CLOUDERA

2016

Hadoop - Hive Lab

Sriram Balasubramanian

CALIFORNIA - UNITED STATES OF AMERICA

**HIVE LAB**



**Table of Contents**

[**Hive Lab Assignment 2**](#_Toc465925373)

[**Hive – Execution 2**](#_Toc465925374)

[**Scenario 1: Create a managed table and load the data from LFS 3**](#_Toc465925375)

[**Scenario 2: Create a managed table and load the data from HDFS 4**](#_Toc465925376)

[**Scenario 3: Create an external table and load the data from LFS 5**](#_Toc465925377)

[**Scenario 4: Create an external table and load the data from HDFS 6**](#_Toc465925378)

[**Scenario 5: Drop a managed table and check the result in HDFS 7**](#_Toc465925379)

[**Scenario 6: Drop an external table and check the data from HDFS 7**](#_Toc465925380)

[**Programming in Hive Script 8**](#_Toc465925381)

[**JOINS using Hive 9**](#_Toc465925382)

[**Static Partitioning using Hive 12**](#_Toc465925383)

[**Dynamic Partitioning using Hive 15**](#_Toc465925384)

[**Bucketing using Hive 18**](#_Toc465925385)

[**Complex data type in Hive: Array 21**](#_Toc465925386)

[**Complex data type in Hive: Struct 23**](#_Toc465925387)

[**Complex data type in Hive: Map 25**](#_Toc465925388)

[**Hive UDF 26**](#_Toc465925389)

[**Integration of Pig with HBase 29**](#_Toc465925390)

# **Hive Lab Assignment**

# **Hive – Execution**

**To start the hive terminal**

**[cloudera@quickstart ~]$ hive**

2016-10-31 23:49:51,032 WARN [main] mapreduce.TableMapReduceUtil: The hbase-prefix-tree module jar containing PrefixTreeCodec is not present. Continuing without it.

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.properties

WARNING: Hive CLI is deprecated and migration to Beeline is recommended.

**hive>**

**Hive Oriented Scenarios**

**Scenario 1: Create a managed table and load the data from LFS**

**Scenario 2: Create a managed table and load the data from HDFS**

**Scenario 3: Create an external table and load the data from LFS**

**Scenario 4: Create an external table and load the data from HDFS**

**Scenario 5: Drop a managed table and check the result in HDFS**

**Scenario 6: Drop an external table and check the data from HDFS**

## **Scenario 1: Create a managed table and load the data from LFS**

**Flat File Creation**

**Flat file:One.txt**

**1,sriram**

**2,raj**

**Table Creation**

**hive> create table sri\_cust(cid int,cname string) rowformat delimited fieldsterminated by ',';**

OK

Time taken: 2.866 seconds

**Loading the data from LFS**

**hive> load data local inpath '/home/cloudera/Desktop/one.txt'into table sri\_cust;**

Loading data to table default.sri\_cust

Table default.sri\_cust stats: [numFiles=1, totalSize=15]

OK

Time taken: 1.185 seconds

**Retrieving the data**

**hive> select \* from sri\_cust;**

OK

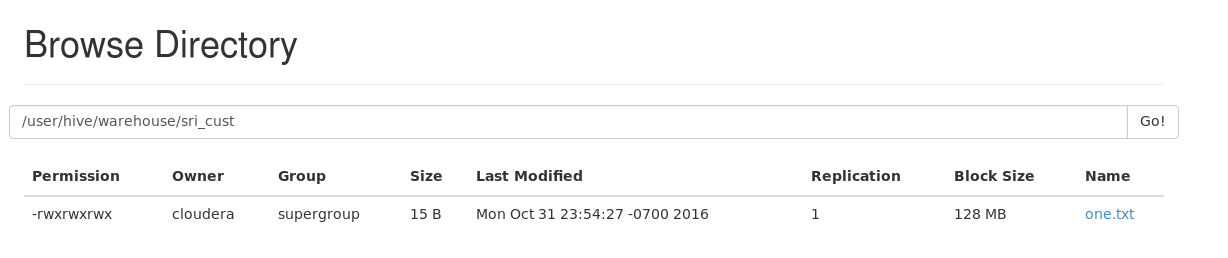
1 sriram

2 raj

Time taken: 0.708 seconds, Fetched: 2 row(s)

Browse the Directory / Check the result in HDFS

**For managed tables the values are stored under hive Meta store**

****

## **Scenario 2: Create a managed table and load the data from HDFS**

**Flat File Creation**

**Flat file:One.txt**

**1,sriram**

**2,raj**

**Open a new Terminal**

**[cloudera@quickstart ~]$ hadoop fs -put /home/cloudera/Desktop/one.txt /**

**Table Creation**

**hive> create table cust\_sri(cid int,cname string)rowformat delimited fieldsterminated by ',';**

OK

Time taken: 0.11 seconds

**Loading data from HDFS**

**hive> load data inpath '/one.txt' into tablecust\_sri;**

Loading data to table default.cust\_sri

Table default.cust\_sri stats: [numFiles=1, totalSize=15]

OK

Time taken: 0.56 seconds

**Retrieving data**

**hive> select \* from cust\_sri;**

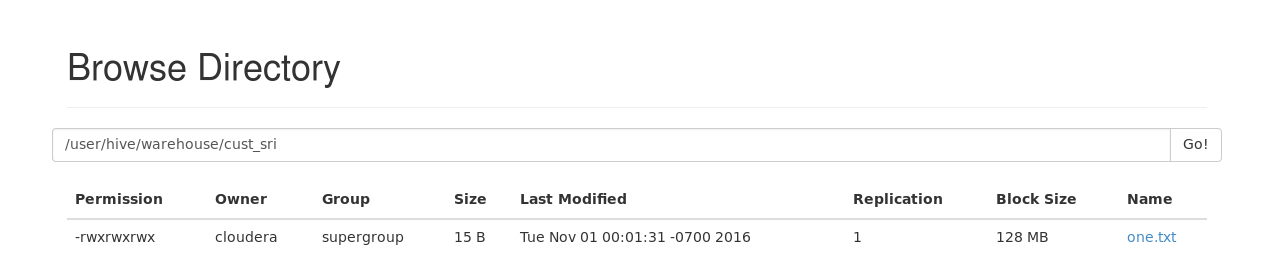
OK

1 sriram

2 raj

Time taken: 0.526 seconds, Fetched: 2 row(s)

Browse the Directory / Check the result in HDFS



## **Scenario 3: Create an external table and load the data from LFS**

**Flat File Creation**

**Sritwo.txt**

US,1,United States

CHN,2,China

**Creating an external table**

hive> create external table sri\_ext1(cname string,cid int,des string) row format delimited fields terminated by ','

> location '/user/cloudera/result\_ext';

OK

Time taken: 0.264 seconds

**Loading data from LFS for external table**

hive> load data local inpath '/home/cloudera/Desktop/sritwo.txt' into table sri\_ext1;

Loading data to table default.sri\_ext1

Table default.sri\_ext1 stats: [numFiles=1, totalSize=31]

OK

Time taken: 0.327 seconds

**Retrieving table information**

hive> select \* from sri\_ext1;

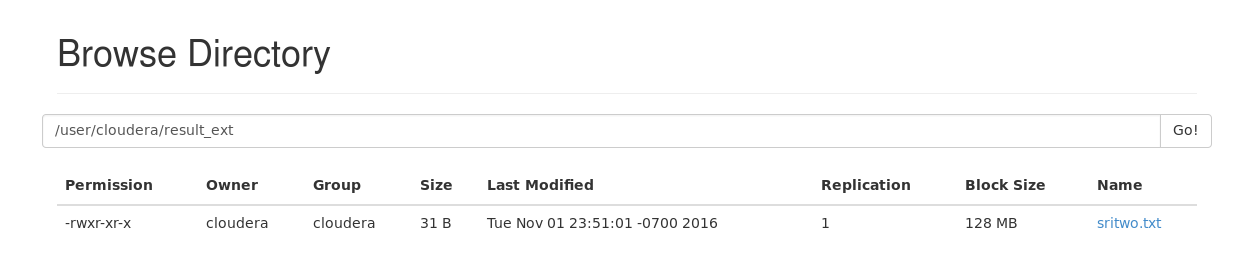
OK

US 1 United States

CHN 2 China

Time taken: 0.09 seconds, Fetched: 2 row(s)

Browse the Directory / Check the result in HDFS



## **Scenario 4: Create an external table and load the data from HDFS**

**Creating external table**

hive> create external table sri\_ext2(cname string,cid int,des string) row format delimited fields terminated by ','

> location '/user/cloudera/result\_ext1';

OK

Time taken: 0.259 seconds

**Loading the data from HDFS**

hive> load data inpath '/sricountry.txt' into table sri\_ext2;

Loading data to table default.sri\_ext2

Table default.sri\_ext2 stats: [numFiles=1, totalSize=31]

OK

Time taken: 0.233 seconds

**Retrieving table information**

hive> select \* from sri\_ext2;

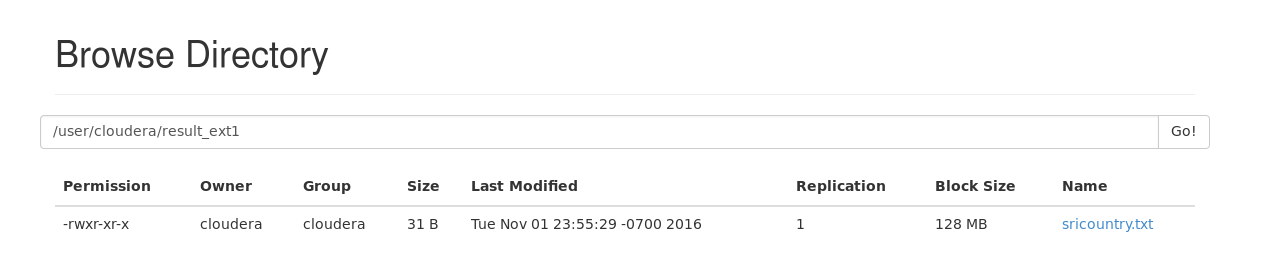
OK

US 1 United States

CHN 2 China

Time taken: 0.09 seconds, Fetched: 2 row(s)

Browse the Directory / Check the result in HDFS



## **Scenario 5: Drop a managed table and check the result in HDFS**

**Dropping scenario**

hive> select \* from cust\_sri;

OK

1 sriram

2 raj

Time taken: 1.291 seconds, Fetched: 2 row(s)

**Dropping internal table**

**If u drop the internal table the meta data and actual data is deleted.**

hive> drop table cust\_sri;

OK

Time taken: 0.729 seconds

hive> select \* from sri\_ext1;

OK

US 1 United States

CHN 2 China

Time taken: 0.088 seconds, Fetched: 2 row(s)

## **Scenario 6: Drop an external table and check the data from HDFS**

**If u drop the external table the meta data is deleted and actual data is not deleted.**

hive> drop table sri\_ext1;

OK

Time taken: 0.128 seconds

# **Programming in Hive Script**

**Loading data file through Hive Script**

**Flat File Creation**

**Product\_sri.txt**

**1 BigBooks 20.1 stationery**

**2 pens 45.6 stationery**

**3 Furniture 67.8 Householditems**

**Code:**

**Productscript.sql**

create table product\_tab(pid int,pname string,price float,des string) row formatdelimited fields terminated by '\t';

load data local inpath '/home/cloudera/Desktop/product\_sri.txt' into table product\_tab;

select \* from product\_tab;

**Execution**

**[cloudera@quickstart ~]$ hive -f /home/cloudera/Desktop/productscript.sql**

2016-11-02 03:17:43,908 WARN [main] mapreduce.TableMapReduceUtil: The hbase-prefix-tree module jar containing PrefixTreeCodec is not present. Continuing without it.

**Logging initialized**using configuration in file:/etc/hive/conf.dist/hive-log4j.properties

OK

Time taken: 1.239 seconds

**Loading data to table default.product\_tab**

Table default.product\_tab stats: [numFiles=1, totalSize=82]

OK

Time taken: 0.835 seconds

OK

1 BigBooks 20.1 stationery

2 pens 45.6 stationery

3 Furniture 67.8 Householditems

Time taken: 0.657 seconds, Fetched: 3 row(s)

# **JOINS using Hive**

**Flat File Creation**

**empdataset.txt**

1RamUS

2DiyaUS

3SriramIND

4JanaIND

**deptdataset.txt**

1IT

2IT

3Analyst

4Admin

Invoke the hive terminal

**$ hive**

Table Creation

**hive> create table empjoin(eid INT,ename STRING,address STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t';**

OK

Time taken: 0.848 seconds

Load the data

**hive> load data local inpath '/home/cloudera/Desktop/empdataset.txt' into table empjoin;**

Loading data to table default.empjoin

Table default.empjoin stats: [numFiles=1, totalSize=43]

OK

Time taken: 0.606 seconds

Verify the loaded records

**hive> select \* from empjoin;**

OK

1RamUS

2DiyaUS

3SriramIND

4JanaIND

5RajCHN

Time taken: 0.451 seconds, Fetched: 4 row(s)

Table Cration

**hive> create table deptjoin(eid INT,dept STRING) row format delimited fields terminated by '\t';**

OK

Time taken: 0.089 seconds

Load the data

**hive> load data local inpath '/home/cloudera/Desktop/deptdataset.txt' into table deptjoin;**

Loading data to table default.deptjoin

Table default.deptjoin stats: [numFiles=1, totalSize=29]

OK

Time taken: 0.243 seconds

Verify the loaded records

**hive> select \* from deptjoin;**

OK

1IT

2IT

3Analyst

4Admin

7HR

Time taken: 0.09 seconds, Fetched: 4 row(s)

Inner JOIN

hive> select \* from empjoin JOIN deptjoin ON (empjoin.eid=deptjoin.eid);

OK

1RamUSIT

2DiyaUS2IT

3SriramIND3Analyst

4JanaIND4Admin

Time taken: 33.448 seconds, Fetched: 4 row(s)

LEFT OUTER JOIN

hive> select e.eid,ename,dept from empjoin e LEFT OUTER JOIN deptjoin d ON(e.eid=d.eid);

OK

1RamIT

2DiyaIT

3SriramAnalyst

4JanaAdmin

5RajNULL

RIGHT OUTER JOIN

hive> select e.eid,ename,dept from empjoin e RIGHT OUTER JOIN deptjoin d ON(e.eid=d.eid);

OK

1RamIT

2DiyaIT

3SriramAnalyst

4JanaAdmin

FULL OUTER JOIN

hive> select e.eid,ename,dept from empjoin e FULL OUTER JOIN deptjoin d ON(e.eid=d.eid);

OK

1RamIT

2DiyaIT

3SriramAnalyst

4JanaAdmin

5RajNULL

NULLNULLHR

Time taken: 32.245 seconds, Fetched: 6 row(s)

# **Static Partitioning using Hive**

**Flat File Creation**

**user\_info.txt**

satyam,kumar,89

prateek,kumar,78

diya,anand,76

ashu,singh,74

**user\_info1.txt**

manish,kumar,76

sohail,tanvir,89

lovely,choudhary,4

Invoke the hive terminal

**$hive**

Table Creation

**hive> create table part\_user(fname varchar(20),lname varchar(20),eid int) partitioned by (country varchar(20),state varchar(20)) row format delimited fields terminated by ','stored as textfile;**

OK

Time taken: 1.653 seconds

**hive> desc part\_user;**

OK

fname varchar(20)

lname varchar(20)

eid int

country varchar(20)

state varchar(20)

# Partition Information

# col\_name data\_type comment

country varchar(20)

state varchar(20)

Time taken: 0.515 seconds, Fetched: 11 row(s)

Load the data

**hive> load data local inpath '/home/cloudera/Desktop/user\_info.txt' into table part\_user partition (country='US',state='FL');**

Loading data to table default.part\_user partition (country=US, state=FL)

Partition default.part\_user{country=US, state=FL} stats: [numFiles=1, numRows=0, totalSize=61, rawDataSize=0]

OK

Time taken: 0.874 seconds

Verify the loaded records

**hive> select \* from part\_user;**

OK

satyam kumar 89 US FL

prateek kumar 78 US FL

diya anand 76 US FL

ashu singh 74 US FL

Time taken: 0.464 seconds, Fetched: 4 row(s)

Load the data

**hive> load data local inpath '/home/cloudera/Desktop/user\_info1.txt' into table part\_user partition (country='CA',state='AU');**

Loading data to table default.part\_user partition (country=CA, state=AU)

Partition default.part\_user{country=CA, state=AU} stats: [numFiles=1, numRows=0, totalSize=52, rawDataSize=0]

OK

Time taken: 1.302 seconds

Verify the loaded records

**hive> select \* from part\_user;**

OK

manish kumar 76 CA AU

sohail tanvir 89 CA AU

lovely choudhary 4 CA AU

satyam kumar 89 US FL

prateek kumar 78 US FL

diya anand 76 US FL

ashu singh 74 US FL

Time taken: 0.572 seconds, Fetched: 7 row(s)

Retrieving information: Verify the loaded records

**hive> select \* from part\_user where part\_user.country='US' and part\_user.state='FL';**

OK

satyam kumar 89 US FL

prateek kumar 78 US FL

diya anand 76 US FL

ashu singh 74 US FL

Time taken: 1.252 seconds, Fetched: 4 row(s)

**hive> select \* from part\_user where part\_user.country='CA' and part\_user.state='AU';**

OK

manish kumar 76 CA AU

sohail tanvir 89 CA AU

lovely choudhary 4 CA AU

Time taken: 0.126 seconds, Fetched: 3 row(s)

# **Dynamic Partitioning using Hive**

Table Creation

**hive> create table par\_user1(fname string,lname string,eid int) partitioned by**

**> (country string,state string) row format delimited fields terminated by ',' stored as textfile;**

OK

Time taken: 0.882 seconds

**hive> create table user1(fname string,lname string,eid int,country string,state**

**> string) row format delimited fields terminated by ',' stored as textfile;**

OK

Time taken: 0.174 seconds

Flat File Creation

**User\_info2.txt**

Ram,Durai,89,US,FL

Sri,Ram,56,US,FL

Raghu,Patel,45,US,FL

Prasad,Kumar,23,CA,AU

Kumar,Singh,55,CA,AU

Loading the data

**hive> load data local inpath '/home/cloudera/Desktop/user\_info2.txt' into table user1;**

Loading data to table default.user1

Table default.user1 stats: [numFiles=1, totalSize=100]

OK

Time taken: 0.966 seconds

Setting of Parameters for dynamic partitioning

**hive> set hive.exec.dynamic.partition=true;**

**hive> set hive.exec.dynamic.partition.mode=nonstrict;**

Retrieving data from the partitioned table

**hive> insert into table par\_user1 partition(country, state) select fname, lname, eid, country, state from user1;**

Query ID = cloudera\_20160928224848\_758afcb0-ab41-4cda-8763-3310e9d7f021

Total jobs = 3

Launching Job 1 out of 3

Number of reduce tasks is set to 0 since there's no reduce operator

Starting Job = job\_1475038007099\_0002, Tracking URL = http://quickstart.cloudera:8088/proxy/application\_1475038007099\_0002/

Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job\_1475038007099\_0002

Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0

2016-09-28 22:57:06,378 Stage-1 map = 0%, reduce = 0%

2016-09-28 22:57:19,580 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 1.46 sec

MapReduce Total cumulative CPU time: 1 seconds 460 msec

Ended Job = job\_1475038007099\_0002

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://quickstart.cloudera:8020/user/hive/warehouse/par\_user1/.hive-staging\_hive\_2016-09-28\_22-56-51\_418\_4721036283115133219-1/-ext-10000

Loading data to table default.par\_user1 partition (country=null, state=null)

Time taken for load dynamic partitions : 349

Loading partition {country=CA, state=AU}

Loading partition {country=US, state=FL}

Time taken for adding to write entity : 3

Partition default.par\_user1{country=CA, state=AU} stats: [numFiles=1, numRows=2, totalSize=31, rawDataSize=29]

Partition default.par\_user1{country=US, state=FL} stats: [numFiles=1, numRows=3, totalSize=39, rawDataSize=36]

MapReduce Jobs Launched:

Stage-Stage-1: Map: 1 Cumulative CPU: 1.46 sec HDFS Read: 3711 HDFS Write: 219 SUCCESS

Total MapReduce CPU Time Spent: 1 seconds 460 msec

OK

Time taken: 31.294 seconds

Retrieving information: Verify the loaded records

**hive> select \* from user1;**

OK

Ram Durai 89 US FL

Sri Ram 56 US FL

Raghu Patel 45 US FL

Prasad Kumar 23 CA AU

Kumar Singh 55 CA AU

Time taken: 0.474 seconds, Fetched: 5 row(s)

**hive> select \* from par\_user1;**

OK

Prasad Kumar 23 CA AU

Kumar Singh 55 CA AU

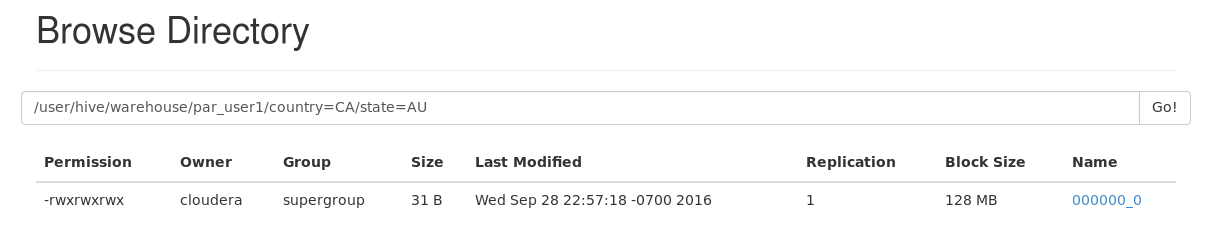
Ram Durai 89 US FL

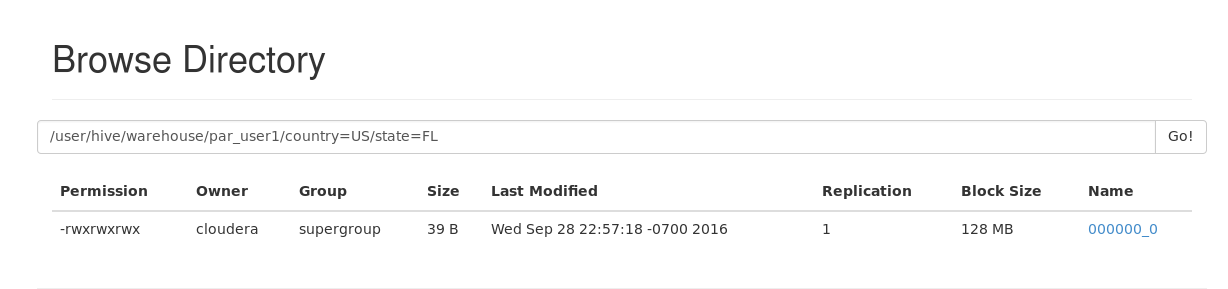
Sri Ram 56 US FL

Raghu Patel 45 US FL

Time taken: 0.244 seconds, Fetched: 5 row(s)

Browse the Directory / Check the result in HDFS

****

****

# **Bucketing using Hive**

Flat File Creation

**empbucket\_old.txt**

1,Ram,34,63000,HR

2,Sriram,32,75000,IT

3,Jana,28,45000,HCLS

4,Diya,22,23000,BNFS

5,sudhir,32,10000,INS

6,raju,24,30000,MF

7,sanjay,22,14000,SE

8,ajay,34,50000,SE

9,soman,21,50000,IT

10,suresh,31,60000,ES

11,john,32,30000,IT

Copying Local File System (LFS) data into HDFS

**$ hadoop fs -put /home/cloudera/Desktop/empbucket\_old.txt /**

**Invoke hive terminal**

**$hive**

**hive> create table empbucketmain (id int,name string,age int,salary float,dept string)**

**>row format delimited fields terminated by ',';**

OK

Time taken: 0.784 seconds

Loading the data

**hive> load data inpath '/empbucket\_old.txt' into table empbucketmain;**

Loading data to table default.empbucketmain

Table default.empbucketmain stats: [numFiles=1, totalSize=224]

OK

Time taken: 0.644 seconds

Table Creation

**hive> create table emp\_bucket (id int,name string,age int,salary float,dept string) clustered by (id) into 5 buckets**

**> row format delimited fields terminated by ',' stored as textfile;**

OK

Time taken: 0.095 seconds

Enforcing Bucketing

**hive> set hive.enforce.bucketing=true;**

Inserting table with bucket

**hive> insert overwrite table emp\_bucket select \* from empbucketmain;**

Query ID = cloudera\_20160927230505\_7e89b0c6-98b7-4fbc-97d9-dc876b7058b8

Total jobs = 1

Launching Job 1 out of 1

Number of reduce tasks determined at compile time: 5

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>

Starting Job = job\_1475038007099\_0001, Tracking URL = http://quickstart.cloudera:8088/proxy/application\_1475038007099\_0001/

Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job\_1475038007099\_0001

Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 5

2016-09-27 23:08:59,071 Stage-1 map = 0%, reduce = 0%

2016-09-27 23:09:08,332 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 1.06 sec

2016-09-27 23:09:47,206 Stage-1 map = 100%, reduce = 67%, Cumulative CPU 4.93 sec

2016-09-27 23:09:52,674 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 9.51 sec

MapReduce Total cumulative CPU time: 9 seconds 510 msec

Ended Job = job\_1475038007099\_0001

Loading data to table default.emp\_bucket

Table default.emp\_bucket stats: [numFiles=5, numRows=11, totalSize=246, rawDataSize=235]

MapReduce Jobs Launched:

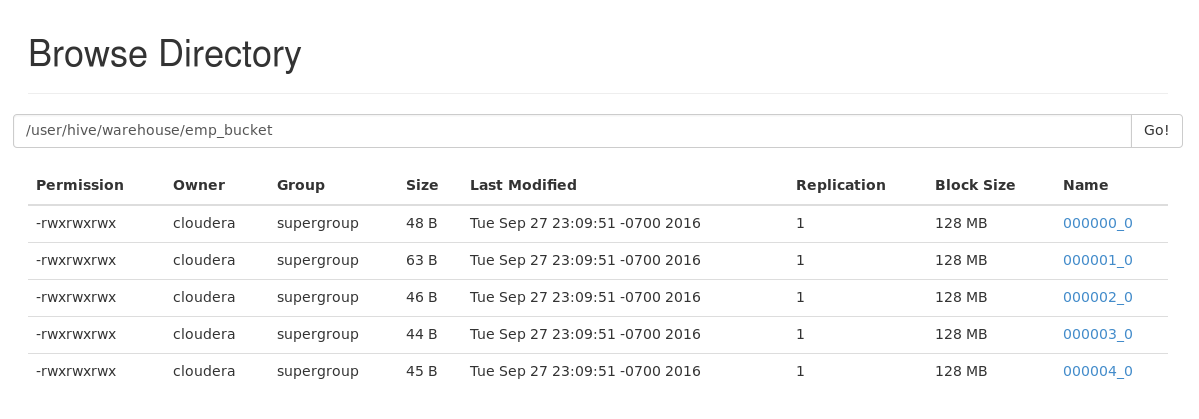
Stage-Stage-1: Map: 1 Reduce: 5 Cumulative CPU: 9.51 sec HDFS Read: 20297 HDFS Write: 616 SUCCESS

Total MapReduce CPU Time Spent: 9 seconds 510 msec

OK

Time taken: 69.862 seconds

Browse the Directory / Check the result in HDFS

****

**Note:**

**[Hash (columns) ] MOD [Number of buckets ]**

# **Complex data type in Hive: Array**

Flat File Creation

**arrayinput.txt**

1,326362$3443$23432$875665$3443$43534$234$342

2,123$323$546$546$5476

3,435$345$678$122$98987

4,234$7234$65242$6272

Invoke Hive Terminal

**$hive**

**Table Creation**

**hive> create table array\_test1 ( id int,all\_nums array<int>) row format delimited fields terminated by ','**

**>collection items terminated by '$' stored as textfile;**

OK

Time taken: 0.215 seconds

Load the data

**hive> load data local inpath '/home/cloudera/Desktop/arrayinput.txt' into table array\_test1;**

Loading data to table default.array\_test1

Table default.array\_test1 stats: [numFiles=1, totalSize=115]

OK

Time taken: 0.648 seconds

Retrieving information: Verify the loaded records

**hive> select id,all\_nums from array\_test1;**

OK

1 [326362,3443,23432,875665,3443,43534,234,342]

2 [123,323,546,546,5476]

3 [435,345,678,122,98987]

4 [234,7234,65242,6272]

Time taken: 0.144 seconds, Fetched: 4 row(s)

**hive> select id,all\_nums[1] from array\_test1;**

OK

1 3443

2 323

3 345

4 7234

Time taken: 0.108 seconds, Fetched: 4 row(s)

**hive> select id,all\_nums[4] from array\_test1;**

OK

1 3443

2 5476

3 98987

4 NULL

Time taken: 0.086 seconds, Fetched: 4 row(s)

# **Complex data type in Hive: Struct**

Flat File Creation

**Weather.txt**

**1,32$65$moderate**

**2,37$78$humid**

**3,43$55$hot**

**4,23$45$cold**

Table Creation

**hive> create table struct\_test ( id int,weather\_reading struct<temp:int,humidity:int,comment:string>)**

**>row format delimited fields terminated by ',' collection items terminated by '$' stored as textfile;**

OK

Time taken: 0.232 seconds

Load the data

**hive> load data local inpath '/home/cloudera/Desktop/weather.txt' into table struct\_test;**

Loading data to table default.struct\_test

Table default.struct\_test stats: [numFiles=1, totalSize=56]

OK

Time taken: 0.424 seconds

Verify data

**hive> select id,weather\_reading from struct\_test;**

OK

1 {"temp":32,"humidity":65,"comment":"moderate"}

2 {"temp":37,"humidity":78,"comment":"humid"}

3 {"temp":43,"humidity":55,"comment":"hot"}

4 {"temp":23,"humidity":45,"comment":"cold"}

Time taken: 0.092 seconds, Fetched: 4 row(s)

**hive> select id,weather\_reading.temp from struct\_test;**

OK

1 32

2 37

3 43

4 23

Time taken: 0.087 seconds, Fetched: 4 row(s)

**hive> select id,weather\_reading.humidity from struct\_test;**

OK

1 65

2 78

3 55

4 45

Time taken: 0.097 seconds, Fetched: 4 row(s)

**hive> select id,weather\_reading.comment from struct\_test;**

OK

1 moderate

2 humid

3 hot

4 cold

Time taken: 0.127 seconds, Fetched: 4 row(s)

# **Complex data type in Hive: Map**

Flat File Creation

**Comments.txt**

1,1@india is great#2@india won icc t20#3@jai hind

2,1@we are awesome#2@i like cricket

3,1@hurray we won#2@what a great match#3@watching cricket all day

4,1@hectic day#3@irctc rocks

Table Creation

**hive> create table map\_test (id int,comments\_map Map<int,string>)**

**> row format delimited fields terminated by ',' collection items terminated by '#'**

**> map keys terminated by '@' stored as textfile;**

OK

Time taken: 0.136 seconds

Load the data

**hive> load data local inpath '/home/cloudera/Desktop/comments.txt' into table map\_test;**

Loading data to table default.map\_test

Table default.map\_test stats: [numFiles=1, totalSize=181]

OK

Time taken: 0.367 seconds

Verify data

**hive> select id,comments\_map from map\_test;**

OK

1 {1:"india is great",2:"india won icc t20",3:"jai hind"}

2 {1:"we are awesome",2:"i like cricket"}

3 {1:"hurray we won",2:"what a great match",3:"watching cricket all day"}

4 {1:"hectic day",3:"irctc rocks"}

Time taken: 0.075 seconds, Fetched: 4 row(s)

**hive> select id,comments\_map[1] from map\_test;**

OK

1 india is great

2 we are awesome

3 hurray we won

4 hectic day

Time taken: 0.087 seconds, Fetched: 4 row(s)

**hive> select id,comments\_map[2] from map\_test;**

OK

1 india won icc t20

2 i like cricket

3 what a great match

4 NULL

Time taken: 0.087 seconds, Fetched: 4 row(s)

# **Hive UDF**

JARS Used

Add the following jars in the build path.

**/usr/lib/hadoop/all hadoop jars**

**And /usr/lib/hive/hive-exec.jar**

Java Code

package com;

import org.apache.hadoop.hive.ql.exec.UDF;

import org.apache.hadoop.io.Text;

public class ReplaceCase extends UDF

{

private Text result=new Text();

public Text evaluate(String str,String str1,String str2)

{

String rep=str.replace(str1,str2);

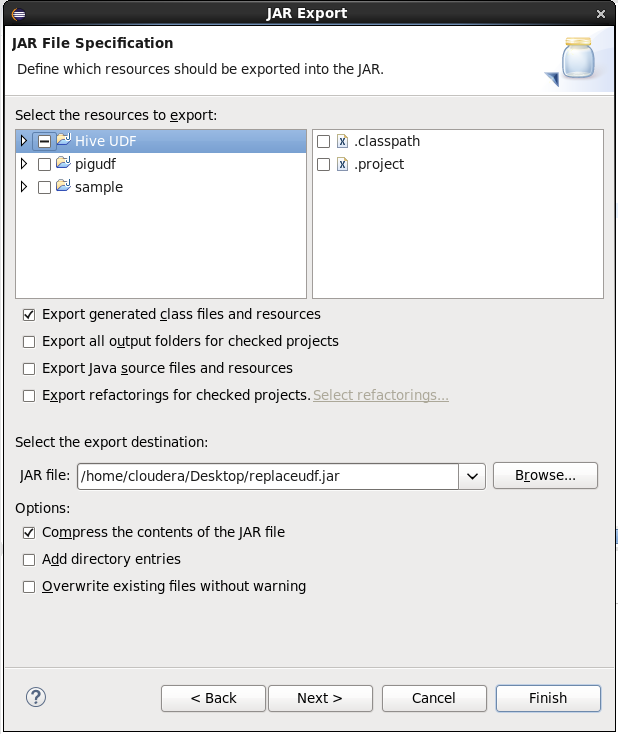
result.set(rep);

return result;

}

}

**Export JAR file**

****

Hive Code

**$ hive**

**hive> add jar /home/cloudera/Desktop/replaceudf.jar;**

Added [/home/cloudera/Desktop/replaceudf.jar] to class path

Added resources: [/home/cloudera/Desktop/replaceudf.jar]

Table Creation

**hive> create table customer (fname STRING,lname STRING) row format delimited fields terminated by '\t'**

**> stored as textfile;**

OK

Time taken: 0.981 seconds

Load the data

**hive> load data local inpath '/home/cloudera/Desktop/names.txt' into table customer;**

Loading data to table default.customer

Table default.customer stats: [numFiles=1, totalSize=8]

OK

Time taken: 1.014 seconds

Creating function

**hive> create temporary function replaceword as 'com.ReplaceCase';**

OK

Time taken: 0.026 seconds

Flat File Creation

**Names.txt**

**sri ram**

Retrieving table

**hive> select replaceword(fname,"sri","raj") from customer;**

OK

raj

Time taken: 0.619 seconds, Fetched: 1 row(s)

# Integration of Pig with HBase

Flat File Creation

**cust\_info.txt**

1,Sriram,IT

2,Ram,LT

3,Jana,RCT

Copying file from LFS to HDFS

**hadoop fs -put /home/cloudera/Desktop/cust\_info.txt /**

Open a hbase terminal and type the following

**[cloudera@quickstart ~]$ hbase shell**

2016-10-05 02:44:55,145 INFO [main] Configuration.deprecation: hadoop.native.lib is deprecated. Instead, use io.native.lib.available

HBase Shell; enter 'help<RETURN>' for list of supported commands.

Type "exit<RETURN>" to leave the HBase Shell

Version 1.2.0-cdh5.7.0, rUnknown, Wed Mar 23 11:39:14 PDT 2016

Table Creation

**hbase(main):001:0> create 'cust\_table','cust\_data'**

0 row(s) in 1.7130 seconds

=> Hbase::Table - cust\_table

Table Scan

**hbase(main):002:0> scan 'cust\_table'**

ROW COLUMN+CELL

0 row(s) in 0.3630 seconds

Open another terminal type the following commands

**[cloudera@quickstart ~]$ export PIG\_CLASSATH=/home/hadoop/HADOOP/hbase-0.98.4-hadoop2/lib/hbase-server-0.98.4-hadoop2:/home/hadoop/HADOOP/hbase-0.98.4-hadoop2/lib/hbase-\*.jar**

**[cloudera@quickstart ~]$ pig**

**grunt> rawd = LOAD '/cust\_info.txt' using PigStorage(',') as (cust\_id:int,cust\_name:chararray,cust\_sector:chararray);**

**grunt> STORE rawd INTO 'hbase://cust\_table' USING org.apache.pig.backend.hadoop.hbase.HBaseStorage('cust\_data:cust\_id,cust\_data:cust\_name,cust\_data:cust\_sector');**

2016-10-05 02:49:30,917 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 0% complete

2016-10-05 02:49:55,369 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 50% complete

2016-10-05 02:50:01,551 [main] INFO org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server

2016-10-05 02:50:02,523 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - mapred.reduce.tasks is deprecated. Instead, use mapreduce.job.reduces

2016-10-05 02:50:02,614 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 100% complete

2016-10-05 02:50:02,619 [main] INFO org.apache.pig.tools.pigstats.SimplePigStats - Script Statistics:

HadoopVersion PigVersion UserId StartedAt FinishedAt Features

2.6.0-cdh5.7.0 0.12.0-cdh5.7.0 cloudera 2016-10-05 02:49:23 2016-10-05 02:50:02 UNKNOWN

Success!

Job Stats (time in seconds):

JobId Maps Reduces MaxMapTime MinMapTIme AvgMapTime MedianMapTime MaxReduceTime MinReduceTime AvgReduceTime MedianReducetime Alias Feature Outputs

job\_1475648951424\_0002 1 0 8 8 8 8 n/a n/a n/a n/a rawd MAP\_ONLY hbase://cust\_table,

Input(s)

Successfully read 3 records (396 bytes) from: "/cust\_info.txt"

Output(s)

Successfully stored 3 records in: "hbase://cust\_table"

Counters:

Total records written : 3

Total bytes written : 0

Spillable Memory Manager spill count : 0

Total bags proactively spilled: 0

Total records proactively spilled: 0

Job DAG:

job\_1475648951424\_0002

**2016-10-05 02:50:02,709 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!**

Check the result in hbase terminal

**hbase(main):003:0> scan 'cust\_table'**

ROW COLUMN+CELL

1 column=cust\_data:cust\_id, timestamp=1475660994141, value=Sriram

1 column=cust\_data:cust\_name, timestamp=1475660994141, value=IT

2 column=cust\_data:cust\_id, timestamp=1475660994151, value=Ram

2 column=cust\_data:cust\_name, timestamp=1475660994151, value =LT

3 column=cust\_data:cust\_id, timestamp=1475660994151, value=Jana

3 column=cust\_data:cust\_name, timestamp=1475660994151, value =RCT

3 row(s) in 0.1130 seconds