Case Study of Hive using Hadoop

[1] Sai Prasad Potharaju, [2] Shanmuk Srinivas A, [3] Ravi Kumar Tirandasu
[1,2,3] Assistant Professor ,
[1,2,3] SRES COE,Department of Computer Engineering ,
Kopargaon,Maharashtra, India

***Abstract* -Hadoop is a framework of tools. It is not a software that you can download on your computer. These tools are used to running applications on big data which has huge in capacity,need to process quickly and can be in variety forms. To manage the big data HIVE used as a data warehouse system for Hadoop that facilitates ad-hoc queries and the analysis of large datasets stored in Hadoop .Hive provides a SQL-LIKE languages called HIVEQL. In this paper we explains how to use hive using Hadoop with a simple real time example and also explained how to create a table,load the data into table from external file ,retrieve the data from table and their different statistics like CPU time for each stage of query execution ,cumulative CPU time and time taken to fetch records.**

***Key Words*:Hadoop,Hive,MapReduce,HDFS,HIVEQL**

1. INTRODUCTION

1.1. Hadoop

Hadoop is a open source and is distributed under Apache license. It is a framework of tools and not a software that you can download. These tools are used to running applications on big data .Big data means data with respective to its volume, speed, variety forms(Unstructured).In traditional approach big data is processed by using powerful computer but this computer will do good job until some limit,because computer is not scalable. It process according to its processor(core) type and speed,memory capacity.

Hadoop takes different approach (Fig 1)than its traditional approach. It breaks the data into smaller pieces .

Breaking the data into smaller pieces is good idea but how about computation?

Computation is also broken into pieces i.e processed into different levels/nodes and result will be combined together.

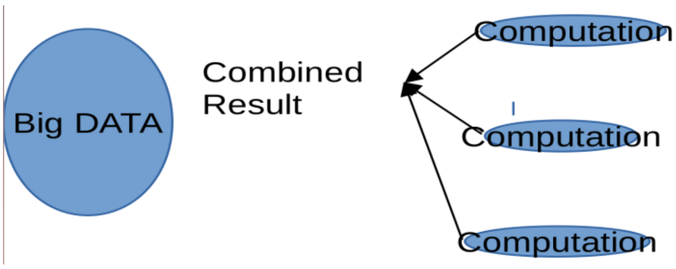

Fig.1.

At the highest level it has a simple architecture which breaks the data and computation into pieces .Hadoop has two important components .

A.MapReduce B. File System(HDFS).

One important characteristic of Hadoop is that it works on Distributed model and it is Linux based set of tools .

All the Machines or nodes under Hadoop will have two components .

A. Task Tracker(T)[1] [2] B. Data Node(D)[1][2].(Fig 2)

The job of task tracker is to process the task given to particular node and the job of data node is manage the data given to particular node .The nodes which process the piece of data and has task tracker and data nodes are called slaves. Above these slaves there will be a master which has additional components called 1.Job Tracker(J) [1][2]2.Name Node(N)[1][2].**(Fig 2)**

Job Tracker and Task Tracker are comes under MapReduce. Data Node and Name Node are comes under HDFS

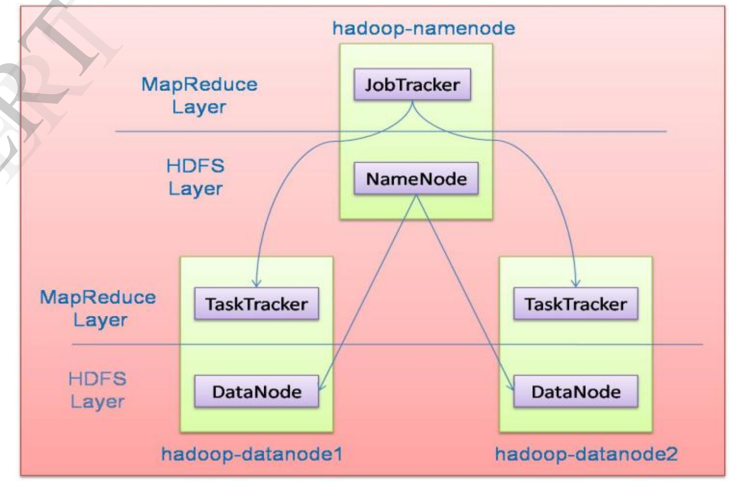

Fig. 2.

1.2. HIVE

Hive is a Data warehouse system layer built on Hadoop. It allows to define a structure for unstructured big data. It simplifies ,analysis and queries with an SQL like scripting language called HIVEQL.

Hive is not Relational Database,it uses a database to store meta data,but the data that hive processes is stored in HDFS. Hive is not designed for on-line transaction processing(OLTP).Hive runs on Hadoop which is a Batch-processing system where jobs can have high latency with substantial overhead. Therefore latency for hive queries is *g*enerally high even for small jobs. Hive is not suited for real-time queries and row level updates and it is best used for batch jobs over large sets of immutable data such as web logs.

2. THE TYPICAL USE CASE OF HIVE

Hive it takes large amount of unstructured data and place it into a structured view as shown in Fig.3.,that can be used by business analysts by the business tools. Hive supports use cases such as Ad-hoc queries,summarization,data analysis.

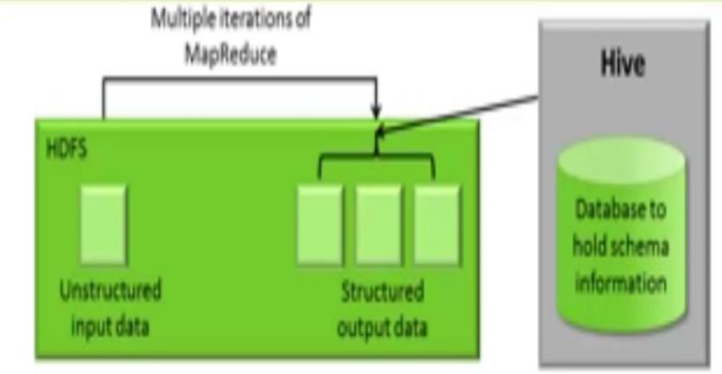
Multiple iterations of MapReduce
HDFS
Unstructured input data
Structured output data
Hive
Database to hold schema information

Fig.3.

HIVEQL can also be exchange with custom scalar functions means user defined functions(UDF ' S),aggregations(UDFA's) and table functions(UDTF's).

3. HIVE ARCHITECTURE

As shown in Fig.4 . Hive Architecture consists of

A. Meta store
It stores schema information and provides a structure to stored data .
B.HIVEQL
Which is for query processing,compiling,optimizing and execution ,those queries turned into MapReduce job which invokes mapper and reducer for processing.
HIVEQL is similar to other SQLs ,it uses familiar relational database concepts(tables,rows,columns,schema),it is based on the SQL-92 specifications. Hive supports multi-table inserts via the code means big data can be accessed via tables. It converts SQL queries into MapReduce jobs. User does not need to know MapReduce. Also supports plugging custom MapReduce scripts into queries.
C. User Interface
Hive can be approached by number of user interfaces. It has web UI,it has own command line interface and can be accessed using HDInsight.

Hive is good choice when you want to query the data,when you need an answer to specific questions if you are familiar with SQL.

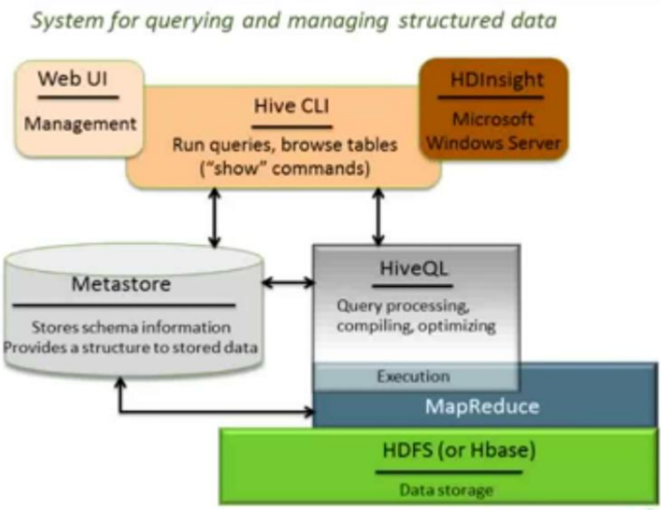
System for querying and managing structured data
Web UI
Management
Hive CLI
Run queries, browse tables ("show" commands)
HDInsight
Microsoft Windows Server
Metastore
Stores schema information
Provides a structure to stored data
HiveQL
Query processing, compiling, optimizing
Execution
MapReduce
HDFS (or Hbase)
Data storage

Fig.4.

4. USING OF HIVE WITH HADOOP.

A .Prerequisites to work with Hive

The prerequisites for setting up Hive and running queries are
1.User should have stable build of Hadoop
2.Machine Should have Java 1,6 installed
3.Basic Java Programming skills
4.Basic SQL Knowledge
5.Basic Knowledge of Linux
To start Hive first start all the services of Hadoop using the command **$ start-all.sh.** Then check all services are running or not using JPS command ,if all are services are running then Use the command $ **hive** to start HIVE

B.HIVE TABLES

Here we are explaining about basics of Hive Tables.

A Hive table consists of
Data: It is typically a file or group of files in HDFS.
Schema:It is in the form of meta data stored in a relational database. Schema and data are separate .A schema can be defined for existing data. Data can be added or removed independently. You have to define a schema if you have existing data in HDFS that you want to use in Hive

C. Managing Tables

See current tables
hive>Show Tables;

Check the schema
hive>Describe mytable;

change the table name
hive> ALTER TABLE mytablename RENAME to mt;

Add a Column

```
hive>ALTER TABLE mytable ADD COLOUMNS (mycol STRING);
```

Drop a Partition

```
hive >ALTER TABLE mytable DROP PARTITION(age=17)
```

D. Loading Data

Use LOAD DATA to import data into Hive Table

```
hive>LOAD DATA LOCAL INPATH ' path of input data' INTO TABLE mytable;
```

The files are not modified by Hive. Use the word OVERWRITE[3][4] to write over a file of the same name. Hive warehouse default location is /hive/warehouse. Hive can read all of the files in a particular directory. The schema is checked when the data is required and if a row does not match the schema,it will be read as null.

E. Insert

Use INSERT[3] statement to populate data into a table from another HIVE table. Since query results are usually large it is best to use an INSERT clause to tell HIVE where to store your query.

Example:

```
hive>CREATE TABLE age_count(name STRING,age INT);
hive>INSERT OVERWRITE TABLE age_count
                    SELECT age,COUNT(age)
                    FROM mytable;
```

F. Performing Queries

SELECT statement is used for this. It supports the following:

Where clause
UNION ALL and DISTINCT
GROUP BY and HAVING
LIMIT clause
Can Use REGULAR EXPRESSION Column Specification

5. CASE STUDY

In this we are giving a real example how to use HIVE on Top of the HADOOP and different statistical analysis. In this example we have a predefined dataset (cricket.csv) having more than 20 columns and more than 10000 records in it. In the below example we are explaining how to display few records based on attributes of cricket.

The cricket.csv file has different attributes like name,country, Player_iD,Year.stint,teamid,run,igl,D,G,G_batting.AB,R,H,2B,3B,HR,RBI,SB,CS,BB,SO,IBB,HBP,SH,SF,GIDP,G_OLD

Follow the below steps in order to retrieve the data from Big tables.

1.Start Hadoop

```
$ start-all.sh
```

2.Create a Folder and copy all predefined datasets

```
$ hadoop fs -mkdir sampledataset
$ hadoop fs -put /home/sai/hivedataset/ /user/sai/sampledataset
```

Note:hivedataset has Big Data tables in .cs format ,those tables are copying from my local file system to Hadoop.

3.Start Hive

```
$ hive
```

4.Create Table in Hive

```
hive>create table temp_cricket (col_value STRING);
```

5.Load the Data from csv file to table temp_batting

```
hive> LOAD DATA INPATH
'/user/sai/sampledataset/hivedataset/cricket.csv'
OVERWRITE INTO TABLE temp_cricket;
```

6.Create Another table to Extract data

```
hive>create table batting (player_id STRING, year INT, runs INT);
```

7.Insert data Into newly created data By extracting

```
hive>insert overwrite table batting
SELECT
regexp_extract(col_value, '^(?:([^,]*)\,?){1}',3) player_id,
regexp_extract(col_value, '^(?:([^,]*)\,?){2}', 4) year,
regexp_extract(col_value, '^(?:([^,]*)\,?){9}', 7) run
from temp_cricket;
```

In above query from temp_cricket table player_id,year,run are loaded into batting table .

Execute Few Queries and observe the result

Example 1

Display year and maximum runs scored in each year

```
hive> SELECT year, max(runs) FROM batting GROUP BY year;
```

```
Total Jobs=1
hadoop job information for stage-1:
Number of mappers:1;Number of reducers:1
Stage-1 map=0% reduce =0%
Stage-1 map=100% reduce =0%  cumulative CPU 3.45 sec
Stage-1 map=100% reduce =100%  cumulative CPU 6.35 sec
MapReduce Total cumulative CPU time:6.35 sec
Time taken :46.09 sec
```

Example 2

Display player_id,year and maximum runs scored in each year

```
hive>     SELECT a.year, a.player_id, a.runs from batting a
          JOIN (SELECT year, max(runs) runs FROM batting
          GROUP BY year ) b
          ON (a.year = b.year AND a.runs = b.runs) ;
```

```
Total Jobs=2
Launching job 1 out of 2
hadoop job information for stage-2:
Number of mappers:1;Number of reducers:1
Stage-2 map=0% reduce =0%
Stage-2 map=100% reduce =0%  cumulative CPU 3.0 sec
Stage-2 map=100% reduce =100%  cumulative CPU 5.95 sec
MapReduce Total cumulative CPU time:5.95 sec

Launching job 2 out of 2
hadoop job information for stage-3:
Number of mappers:1;Number of reducers:0
Stage-3 map=0% reduce =0%
Stage-3 map=100% reduce =0%  cumulative CPU 3.12 sec

Stage-3 map=100% reduce =100%  cumulative CPU 3.12 sec
MapReduce Total cumulative CPU time:3.12 sec
Time taken :65.743sec
```

As there are two queries (Select Statements) two jobs have been executed. The CPU time varies according to the system configuration

CONCLUSION

Hive works with Hadoop to allow you to query and manage large-scale data using a familiar SQL-like interface. Hive provides command line interface to the shell and Microsoft HDInsight provides console access. Hive tables consist of data and schema and they are separated for maximum flexibility. Hive Query Language(HIVEQL) supports familiar SQL operations including joins,sub queries,Order By,Sort by etc. The CPU analysis depends on Hadoop configuration nodes and System configuration

REFERENCES

[1] Hadoop. http://hadoop.apache.org, 2009.

[2] HDFS(hadoop distributed file system) architecture. http://hadoop.apache.org/common/docs/current/hdfsdesin.html, 2009

[3] Ashish Thusoo, Joydeep Sen Sarma, Namit Jain, Zheng Shao,Prasad Chakka, Suresh Anthony, Hao Liu, Pete Wyckoff and Raghotham Murthy.Hive - A Warehousing Solution Over a Map-Reduce Framework .http://www.vldb.org/pvldb/2/vldb09-938.pdf

[4] JasonRutherglen,Dean Wampler,Edward CaprioloE. ttp://www.reedbushey.com/99Programming%20Hive.pdf