

Data Processing using Hive

Tushar B. Kute,
<http://tusharkute.com>

What is Apache Hive

- Hive is a data warehouse infrastructure tool to process structured data in Hadoop.
- It resides on top of Hadoop to summarize Big Data, and makes querying and analyzing easy.
- Initially Hive was developed by Facebook, later the Apache Software Foundation took it up and developed it further as an open source under the name Apache Hive.
- It is used by different companies. For example, Amazon uses it in Amazon Elastic MapReduce.

Features of Hive

- It stores schema in a database and processed data into HDFS.
- It is designed for OLAP.
- It provides SQL type language for querying called HiveQL or HQL.
- It is familiar, fast, scalable, and extensible.

Hive Installation

- Please download the method from here:
<http://mitu.co.in/bigdata-presentations>

Problem Statement

- Write an application using HiveQL for flight information system which will include:
 - Creating, Dropping, and Altering Database tables.
 - Load table with data, insert new values and field in the table, Join tables with Hive.
 - Create index on Flight information Table.
 - Find the average departure delay per day in 2008.

Sample database operations

```
mitu@skillologies: ~  
hive> create database db1;  
OK  
Time taken: 0.069 seconds  
hive> use db1;  
OK  
Time taken: 0.012 seconds  
hive> create table flight (fno int, year int, dest varchar(10),  
    delay float);  
OK  
Time taken: 0.191 seconds  
hive> alter table flight rename to air_flight;  
OK  
Time taken: 0.897 seconds
```

More alter commands

```
mitu@skillologies: ~  
hive> alter table air_flight add columns (source varchar(10));  
OK  
Time taken: 0.247 seconds  
hive> alter table air_flight change source src varchar(15);  
OK  
Time taken: 0.244 seconds
```

```
mitu@skillologies: ~  
hive> drop table flight;  
OK  
Time taken: 0.288 seconds  
hive>
```

Start using Hive command line

```
mitu@skillologies:~$ jps
```

```
9669 NodeManager
```

```
9209 DataNode
```

```
9065 NameNode
```

```
10219 Jps
```

```
9549 ResourceManager
```

```
9391 SecondaryNameNode
```

```
mitu@skillologies:~$ hive
```

```
Logging initialized using configuration in jar:file:/usr/local/hive  
e-common-1.2.1.jar!/hive-log4j.properties
```

```
hive> create database mydb;
```

```
OK
```

```
Time taken: 1.528 seconds
```

```
hive> use mydb;
```

```
OK
```

```
Time taken: 0.109 seconds
```


Create the table

```
mitu@skillologies: ~  
hive> create table flight (fno int, year int, dest varchar(10),  
    delay float);  
OK  
Time taken: 2.082 seconds  
hive> desc flight;  
OK  
fno                int  
  
year               int  
  
dest               varchar(10)  
  
delay              float  
  
Time taken: 0.877 seconds, Fetched: 4 row(s)
```

Table creating methodology

```
mitu@skillologies: ~  
hive> create table flight (fno int, year int, dest varchar(10), delay float)  
      > row format delimited  
      > fields terminated by ','  
      > lines terminated by '\n'  
      > stored as textfile;  
OK  
Time taken: 0.691 seconds  
hive>
```

Insert the values

```
mitu@skillologies: ~  
hive> insert into flight values (123, 2009, "Mumbai", 30.0);  
Query ID = mitu_20180328120405_5ac7f04f-19ba-413f-827d-d311a611dd51  
Total jobs = 3  
Launching Job 1 out of 3  
Number of reduce tasks is set to 0 since there's no reduce operator  
Job running in-process (local Hadoop)  
2018-03-28 12:04:20,804 Stage-1 map = 0%, reduce = 0%  
2018-03-28 12:04:21,845 Stage-1 map = 100%, reduce = 0%  
Ended Job = job_local1290390528_0001  
Stage-4 is selected by condition resolver.  
Stage-3 is filtered out by condition resolver.  
Stage-5 is filtered out by condition resolver.  
Moving data to: hdfs://localhost:54310/user/hive/warehouse/mydb.db/flight/.hive-  
staging_hive_2018-03-28_12-04-05_186_4559567806608736141-1/-ext-10000  
Loading data to table mydb.flight
```

Insert queries

- insert into flight values (123, 2009, "Mumbai", 30.0);
- insert into flight values (342, 2008, "Nagpur", 13.0);
- insert into flight values (232, 2008, "Aurangabad", 0.0);
- insert into flight values (103, 2009, "Kolhapur", 10.0);
- insert into flight values (200, 2008, "Jalgaon", 50.0);
- insert into flight values (112, 2009, "Amravati", 0.0);

Show table contents

```
mitu@skillologies: ~  
hive> select * from flight;  
OK  
123      2009      Mumbai    30.0  
342      2008      Nagpur    13.0  
232      2008      Aurangabad      0.0  
103      2009      Kolhapur      10.0  
200      2008      Jalgaon 50.0  
112      2009      Amravati      0.0  
Time taken: 0.661 seconds, Fetched: 6 row(s)  
hive>
```

Loading a text data locally

flight_data.txt

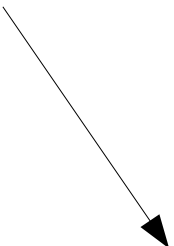
```
1 923,2009,Navi Mumbai,60.0  
2 156,2009,Kolhapur,30.0  
3 112,2009,Amravati,0.0  
4 322,2008,Nagpur,0.0  
5 132,2008,Aurangabad,10.0  
6 170,2008,Jalgaon,40.0
```

Loading a text data locally

```
mitu@skillologies: ~  
hive> load data local inpath "flight_data.txt"  
      > overwrite into table flight;  
Loading data to table mydb.flight  
Table mydb.flight stats: [numFiles=1, numRows=0, totalSize=138, rawDataSize=0]  
OK  
Time taken: 0.683 seconds  
hive> select * from flight;  
OK  
923      2009      Navi Mumba      60.0  
156      2009      Kolhapur        30.0  
112      2009      Amravati        0.0  
322      2008      Nagpur 0.0  
132      2008      Aurangabad      10.0  
170      2008      Jalgaon 40.0  
Time taken: 0.038 seconds, Fetched: 6 row(s)  
hive>
```

Creating a new table

```
mitu@skillologies: ~  
hive> create table nflight (fno int, year int, source varchar(10))  
  > row format delimited  
  > fields terminated by ','  
  > lines terminated by '\n'  
  > stored as textfile;  
OK  
Time taken: 0.48 seconds  
hive>
```



```
mitu@skillologies: ~  
hive> select * from nflight;  
OK  
112      2007      Pune  
322      2009      Pune  
170      2009      Pune  
Time taken: 0.166 seconds, Fetched: 3 row(s)  
hive>
```


Joining the tables

```

mitu@skillologies: ~
hive> select a.fno, a.year, a.dest, a.delay, b.source
      > from flight a join nflight b
      > on (a.fno = b.fno);
Query ID = mitu_20180328134721_284e7df1-a0b0-40d3-9b8c-8e345a7488
Total jobs = 1
2018-03-28 13:47:30,968 WARN [main] util.NativeCodeLoader: Unable
to load native-hadoop library for your platform... using builtin-java classes
instead
Execution log at: /tmp/mitu/mitu_20180328134721_284e7df1-a0b0-40d3-9b8c-8e345a748821.log

```

Total MapReduce CPU Time Spent: 0 msec
 OK

112	2009	Amravati	0.0	Pune
322	2008	Nagpur	0.0	Pune
170	2008	Jalgaon	40.0	Pune

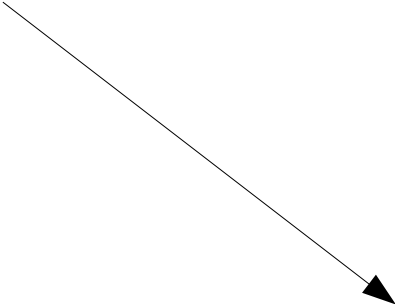
 Time taken: 16.548 seconds, Fetched: 3 row(s)
 hive>

Creating an index

- An Index is nothing but a pointer on a particular column of a table.
- Creating an index means creating a pointer on a particular column of a table.

Creating index

```
mitu@skillologies: ~  
hive> create index flight_index on table flight(fno)  
      > as 'org.apache.hadoop.hive ql.index.compact.CompactIndexHandler'  
      > WITH DEFERRED REBUILD;  
OK  
Time taken: 0.988 seconds  
hive>
```



```
mitu@skillologies: ~  
hive> show tables;  
OK  
flight  
mydb_flight_flight_index  
nflight  
values __tmp__table__1  
values __tmp__table__10  
values __tmp__table__11
```

Sample Query


- Find the average departure delay per day in 2008.

- Example:

```
select avg(delay) from flight where  
year = 2008;
```

Query output

```
mitu@skillologies: ~  
hive> select avg(delay) from flight where year = 2008;  
Query ID = mitu_20180328140022_2c343f4a-781a-47ce-88fd-1cbacd629087  
Total jobs = 1  
Launching Job 1 out of 1  
Number of reduce tasks determined at compile time: 1  
In order to change the average load for a reducer (in bytes):  
    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>  
Job running in-process (local Hadoop)  
2018-03-28 14:00:24,656 Stage-1 map = 100%,  reduce = 100%  
Ended Job = job_local1843554894_0017  
MapReduce Jobs Launched:  
Stage-Stage-1:  HDFS Read: 4564 HDFS Write: 3800 SUCCESS  
Total MapReduce CPU Time Spent: 0 msec  
OK  
16.666666666666668
```



Thank you

This presentation is created using LibreOffice Impress 5.1.6.2, can be used freely as per GNU General Public License



@mitu_skillologies



/mITuSkillologies



@mitu_group



/company/mitu-
skillologies

Web Resources

<http://mitu.co.in>

<http://tusharkute.com>

contact@mitu.co.in

tushar@tusharkute.com