ENHANCE PERFORMANCE OF SQL-ON-HADOOP SYSTEM BY USING VARIOUS FILE FORMATS

Ramandeep Kaur
Mtech(Research Scholar)
CSE LNCT Bhopal

Ratnesh Kumar Dubey
Assistant Professor
CSE LNCT Bhopal

Dr.Sadhna Mishra
Professor &Head
CSE LNCT Bhopal

Abstract: Today the industry and companies are analyzing petabytes of data every day , But the data is still growing and analysis with the help of traditional RDBMS solutions becomes expensive and time taking. Hadoop is a popular solution for handing large amount of data which store huge amount of data on a cluster of commodity hardware machines and process these data by using mapreduce which process the data paralley. But HAdoop mapreduce support java programming language and for data analyst its very time taking taking process to write large program for process the data. In these paper we proposed a hive(SQL on Hadoop) which is a analytical tools and runs on top of the hadoop. In these we porposed various file format such as textfile, sequence file, RC(Record Columnar) and ORC (Optimized Record Columnar) files through which we can enhance the performance of SQL on Hadoop system.

Keywords-- **Hadoop, hive, file format, textfile, sequence file, RC file, ORC file, performance optimization, SQL-on-Hadoop.**

I. INTRODUCTION

Hadoop Mapreduce might be a coding system programming model to technique big datasets on clusters in associate economical manner. A MapReduce task divides the pc file into style of chunks that square measure appointed to whole totally different DataNodes and square measure processed in parallel manner. the pc file is given as input into map perform that ends in another set of pairs. This another set is commonly thought-about as intermediate strive. The output of the upper than map perform is sorted automatically [8]. The output of the map perform consists whole totally different values associated with the corresponding key. And once the results of the map perform is sorted, it's fed into shrink perform. The task of shrink perform is to cut back all pairs that's obtained from the map perform. The MapReduce framework splits the information across several DataNodes that schedules the execution of task in conjunction with cluster-cluster communications. If any DataNode failure happens it's detected by NameNode that endlessly monitors the DataNodes. There comes the role of copy where the failure node info is given to various active DataNodes and additionally the execution continues in parallel manner.

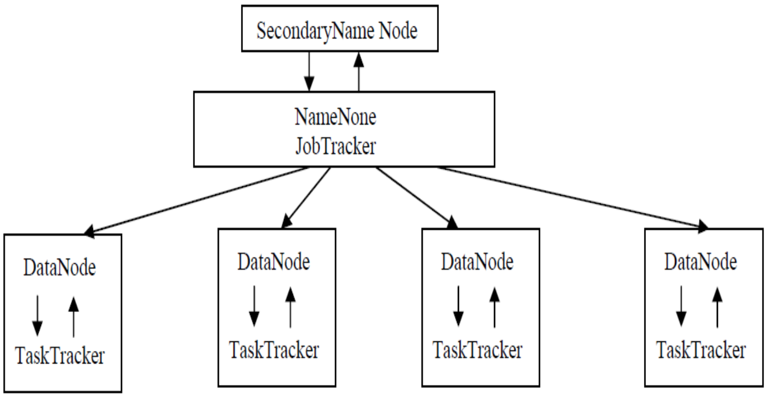


Figure-1. Hadoop Architecture

HIVE

Facebook created Hive for analyzing large datasets. It is a most widely adopted data warehousing application which can provide the Relational model and SQL interface [7]. Hive infrastructure runs on the top of Hadoop. It mainly helps in providing summary of the data, query and analysis of the unstructured data.

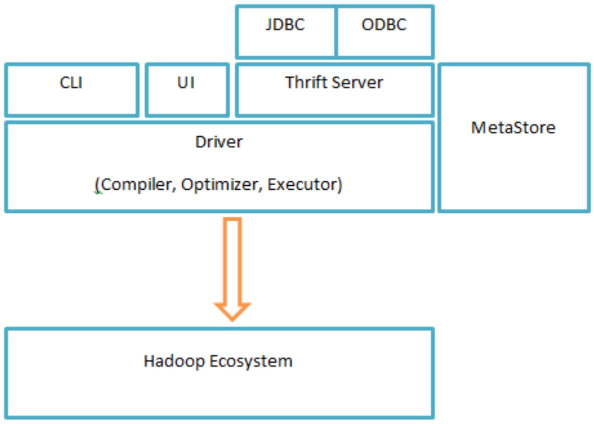


Figure-2. Hive Architecture

II. LITERATURE REVIEW

In [1] Hadoop is now the de facto standard for storingand processing big data, not only for unstructured data butalso for some structured data. As a result, providing SQLanalysis functionality to the big data resided in HDFS becomesmore and more important. Hive is a pioneer system thatsupports SQL-like analysis to the data in HDFS. However, theperformance of the early-version of Hive is not satisfactory.This leads to the quick emergence of dozens of SQL-on-Hadoop systems that try to support interactive SQL queryprocessing to the data stored in HDFS [10]. This paper firstly gives a brief technical review on recent efforts of SQL on-Hadoopsystems. Then we test and compare the performance of threerepresentative SQL-on-Hadoop systems, based on the TPCHbenchmark. According to the results, we show that suchsystems can benefit more from applications of many parallel
query processing techniques that have been widely studied inthe traditional massively parallel processing databases.

Hadoop is Java primarily based altogether programming framework for distributed storage and methodology of big information sets on object hardware [5]. It's developed by Apache package Foundation as open give framework. Hadoop primarily has 2 main elements. initial one is Hadoop Distributed system (HDFS) for distributed storage and half is MapReduce for distributed methodology. HDFS may well be a system that builds on this system. It's Java-based sub project of Apache Hadoop. HDFS provides ascendable and reliable information storage on commandity hardware. A master/slave vogue is used by HDFS. Throughout this vogue, HDFS incorporates one NameNode and quite one DataNodes. The NameNode manages the system and stores the info. It acts kind of a file manager on HDFS. as a results of all files and directories ar delineate on the NameNode. DataNodes stores the a district of data [6]. A MapReduce job typically splits the computer file set into freelance blocks that ar processed by the map tasks throughout a completely parallel manner. Commencement is mapping of data set in MapReduce vogue. Input so the output of the work ar hold on throughout a file-system. The MapReduce framework consists of 2 technique that areJobTracker and TaskTracker. The JobTracker manages the resources that areTaskTracker. The TaskTracker may well be a way node at intervals the cluster [9].

Giant information analytics has attracted intense interest from all domain and trade recently for it's conceive to extract information, knowledge. massive information analytical capabilities victimization cloud delivery models might ease adoption for several trade, and most vital thinking to price saving, it might alter helpful insights which will providing them with totally fully totally different types of competitive advantage. several corporations to supply on-line massive information analytical tools kind of the absolute best most corporations like Amazon massive information Analytics Platform ,HIVE [2] internet primarily based Interface, SAP massive information Analytics, IBM InfoSphereBigInsights, TERADATA massive information Analytics, 1010data massive information Platform, Cloudera massive information answer etc.Those firms analyzeBrobdingnagian amount of information with facilitate of varied sort of tools and in addition supply easy or easy programme for analyzing information. During this paper the author offers kind of Brobdingnagian information analytical tools through that we'll merely analyze the large and complex datasets and therefore the author also provides a comparison between these bigdata analytical tools.

III OBSERVATION

The information that we have a tendency to generating was growing in no time - as associate example we grew from a 15TB information set in 2007 to a 700TB data set these days. The infrastructure at that point was thus inadequate that some daily processing jobs were taking quite daily to process and also the scenario was simply obtaining worse with each passing day. we have a tendency to had associate imperative would like for infrastructure that would scale in conjunction with our information [3]. As a result we have a tendency to started exploring Hadoop as a technology to deal with our scaling desires. the very fact that Hadoop was already associate open supply project that was use at computer memory unit scale and provided measurability victimization trade goods hardware was a awfully compelling proposition for us. identical jobs that had taken quite daily to complete might currently be completed inside many hours victimization Hadoop. however writing a mapreduce program terribly troublesome currently a days for a posh downside which require very high finish programming skills and additionally takes code maintenance time.

IV PROPOSED WORK

For analzing these large and complex data we use Hive which is a popular query languages like SQL and as a result users ended up spending hours (for writing mapreduce code) to write programs for even simple analysis.

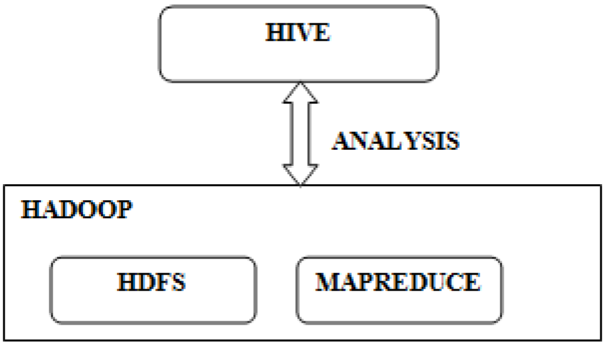


Figure-3. Workflow Diagram

Our Steps or Algorithm Steps will follow:

1. In first we can configure the hadoop on ubuntu and after integrate the hive on top of the hadoop.
2. Large amount of data is stored into HDFS through hadoop load command.
3. These large and complex data is pre-processed by mapreduce framework which process the data in parallel way.
4. Data is analyze by hive , on which we can write the hive query and hive compiler launches a mapreduce program to process the data and generates the result.
5. After that we can perform some optimization process on hive and compare the results.

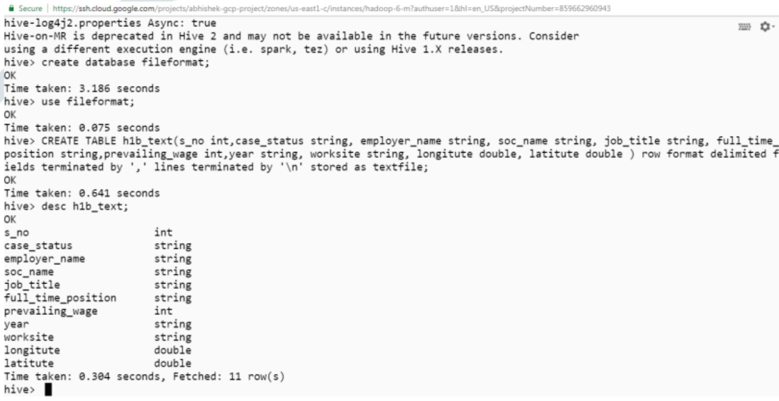
V. EXPERIMENTAL & RESULT ANALYSIS

All the experiments are perform on google cloud platform (GCP) on which we developed a heterogeneous clusters of five nodes. Cluster is implemented on ubuntu with hadoop is configure on it and top of the hadoop hive is working with default configuration, So to achieve this we are going to follow the following methods:

- Creating a tables with different file format.
- Launch a mapreduce job on these table.
- compare the performance of using these file formats.

Creating a table

Before analyzing the data we have to create a table to stored the data , So first we can create a four tables with different file formats, the data is stored in the table are based on there file formats so these file formats we can comparing are textfile, sequence file, RC file and ORC file. Figure 4 show the table creating by using text file.


Secure | https://ssh.cloud.google.com/projects/abhishek-gcp-project/zones/us-east1-c/instances/hadoop-6-m?authuser=1&hl=en_US&projectNumber=859662960943

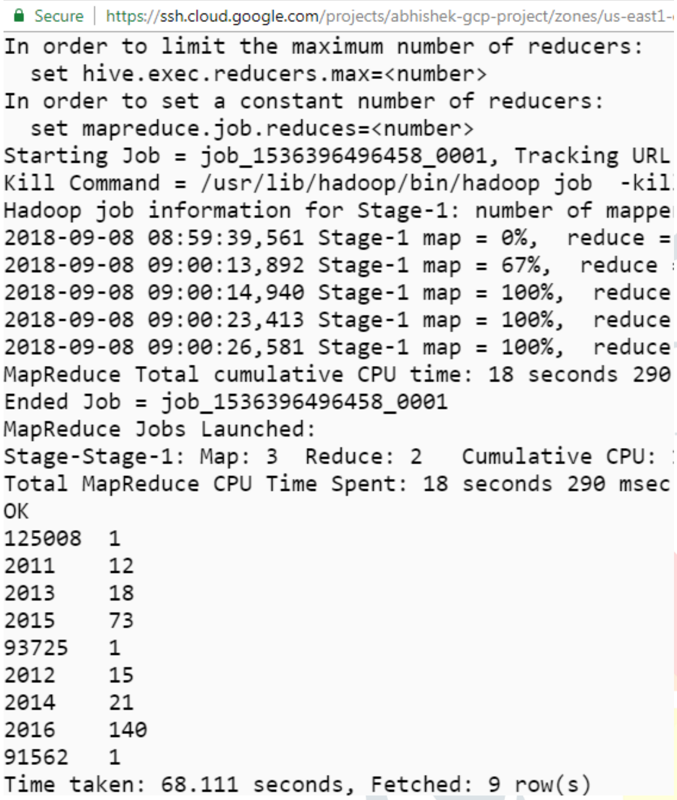
```
hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider
using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
hive> create database fileformat;
OK
Time taken: 3.186 seconds
hive> use fileformat;
OK
Time taken: 0.075 seconds
hive> CREATE TABLE h1b_text(s_no int,case_status string, employer_name string, soc_name string, job_title string, full_time_
position string,prevailing_wage int,year string, worksite string, longitute double, latitute double ) row format delimited f
ields terminated by ',' lines terminated by '\n' stored as textfile;
OK
Time taken: 0.641 seconds
hive> desc h1b_text;
OK
s_no                    int
case_status             string
employer_name           string
soc_name                string
job_title               string
full_time_position      string
prevailing_wage         int
year                    string
worksite                string
longitute               double
latitute                double
Time taken: 0.304 seconds, Fetched: 11 row(s)
hive> █
```

Figure 4. Table creation using file formats

After the table is created or data is being loaded into the table we can launched the sql query to analyse these data. So we can launch same query on all the different file format tables and compares it performance.

TEXTFILE

By default, if we use TEXTFILE format then each line is considered as a record. At the end, we need to specify the type of file format. If we do not specify anything it will consider the file format as TEXTFILE format.



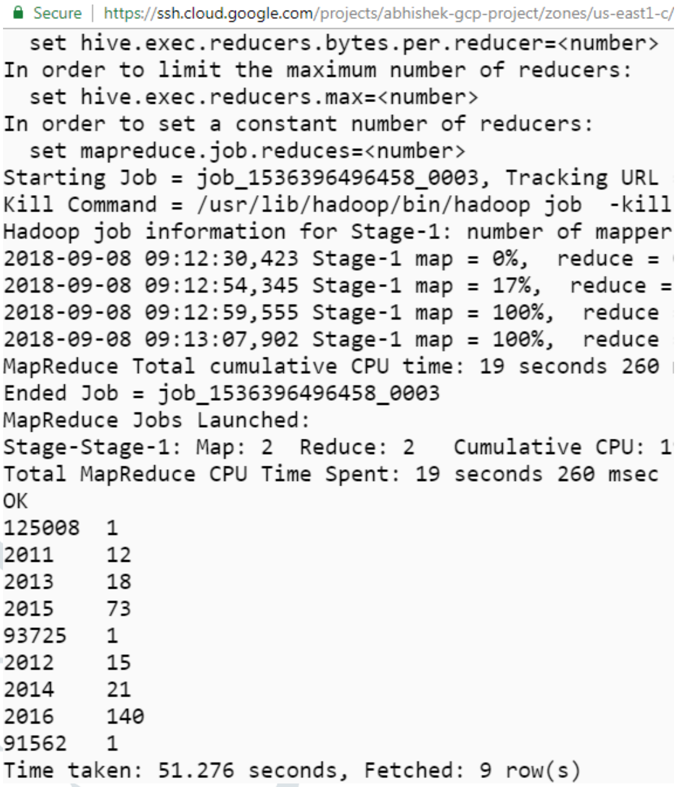
Secure | https://ssh.cloud.google.com/projects/abhishek-gcp-project/zones/us-east1-

```
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1536396496458_0001, Tracking URL
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kil
Hadoop job information for Stage-1: number of mappe
2018-09-08 08:59:39,561 Stage-1 map = 0%,  reduce =
2018-09-08 09:00:13,892 Stage-1 map = 67%,  reduce
2018-09-08 09:00:14,940 Stage-1 map = 100%,  reduce
2018-09-08 09:00:23,413 Stage-1 map = 100%,  reduce
2018-09-08 09:00:26,581 Stage-1 map = 100%,  reduce
MapReduce Total cumulative CPU time: 18 seconds 290
Ended Job = job_1536396496458_0001
MapReduce Jobs Launched:
Stage-Stage-1: Map: 3  Reduce: 2   Cumulative CPU:
Total MapReduce CPU Time Spent: 18 seconds 290 msec
OK
125008  1
2011    12
2013    18
2015    73
93725   1
2012    15
2014    21
2016    140
91562   1
Time taken: 68.111 seconds, Fetched: 9 row(s)
```

'Figure 5. Time taken on Textfile

SEQUENCEFILE

We know that Hadoop's performance is drawn out when we work with a small number of files with big size rather than a large number of files with small size. If the size of a file is smaller than the typical block size in Hadoop, we consider it as a small file. Due to this, a number of metadata increases which will become an overhead to the NameNode. To solve this problem sequence files are introduced in Hadoop. Sequence files act as a container to store the small files.



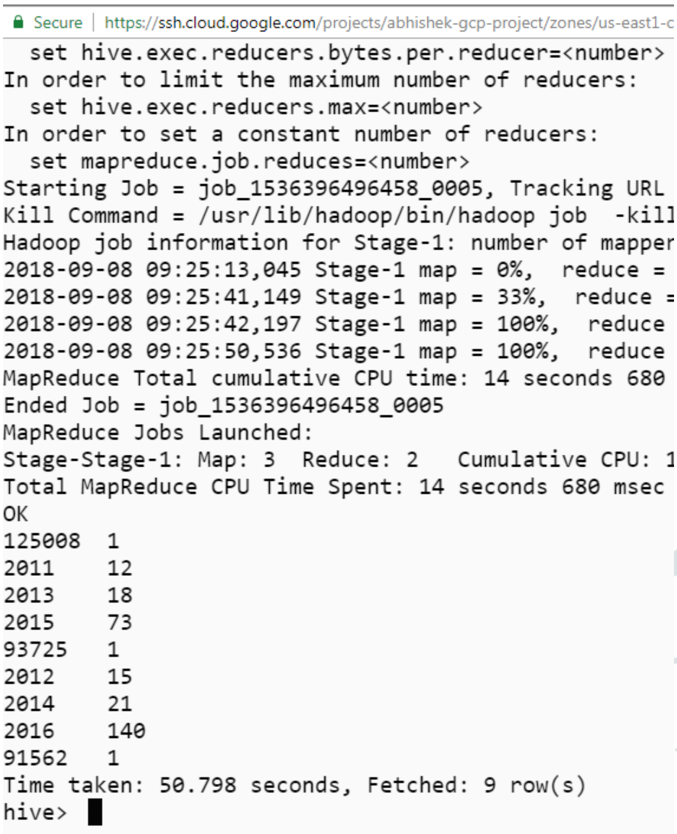
Secure | https://ssh.cloud.google.com/projects/abhishek-gcp-project/zones/us-east1-c/

```
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1536396496458_0003, Tracking URL
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill
Hadoop job information for Stage-1: number of mapper
2018-09-08 09:12:30,423 Stage-1 map = 0%,  reduce =
2018-09-08 09:12:54,345 Stage-1 map = 17%,  reduce =
2018-09-08 09:12:59,555 Stage-1 map = 100%,  reduce
2018-09-08 09:13:07,902 Stage-1 map = 100%,  reduce
MapReduce Total cumulative CPU time: 19 seconds 260
Ended Job = job_1536396496458_0003
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2  Reduce: 2   Cumulative CPU: 1
Total MapReduce CPU Time Spent: 19 seconds 260 msec
OK
125008  1
2011    12
2013    18
2015    73
93725   1
2012    15
2014    21
2016    140
91562   1
Time taken: 51.276 seconds, Fetched: 9 row(s)
```

Figure 6. ime taken on Sequencefile file format

RCFILE

RCFILE stands of Record Columnar File which is another type of binary file format which offers high compression rate on the top of the rows. RCFILE is used when we want to perform operations on multiple rows at a time.

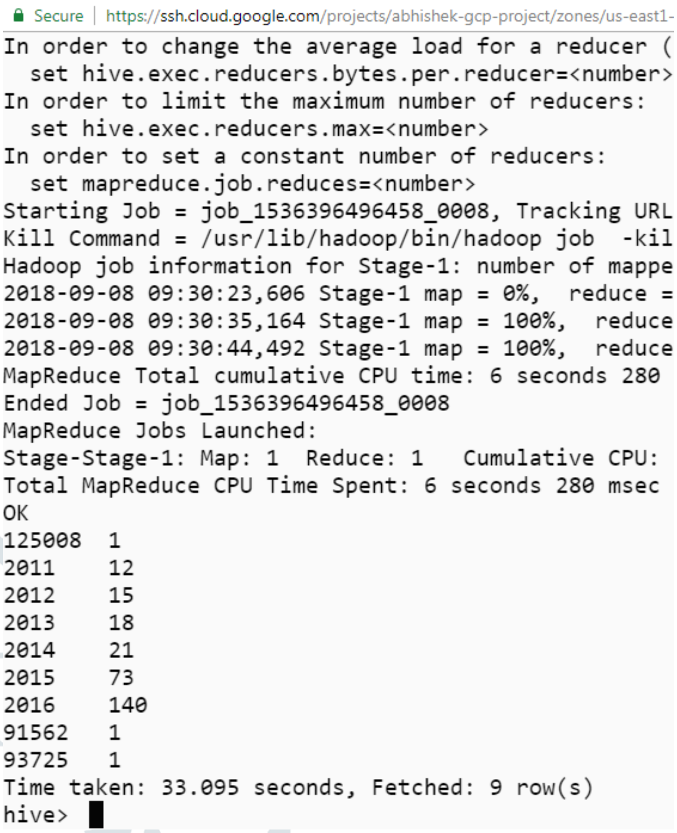


```
Secure | https://ssh.cloud.google.com/projects/abhishek-gcp-project/zones/us-east1-c
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1536396496458_0005, Tracking URL
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill
Hadoop job information for Stage-1: number of mapper
2018-09-08 09:25:13,045 Stage-1 map = 0%,  reduce =
2018-09-08 09:25:41,149 Stage-1 map = 33%,  reduce =
2018-09-08 09:25:42,197 Stage-1 map = 100%,  reduce
2018-09-08 09:25:50,536 Stage-1 map = 100%,  reduce
MapReduce Total cumulative CPU time: 14 seconds 680
Ended Job = job_1536396496458_0005
MapReduce Jobs Launched:
Stage-Stage-1: Map: 3  Reduce: 2   Cumulative CPU: 1
Total MapReduce CPU Time Spent: 14 seconds 680 msec
OK
125008  1
2011    12
2013    18
2015    73
93725   1
2012    15
2014    21
2016    140
91562   1
Time taken: 50.798 seconds, Fetched: 9 row(s)
hive> █
```

Figure 7. Time taken on RCfile file format

ORCFILE

ORC stands for Optimized Row Columnar which means it can store data in an optimized way than the other file formats. ORC reduces the size of the original data up to 75%(eg: 20GB file will become 5GB). As a result the speed of data processing also increases. ORC shows better performance than Text, Sequence and RC file formats.



```
Secure | https://ssh.cloud.google.com/projects/abhishek-gcp-project/zones/us-east1-
In order to change the average load for a reducer (
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1536396496458_0008, Tracking URL
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kil
Hadoop job information for Stage-1: number of mappe
2018-09-08 09:30:23,606 Stage-1 map = 0%,  reduce =
2018-09-08 09:30:35,164 Stage-1 map = 100%,  reduce
2018-09-08 09:30:44,492 Stage-1 map = 100%,  reduce
MapReduce Total cumulative CPU time: 6 seconds 280
Ended Job = job_1536396496458_0008
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU:
Total MapReduce CPU Time Spent: 6 seconds 280 msec
OK
125008  1
2011    12
2012    15
2013    18
2014    21
2015    73
2016    140
91562   1
93725   1
Time taken: 33.095 seconds, Fetched: 9 row(s)
hive> █
```

Figure 8. Time taken on ORCFILE file format

After launching the sql query on all the four tables which we create on the basis of different file format, we can compare the time taken by sql query on these file formats.

FILEFORMAT	TIME TAKEN IN SEC.
TEXTFILE	68.11
SEQUENCEFILE	51.27
RCFILE	50.79
ORCFILE	33.09

Tabe-1. Time taken on different file formats

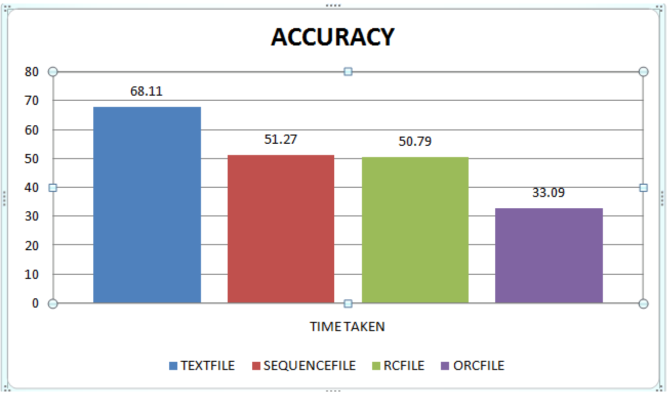


Figure 9. Comparison between file formats on time taking

VI CONCLUSION

Hive is a work in progress. It is an open-source project, and is being actively worked on by Facebook as well as several external contributors. Hive support SQL like language and transform these query into mapreduce job which is processed by hadoop framework and generates the output. In these we enhance the performance of SQL on Hadoop system by using different file format and proposed which file format is used in what kind of data . In these we can discussed Textfile, Sequencefile, RCfile, and ORCfile file format and ORC file format shows better performance that others.

REFERENCES

[01]Xiongpai Qin, Yueguo Chen*, Jun Chen, Shuai Li, Jiesi Liu, Huijie Zhang, "The Performance of SQL-on-Hadoop Systems: An Experimental Study" in IEEE, 2017 IEEE 6th International Congress on Big Data.

[02] Divya Kamath , Praveen Srinivas , Ashika Gopal , Lanchana B.V , Suma V "*A Profiling Tool for Apache HIVE Run-time Query*" in Proceedings of the IEEE 2017 International Conference on Computing Methodologies and Communication(ICCMC).

[03] Kamil Bajda-Pawlikowski, Daniel J. Abadi, Avi Silberschatz , Erik Paulson, "Efficient Processing of Data Warehousing Queries in a Split Execution Environment", SIGMOD'11, June 12–16, 2011, Athens, Greece.

[04] Swathi Prabhu, Anisha P Rodrigues, Guru Prasad M S & Nagesh H R, "Performance Enhancement of Hadoop MapReduce Framework for Analyzing BigData", IEEE 2015, 978-1-4799-608S-9/1S.

[05] White Paper Big Data Analytics Extract, Transform, and Load Big Data with Apache Hadoop-Intel corporation.

[06] Improving MapReduce Performance Using Data Prefetching mechanism in heterogeneous or Shared Environments Tao gu,Chuang Zuo,Qun Liao , Yulu Yang and Tao Li, International Journal of grid and distributed computing (2013).

[07] "Improve the MapReduce Performance through complexity and performance based on data placement in Heterogeneous Hadoop Cluster " Rajashekhar M. Arasanal, Daanish U. Rumani Department of Computer Science University of Illinois at Urbana-Champaign.

[08]Improving MapReduce Performance in Heterogeneous Network Environments and Resource Utilization, Zhenhua Guo, Geoffrey Fox IEEE (2012)

[09] Improving MapReduce Performance Using Smart Speculative Execution Strategy Qi Chen, Cheng Liu, and Zhen Xiao, Senior Member, IEEE 0018-9340/13/$26.00 ©
2013 IEEE

[10] S. Khalil, S. A. Salem, S. Nassar and E. M. Saad, "Mapreduce Performance in Heterogeneous Environments: A Review", International Journal of Scientific & Engineering Research, vol. 4, no. 4, (2013).