to continue on them in a united store, for instance, HDFS. In it, we have to make data streams by building up chains of sensible center points and partner them to source and sinks. For example, if you have to move data from an apache access sign into HDFS then you have to make a source by tail access.log and use an astute center point to course this to a HDFS sink. By far most of flume game plans have three level diagram. The administrators level have flume masters assembled with wellsprings of data which is to be moved. Power level involve various gatherers each of which accumulate data coming in from distinctive authorities and forward it on to limit level which include archive system like HDFS or GFS.

A **Flume agent** is a **JVM** process which has 3 components -**Flume Source**, **Flume Channel** and **Flume Sink**- through which events propagate after initiated at an external source. Figure 2 demonstrating the working of flume.

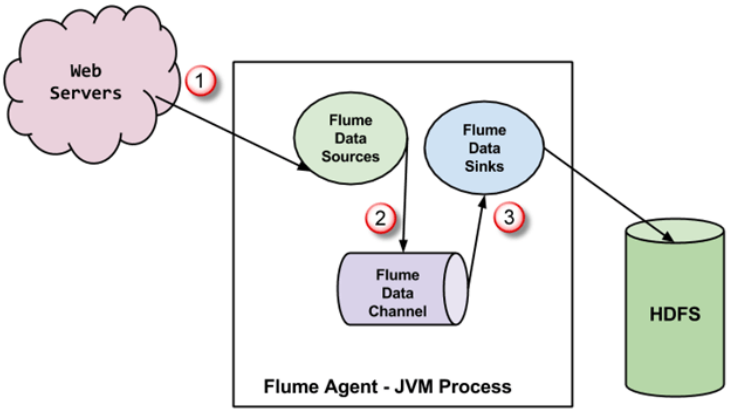


Figure 2 Working of Flume

1. In above graph, the occasions created by outer source (WebServer) are devoured by Flume Data Source. The outer source sends occasions to Flume source in a configuration that is perceived by the objective source.

2. Flume Source gets an occasion and stores it into one or more channels. The channel goes about as a store which keeps the occasion until it is devoured by the flume sink. This channel may utilize neighborhood document framework keeping in mind the end goal to store these occasions.

3. Flume sink expels the occasion from channel and stores it into an outer storehouse like e.g., HDFS. There could be different flume operators, in which case flume sink advances the occasion to the flume wellspring of next flume specialists in the stream.

III. Organize Data

After acquiring data, it has to be organize using a distributed file system. First of all, we have to break this data into fixed size chunks so that they can store and access easily. Mainly we use GFS and HDFS file systems.

A. Google File System

Google Inc. built up an appropriated record framework for their own particular use which was intended for proficient and solid acess to information utilizing extensive bunch of product equipment. It utilizes the methodology of "BigFiles", which are created by Larry Page and Sergey Brin. Here records are partitioned in fixedsize pieces of 64 MB. It has two sorts of hubs one expert hub and numerous chunkserver hubs.

Documents in altered size lumps are put away on chunkservers which are relegated a 64 bit name by expert at creation time. There are atleast 3 replication for each piece however it can be more. Expert hub doesn't have information pieces, it keeps the metadata about lumps, for example, their mark, their duplicate areas and their perusing or composing procedures. It additionally have the obligation to duplicate a piece when it's duplicates turn out to be under three. Figure 5 demonstrating the structural planning of GFS is taking after.

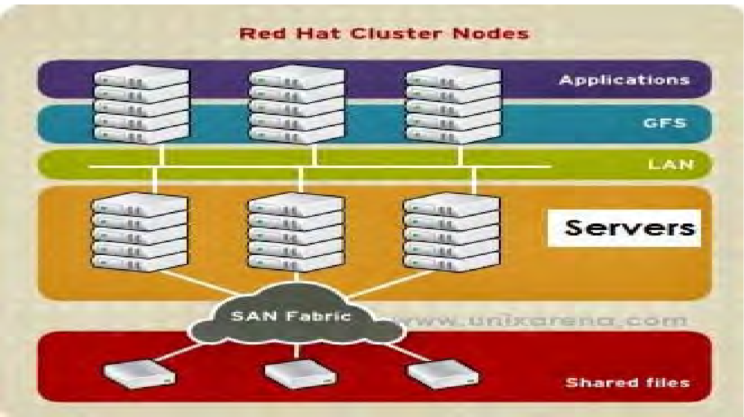


Figure 3 GFS

B. *Hadoop Distributed File System*

Hadoop dispersed document framework is a conveyed, adaptable and convenient record framework which is composed in java. All machines which bolster java can run it. In it, each group has a solitary namenode and numerous datandes. A datanode has numerous squares of same size aside from last piece which have distinctive size. It do correspondence utilizing TCP/IP layer yet customers utilizes RPC to speak with one another. Each document in HDFS has a size of 64 MB or numerous of 64 MB. Unwavering quality is because of replication of information. Atleast 3 duplicates of each datanode are available. Datanodes can correspond with one another to rebalance information or duplicate information or to keep high replication of information. HDFS has high accessibility by permitting namenode to be physically fizzled over to reinforcement if there should be an occurrence of disappointment. Presently a days, programmed failover is additionally creating. It additionally utilizes a secondry namenode which persistently takes the previews of essential namenode with the goal that it can be dynamic when disappointment of essential hub happens. Information mindfulness in the middle of tasktracker and jobtracker is favorable position. Jobtracker can plan mapreduce occupation to tasktracker productively because of this information mindfulness. Figure 4 demonstrating the HDFS.

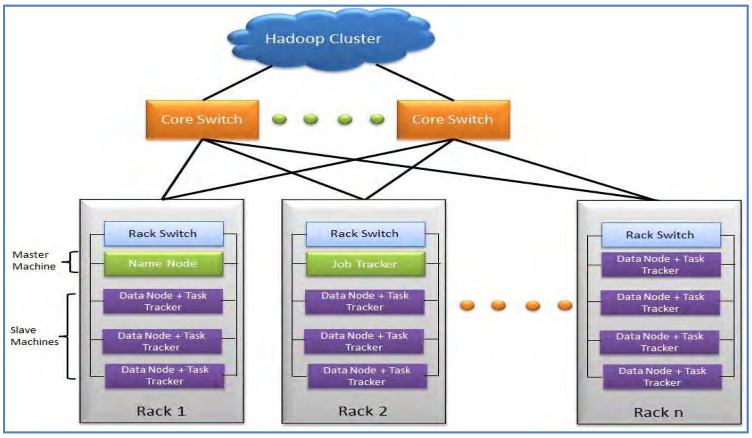


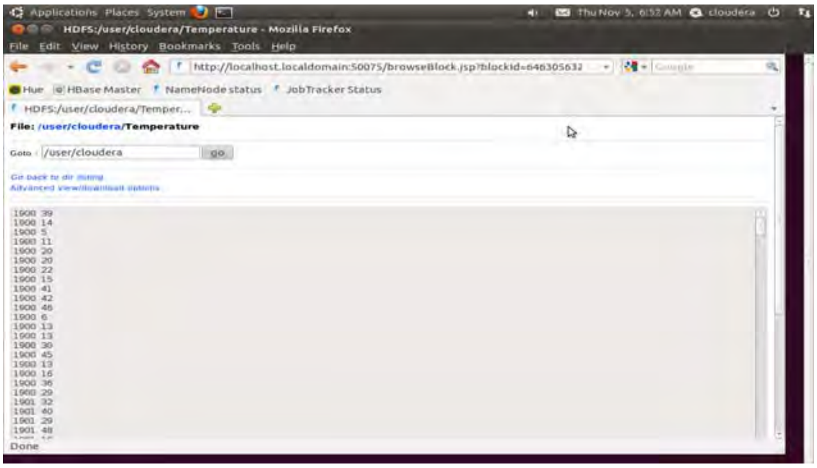
Figure 4 HDFS

IV. ANALYZE DATA

After organizing data, it has to be analyze to get fast and efficient results when a query is made. Mapreducer's are mainly used to analyze data. Mapreducer, Pig and Hive are very efficient for this purpose.

Setup for Analysis

An analysis is performed on a big database of 8 lakh records using Pig, Hive and MapReduce. For this purpose, we install Hadoop, Pig, Hive on cloudera. And analysis time is calculated for each [9]. Figure 5 showing the temperature datasets present in the HDFS. From the temperature datasets we are generating the maximum temperature from the year 1900 to 2014.


Applications Places System
Thu Nov 5, 6:52 AM cloudera
HDFS:/user/cloudera/Temperature - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost.localdomain:50075/browseBlock.jsp?blockId=646305632
Hue HBase Master NameNode status JobTracker Status
HDFS:/user/cloudera/Temper...

File: /user/cloudera/Temperature

Goto : /user/cloudera go

Go back to dir listing
Advanced view/download options

```
1900 39
1900 14
1900 5
1900 11
1900 20
1900 20
1900 22
1900 15
1900 41
1900 42
1900 46
1900 6
1900 13
1900 13
1900 30
1900 45
1900 13
1900 16
1900 36
1900 29
1901 32
1901 40
1901 29
1901 48
```

Done

Figure 5 Temperature dataset in the HDFS

A. MapReduce

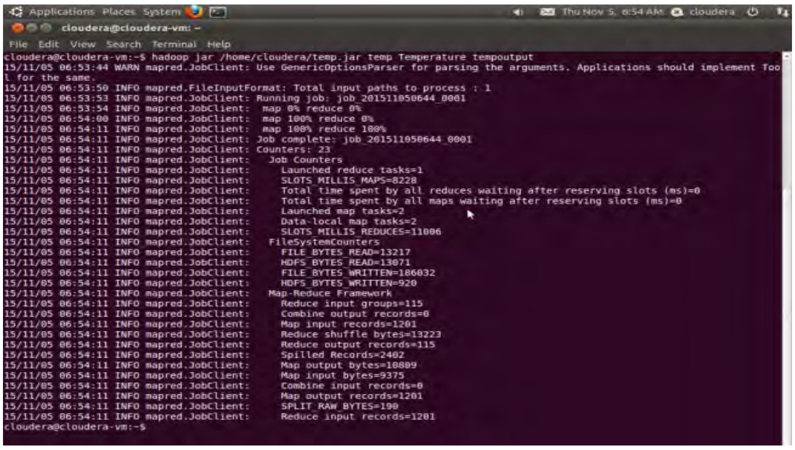
Hadoop MapReduce is a product structure for effortlessly composing applications which prepare tremendous measures of information (multi-terabyte information sets) in-parallel on substantial groups (a huge number of hubs) of item equipment in a dependable, flaw tolerant way.

A MapReduce work for the most part parts the info information set into free pieces which are prepared by the guide assignments in a totally parallel way. The structure sorts the yields of the maps, which are then data to the diminish assignments. Ordinarily both the information and the yield of the employment are put away in a document framework. The system deals with planning errands, observing them and re-executes the fizzled assignments.

Normally the register hubs and the stockpiling hubs are the same, that is, the MapReduce structure and the Hadoop Distributed File System are running on the same arrangement of hubs. This arrangement permits the system to adequately timetable undertakings on the hubs where information is as of now present, bringing about high total transfer speed over the group.

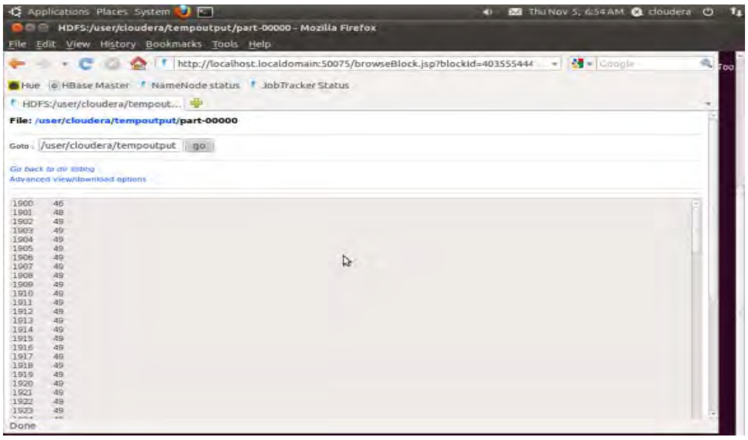
The MapReduce structure comprises of a solitary expert JobTracker and one slave TaskTracker per group hub. The expert is in charge of booking the occupations' part assignments on the slaves, checking them and re-executing the fizzled errands. The slaves execute the assignments as coordinated by the expert.

Negligibly, applications indicate the information/yield areas and supply guide and lessen capacities through executions of suitable interfaces and/or conceptual classes. These, and other employment parameters, contain the occupation design. The Hadoop work customer then presents the occupation (jug/executable and so on.) and setup to the JobTracker which then expect the obligation of disseminating the product/design to the slaves, booking errands and observing them, giving status and demonstrative data to the employment customer. Figure 6 exhibiting the execution of MapReduce and Figure 8 demonstrating the status report of jobtracker and 100% fruition of guide and decrease occupations.


Applications Places System
Thu Nov 5, 6:54 AM cloudera
cloudera@cloudera-vm: ~
File Edit View Search Terminal Help

```
cloudera@cloudera-vm:~$ hadoop jar /home/cloudera/temp.jar temp Temperature tempoutput
15/11/05 06:53:44 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Applications should implement Too
l for the same.
15/11/05 06:53:50 INFO mapred.FileInputFormat: Total input paths to process : 1
15/11/05 06:53:53 INFO mapred.JobClient: Running job: job_201511050644_0001
15/11/05 06:53:54 INFO mapred.JobClient:  map 0% reduce 0%
15/11/05 06:54:00 INFO mapred.JobClient:  map 100% reduce 0%
15/11/05 06:54:11 INFO mapred.JobClient:  map 100% reduce 100%
15/11/05 06:54:11 INFO mapred.JobClient: Job complete: job_201511050644_0001
15/11/05 06:54:11 INFO mapred.JobClient: Counters: 23
15/11/05 06:54:11 INFO mapred.JobClient:   Job Counters
15/11/05 06:54:11 INFO mapred.JobClient:     Launched reduce tasks=1
15/11/05 06:54:11 INFO mapred.JobClient:     SLOTS_MILLIS_MAPS=8228
15/11/05 06:54:11 INFO mapred.JobClient:     Total time spent by all reduces waiting after reserving slots (ms)=0
15/11/05 06:54:11 INFO mapred.JobClient:     Total time spent by all maps waiting after reserving slots (ms)=0
15/11/05 06:54:11 INFO mapred.JobClient:     Launched map tasks=2
15/11/05 06:54:11 INFO mapred.JobClient:     Data-local map tasks=2
15/11/05 06:54:11 INFO mapred.JobClient:     SLOTS_MILLIS_REDUCES=11006
15/11/05 06:54:11 INFO mapred.JobClient:   FileSystemCounters
15/11/05 06:54:11 INFO mapred.JobClient:     FILE_BYTES_READ=13217
15/11/05 06:54:11 INFO mapred.JobClient:     HDFS_BYTES_READ=13071
15/11/05 06:54:11 INFO mapred.JobClient:     FILE_BYTES_WRITTEN=186032
15/11/05 06:54:11 INFO mapred.JobClient:     HDFS_BYTES_WRITTEN=920
15/11/05 06:54:11 INFO mapred.JobClient:   Map-Reduce Framework
15/11/05 06:54:11 INFO mapred.JobClient:     Reduce input groups=115
15/11/05 06:54:11 INFO mapred.JobClient:     Combine output records=0
15/11/05 06:54:11 INFO mapred.JobClient:     Map input records=1201
15/11/05 06:54:11 INFO mapred.JobClient:     Reduce shuffle bytes=13223
15/11/05 06:54:11 INFO mapred.JobClient:     Reduce output records=115
15/11/05 06:54:11 INFO mapred.JobClient:     Spilled Records=2402
15/11/05 06:54:11 INFO mapred.JobClient:     Map output bytes=10809
15/11/05 06:54:11 INFO mapred.JobClient:     Map input bytes=9375
15/11/05 06:54:11 INFO mapred.JobClient:     Combine input records=0
15/11/05 06:54:11 INFO mapred.JobClient:     Map output records=1201
15/11/05 06:54:11 INFO mapred.JobClient:     SPLIT_RAW_BYTES=190
15/11/05 06:54:11 INFO mapred.JobClient:     Reduce input records=1201
cloudera@cloudera-vm:~$
```

Figure 6 Execution of MapReduce

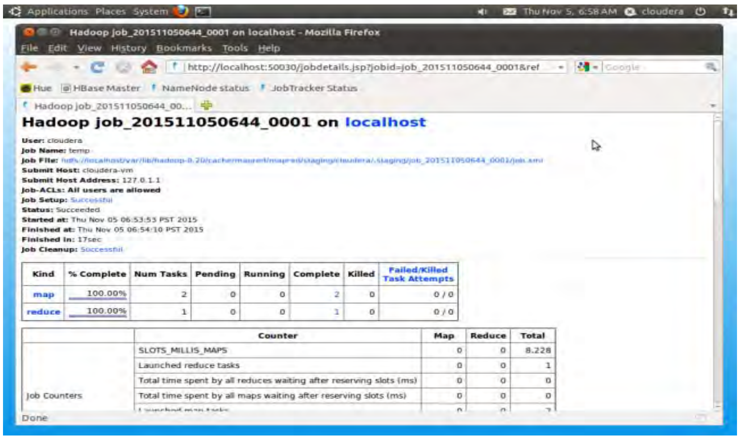

Applications Places System
Thu Nov 5, 6:54 AM cloudera
HDFS:/user/cloudera/tempoutput/part-00000 - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost.localdomain:50075/browseBlock.jsp?blockId=403555444
Google
Hue HBase Master NameNode status JobTracker Status
HDFS:/user/cloudera/tempout...
Go back to dir listing
Advanced view/download options
Done

File: /user/cloudera/tempoutput/part-00000

Goto : /user/cloudera/tempoutput go

```
1900    46
1901    48
1902    49
1903    49
1904    49
1905    49
1906    49
1907    49
1908    49
1909    49
1910    49
1911    49
1912    49
1913    49
1914    49
1915    49
1916    49
1917    49
1918    49
1919    49
1920    49
1921    49
1922    49
1923    49
```

Figure 7 output screen, showing the maximum temperature from the year 1900 to 2014


Applications Places System
Thu Nov 5, 6:58 AM cloudera
Hadoop job_201511050644_0001 on localhost - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:50030/jobdetails.jsp?jobid=job_201511050644_0001&ref
Google
Hue HBase Master NameNode status JobTracker Status
Hadoop job_201511050644_00...
Done

Hadoop job_201511050644_0001 on localhost

User: cloudera
Job Name: temp
Job File: hdfs://localhost/var/lib/hadoop-0.20/cache/mapred/mapred/staging/cloudera/.staging/job_201511050644_0001/job.xml
Submit Host: cloudera-vm
Submit Host Address: 127.0.1.1
Job-ACLs: All users are allowed
Job Setup: Successful
Status: Succeeded
Started at: Thu Nov 05 06:53:53 PST 2015
Finished at: Thu Nov 05 06:54:10 PST 2015
Finished in: 17sec
Job Cleanup: Successful

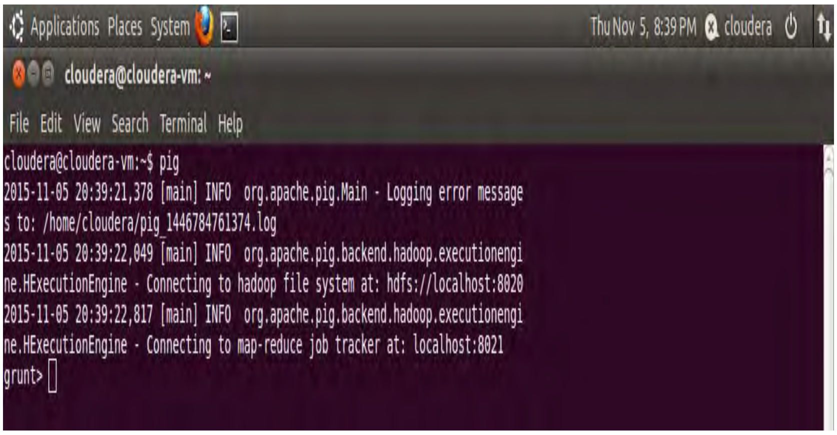
Kind	% Complete	Num Tasks	Pending	Running	Complete	Killed	Failed/Killed Task Attempts
map	100.00%	2	0	0	2	0	0 / 0
reduce	100.00%	1	0	0	1	0	0 / 0

	Counter	Map	Reduce	Total
Job Counters	SLOTS_MILLIS_MAPS	0	0	8,228
	Launched reduce tasks	0	0	1
	Total time spent by all reduces waiting after reserving slots (ms)	0	0	0
	Total time spent by all maps waiting after reserving slots (ms)	0	0	0

Figure 8 showing the status of jobtracker and 100% completion of map and reduce jobs.

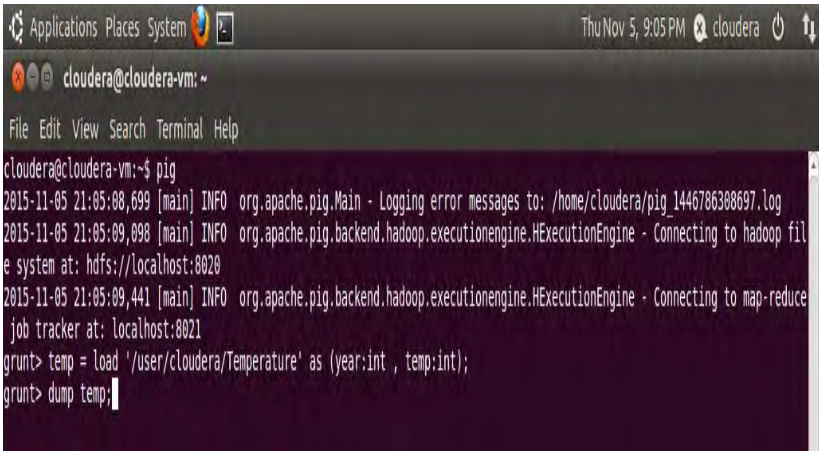
B. *Pig*

Pig was initially developed at Yahoo Research around 2006 but moved into the Apache Software Foundation in 2007. Pig consists of a language and an execution environment. Pig's language, called as PigLatin[6]. Pig have a language and an execution environment. Piglatin which is a dataflow language is used by Pig. Piglatin is a type of language in which you program by connecting things together. Pig can handle complex data structure, even those who have levels of nesting. It has two types of execution environment local and distributed environment. Local environment is used for testing when distributed environment cannot be deployed. PigLatin program is collection of statements. A statement can be a operation or command. Here is a program in PigLatin to analyze a database. Figure 9 showing the pig's grunt shell and Figure 10 showing the Loading of temperature datasets to temp relation to perform the temperature analysis with less time than by writing map reduce code.


Applications Places System
Thu Nov 5, 8:39 PM cloudera
cloudera@cloudera-vm: ~
File Edit View Search Terminal Help

```
cloudera@cloudera-vm:~$ pig
2015-11-05 20:39:21,378 [main] INFO  org.apache.pig.Main - Logging error message
s to: /home/cloudera/pig_1446784761374.log
2015-11-05 20:39:22,049 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.HExecutionEngine - Connecting to hadoop file system at: hdfs://localhost:8020
2015-11-05 20:39:22,817 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.HExecutionEngine - Connecting to map-reduce job tracker at: localhost:8021
grunt> 
```

Figure 9 pig's grunt shell


Applications Places System
Thu Nov 5, 9:05 PM cloudera
cloudera@cloudera-vm: ~
File Edit View Search Terminal Help

```
cloudera@cloudera-vm:~$ pig
2015-11-05 21:05:08,699 [main] INFO  org.apache.pig.Main - Logging error messages to: /home/cloudera/pig_1446786308697.log
2015-11-05 21:05:09,098 [main] INFO  org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to hadoop fil
e system at: hdfs://localhost:8020
2015-11-05 21:05:09,441 [main] INFO  org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to map-reduce
 job tracker at: localhost:8021
grunt> temp = load '/user/cloudera/Temperature' as (year:int , temp:int);
grunt> dump temp;
```

Figure 10 Loading of temperature datasets to temp relation

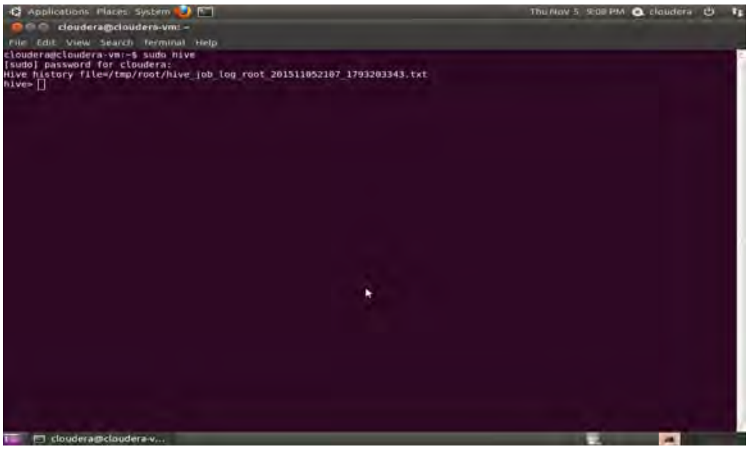
C. Hive

Apache Hive is a data warehouse system for Apache Hadoop [7].Hive is a technology which is developed by Facebook and which turns Hadoop into a datawarehouse complete with an extension of sql for querying. HiveQL which is a declarative language is used by Hive. In piglatin, we have to describe the dataflow but in Hive results must be describe. Hive itself find out a dataflow to get those results. Hive must have a schema but it can be more than one.

Hive must be configured before use. It can be configured in three different ways:

- ✓ By editing a file hive-site.xml,
- ✓ By hiveconf option in Hive command shell
- ✓ By using set command.

Here we have a database with more than eight lakh records which is analyzed by using Hive to get an temperature.


Applications Places System
Thu Nov 5 9:08 PM cloudera
cloudera@cloudera-vm: ~
File Edit View Search Terminal Help
cloudera@cloudera-v...

```
cloudera@cloudera-vm:~$ sudo hive
[sudo] password for cloudera:
Hive history file=/tmp/root/hive_job_log_root_201511052107_1793203343.txt
hive> 
```

Figure 11 Analysis using Hive

Analysis on Hive can be done by the Hive Query Language (HQL) commands, few the commands are demonstrated below:

A. Creating Database
   i. *Create database temp;*
   ii. *Use temp;*
B. Create table for storing temp records
   i. *Create table temps(year INT, year INT)*
C. Load the data into the table
   i. *LOAD DATA LOCAL INPATH '/home/cloudera/hive/data/Temperature.csv'*
   ii. *OVERWRITE INTO TABLE temps;*
D. Describing metadata or schema of the table
   i. *Describe temps*
E. Selecting data

*Select * from temps;*

To perform the analysis on temperature datasets , we used the above hive commands, we also used hive along with pig and mapreduce coding, using hive we performed the analysis by using above HIVE Query Language commands. Therefore the research paper, demonstrates that, to perform the quick analysis HIVE and PIG can be used than mapreduce, where we need to write lengthy coding, which is absent in the hive and pig.

V. COMPARISON

Keeping in mind the end goal to arrive at a decision about the useful correlation of Apache Map Reduce, pig and Hive, we performed a near examination utilizing these systems on a dataset that permits us to perform analysis basing on following metrics:

A. Performance
B. Development time

Map Reduce is a inner component of hadoop, other Pig and hive are hadoop eco systems it means run on the top of hadoop. The purpose of both mapreduce, pig and hive purpose is process the vast amount of data in different manner.

Mapreduce: apache implemented it. highly recommendable to process entire data, it's time consume and required program skills like java (highly recommendable), pyghon, ruby and other programming languages. total data aggregate and sort by using mapper and reducer functions. Hadoop use it by default.

Hive: Facebook implemented it. most of the analysts especially bigdata analysts use this tool to analyze the data especially structure data. Backend this hive tool use mapreduce to be processed. Internally Hive use special language called HQL, It's subset of SQL language. Who is wellever in SQL, they can goes with Hive. It's highly recommended to the Datawarehouse oriented projects. Much difficult to process un structured especially schema-less data.

Pig: Pig is a scripting language, implemented by Yahoo. The main difference between pig and Hive is pig can process any type of data, either structured or unstructured data. It means it's highly recommendable for streaming data like satellite generated data, live events, schema-less data etc. Pig first load the data later programmer write a program depends on data to make it structured. Who is expert in programming languages they will choose this Hadoop ecosystems.

Table 1 Temperature dataset

Year:	int
temperature	int

Dataset Description

The Data Set includes temperature datasets size of 8 MB collected over the years, and includes year and other temperature records. A sample of the data records is shown as below: The data record is demonstrated in the table1:

Sample Record:

1900 39

1900 14

1900 5

1901 48

1901 16

1901 11

1901 21

1901 6

1901 22

1908 26

1909 37

1909 38

1909 29

1909 25

1933 20

1933 20

Result analysis of Map Reduce, Hive and Pig with respect the two metrics i.e. Performance and development time

- Map-Reduce: Has better performance than pig or hive but requires more development time.
- PIg: Less development time but poor performance when compared to map-reduce.
- Hve: SQL type language with some good features like partitioning and bucketing to improve performance, hive enforces schema on read.

To gain a varied analysis, we considered 64MB, 3.13 MB with a single node and 3.13MB with 10 nodes and monitored the performance in terms of the time taken for clustering as per our requirements using K-Means algorithm. The machines used had a configuration as follows:

- 4GB RAM
- Linux Ubuntu
- 500 GB Hard Drive

Performance: Map Reduce, Hive and Pig

The performance graph which is tested on cluster which consists of 10 nodes, shown in figure shows that Map Reduce has better performance than pig or hive.

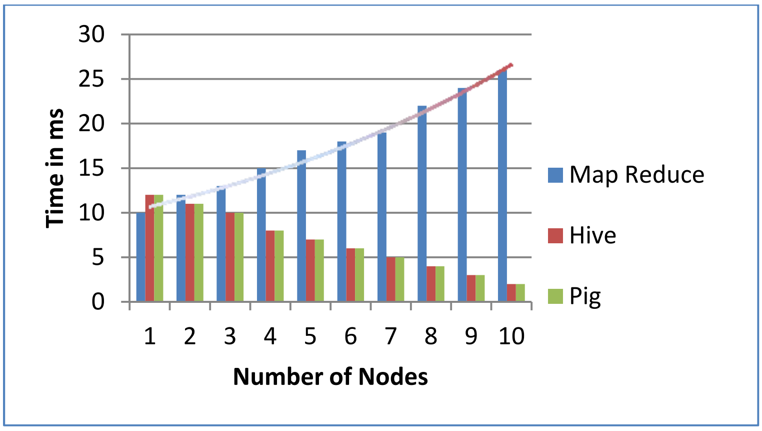


Figure showing the High performance of Map Reduce than Hive and Pig

Development time: Map Reduce, Hive and Pig

The Development time graph which is tested on 10 nodes of cluster proved that the development time of map reduce of more as there involves large number of coding where the hive and pig shows that development time is very less as hive involves simple HQL and pig invokes short script.

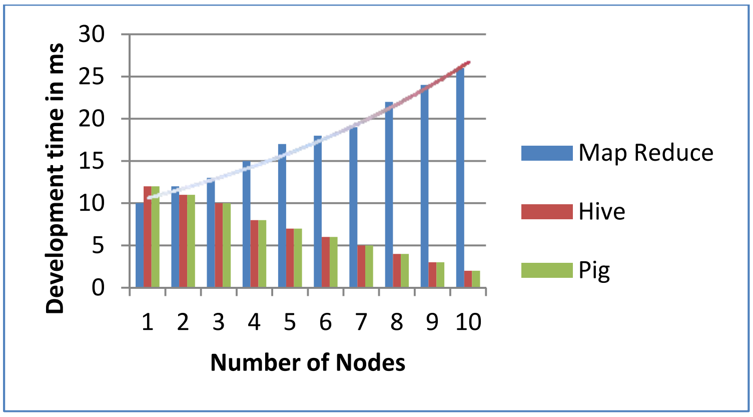


Figure showing the Development time of Map Reduce Hive and Pig

VI. CONCLUSION

It is impractical to handle Big information utilizing customary database administration frameworks like social databases. So we utilize some exceedingly parallel programming to handle huge databases. A few segments are likewise used to handle them. Firstly we need to obtain information from diverse sources which can be effectively done by utilizing segments like Flume. Flume can specifically get tweets from sites like twitter and store them in Bid databases. At that point we can compose them utilizing dispersed document framework like

GFS and HDFS. Finally they can be dissected for quicker get to and inquiry reaction. After investigation get to and question reaction takes lesser time and exertion on these huge databases. Pig, Hive and MapReduce like segments can do examination in brief time. Hive can break down a database of more than 8 lakh records in only 34 seconds. So every one of these parts make it conceivable to handle and to utilize Big database in a simple and proficient way

REFERENCES

[1] https://hadoop.apache.org/.

[2] R. Lee, T. Luo, Y. Huai, F. Wang, Y. He, and X. Zhang. YSmart: Yet Another SQL-to-MapReduce Translator. In ICDCS, 2011.

[3] H. Lim, H. Herodotou, and S. Babu. Stubby: A Transformation-based Optimizer for Mapreduce Workflows. In VLDB, 2012.

[4] T. Nykiel, M. Potamias, C. Mishra, G. Kollios, and N. Koudas. MRShare: Sharing Across Multiple Queries in Mapreduce. In VLDB, 2010.

[5] X. Wang, C. Olston, A. D. Sarma, and R. Burns. CoScan: Cooperative Scan Sharing in the Cloud. In SoCC, 2011.

[6] 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” By Facebook Data Infrastructure Team

[7] Apache Hadoop. Available at http://hadoop.apache.org.

[8] OSDI 2006.

[9] Zhifeng YANG, Qichen TU, Kai FAN, Lei ZHU, Rishan CHEN, BoPENG, “Performance Gain with Variable Chunk Size in GFS-like File Systems”, Journal of Computational Information Systems4:3 pp-1077-1084, 2008.

[10] Sam Madden , “From Databases to Big Data”, IEEE Computer Society , 2012.

[11] Sanjeev Dhawan & Sanjay Rathee, “Big Data Analytics using Hadoop Components like Pig and Hive,” American International Journal of Research in Science, Technology, Engineering & Mathematics, pp:1-5,2013.