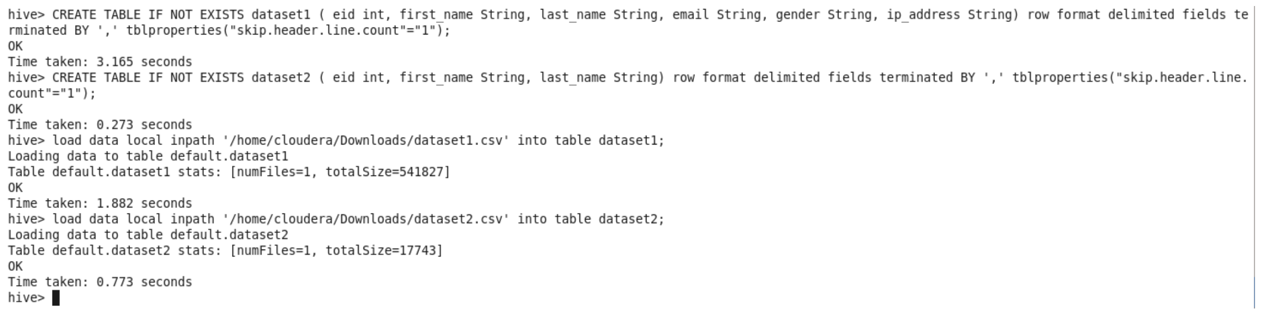
JOIN OPTIMIZATIONS IN HIVE

1. Below are two datasets that are used for our examples.

   dataset1.csv   dataset2.csv

2. Download these files and open hive session and run below queries to create tables and load data.

```
CREATE TABLE IF NOT EXISTS dataset1 ( eid int, first_name String, last_name
String, email String, gender String, ip_address String) row format delimited
fields terminated BY ',' tblproperties("skip.header.line.count"="1");

CREATE TABLE IF NOT EXISTS dataset2 ( eid int, first_name String, last_name
String) row format delimited fields terminated BY ','
tblproperties("skip.header.line.count"="1");

Load data local inpath 'home/cloudera/Downloads/dataset1.csv' into table
dataset1;

Load data local inpath 'home/cloudera/Downloads/dataset2.csv' into table
dataset2;
```



```
hive> CREATE TABLE IF NOT EXISTS dataset1 ( eid int, first_name String, last_name String, email String, gender String, ip_address String) row format delimited fields te
rminated BY ',' tblproperties("skip.header.line.count"="1");
OK
Time taken: 3.165 seconds
hive> CREATE TABLE IF NOT EXISTS dataset2 ( eid int, first_name String, last_name String) row format delimited fields terminated BY ',' tblproperties("skip.header.line.
count"="1");
OK
Time taken: 0.273 seconds
hive> load data local inpath '/home/cloudera/Downloads/dataset1.csv' into table dataset1;
Loading data to table default.dataset1
Table default.dataset1 stats: [numFiles=1, totalSize=541827]
OK
Time taken: 1.882 seconds
hive> load data local inpath '/home/cloudera/Downloads/dataset2.csv' into table dataset2;
Loading data to table default.dataset2
Table default.dataset2 stats: [numFiles=1, totalSize=17743]
OK
Time taken: 0.773 seconds
hive> 
```

Now, let us do normal join first and then do map-side joins later.

Normal join:

1. Run below command in hive:

set hive.auto.convert.join;

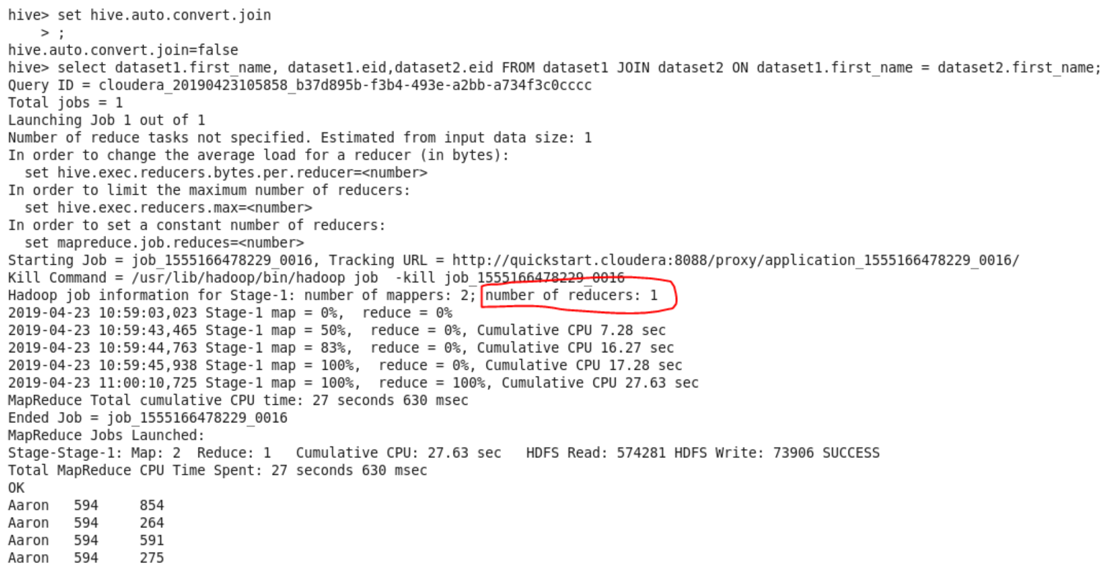
if it shows true, then set it to false for our test.

set hive.auto.convert.join=false;

2. run below query in hive

```
SELECT dataset1.first_name, dataset1.eid,dataset2.eid FROM dataset1 JOIN
dataset2 ON dataset1.first_name = dataset2.first_name;
```

You can observe the plan which show number of reducer is one.



```
hive> set hive.auto.convert.join
    > ;
hive.auto.convert.join=false
hive> select dataset1.first_name, dataset1.eid,dataset2.eid FROM dataset1 JOIN dataset2 ON dataset1.first_name = dataset2.first_name;
Query ID = cloudera_20190423105858_b37d895b-f3b4-493e-a2bb-a734f3c0cccc
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 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>
Starting Job = job_1555166478229_0016, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1555166478229_0016/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1555166478229_0016
Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 1
2019-04-23 10:59:03,023 Stage-1 map = 0%,  reduce = 0%
2019-04-23 10:59:43,465 Stage-1 map = 50%,  reduce = 0%, Cumulative CPU 7.28 sec
2019-04-23 10:59:44,763 Stage-1 map = 83%,  reduce = 0%, Cumulative CPU 16.27 sec
2019-04-23 10:59:45,938 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 17.28 sec
2019-04-23 11:00:10,725 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 27.63 sec
MapReduce Total cumulative CPU time: 27 seconds 630 msec
Ended Job = job_1555166478229_0016
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2  Reduce: 1   Cumulative CPU: 27.63 sec   HDFS Read: 574281 HDFS Write: 73906 SUCCESS
Total MapReduce CPU Time Spent: 27 seconds 630 msec
OK
Aaron   594     854
Aaron   594     264
Aaron   594     591
Aaron   594     275
```

Map Joins

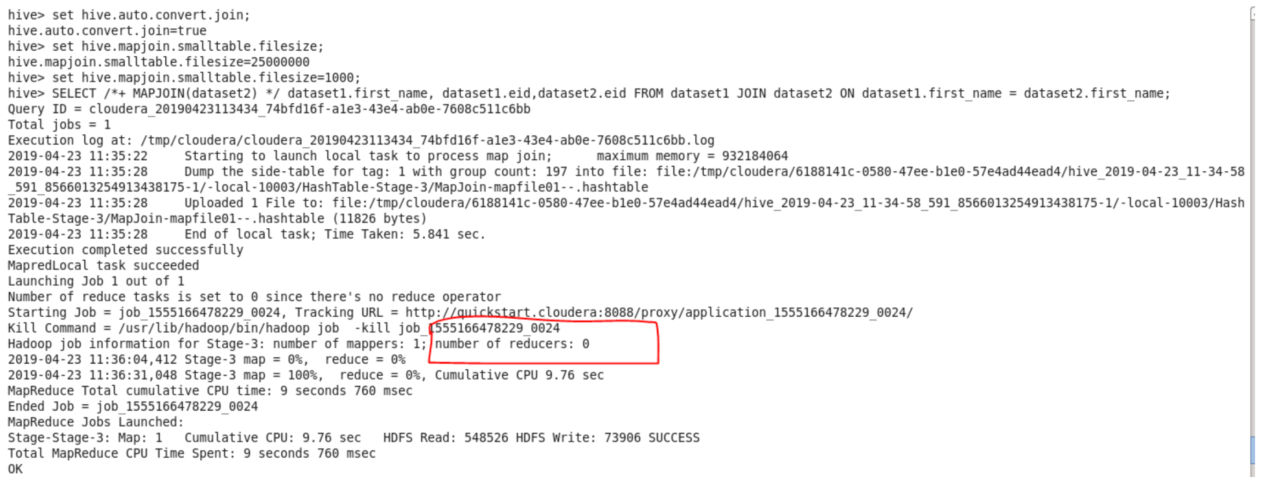
Map join can be performed as below:

1. By setting the following property to true.

```
set hive.auto.convert.join=true
```

2. Run the below query and see the number of reducers set to 0.

```
SELECT /*+ MAPJOIN(dataset2) */ dataset1.first_name,
dataset1.eid,dataset2.eid FROM dataset1 JOIN dataset2 ON dataset1.first_name
= dataset2.first_name;
```



```
hive> set hive.auto.convert.join;
hive.auto.convert.join=true
hive> set hive.mapjoin.smalltable.filesize;
hive.mapjoin.smalltable.filesize=25000000
hive> set hive.mapjoin.smalltable.filesize=1000;
hive> SELECT /*+ MAPJOIN(dataset2) */ dataset1.first_name, dataset1.eid,dataset2.eid FROM dataset1 JOIN dataset2 ON dataset1.first_name = dataset2.first_name;
Query ID = cloudera_20190423113434_74bfd16f-a1e3-43e4-ab0e-7608c511c6bb
Total jobs = 1
Execution log at: /tmp/cloudera/cloudera_20190423113434_74bfd16f-a1e3-43e4-ab0e-7608c511c6bb.log
2019-04-23 11:35:22     Starting to launch local task to process map join;      maximum memory = 932184064
2019-04-23 11:35:28     Dump the side-table for tag: 1 with group count: 197 into file: file:/tmp/cloudera/6188141c-0580-47ee-b1e0-57e4ad44ead4/hive_2019-04-23_11-34-58
_591_8566013254913438175-1/-local-10003/HashTable-Stage-3/MapJoin-mapfile01--.hashtable
2019-04-23 11:35:28     Uploaded 1 File to: file:/tmp/cloudera/6188141c-0580-47ee-b1e0-57e4ad44ead4/hive_2019-04-23_11-34-58_591_8566013254913438175-1/-local-10003/Hash
Table-Stage-3/MapJoin-mapfile01--.hashtable (11826 bytes)
2019-04-23 11:35:28     End of local task; Time Taken: 5.841 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1555166478229_0024, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1555166478229_0024/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1555166478229_0024
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 0
2019-04-23 11:36:04,412 Stage-3 map = 0%,  reduce = 0%
2019-04-23 11:36:31,048 Stage-3 map = 100%,  reduce = 0%, Cumulative CPU 9.76 sec
MapReduce Total cumulative CPU time: 9 seconds 760 msec
Ended Job = job_1555166478229_0024
MapReduce Jobs Launched:
Stage-Stage-3: Map: 1   Cumulative CPU: 9.76 sec   HDFS Read: 548526 HDFS Write: 73906 SUCCESS
Total MapReduce CPU Time Spent: 9 seconds 760 msec
OK
```

NOTE: You can set the size of the small table using below command in hive

```
set hive.mapjoin.smalltable.filesize=100000   //Default is 25MB 25000000
```

Bucket Map Joins

Constraint:

If tables being joined are bucketed on the join columns, and the number of buckets in one table is multiple of the number of buckets in the other table, then buckets can be joined with each other.

Creating bucket tables:

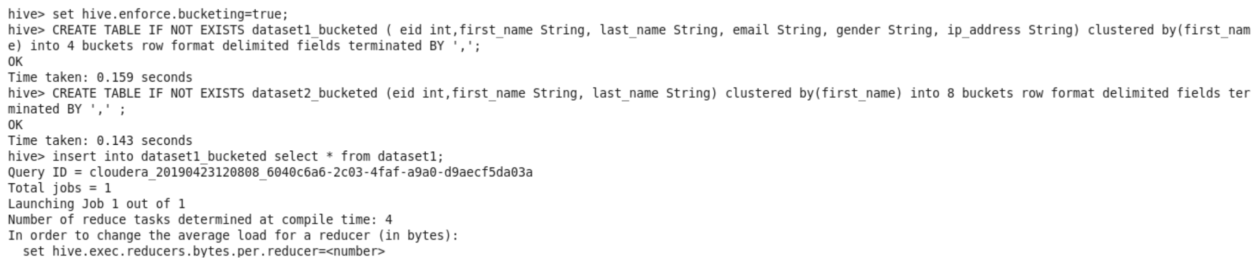
a) Enable bucketing before running below queries:

```
set hive.enforce.bucketing=true;
```

b) Run queries to load data to bucketed tables.

```
CREATE TABLE IF NOT EXISTS dataset1_bucketed ( eid int,first_name String,
last_name String, email String, gender String, ip_address String) clustered
by(first_name) into 4 buckets row format delimited fields terminated BY ',';

insert into dataset1_bucketed select * from dataset1;
```

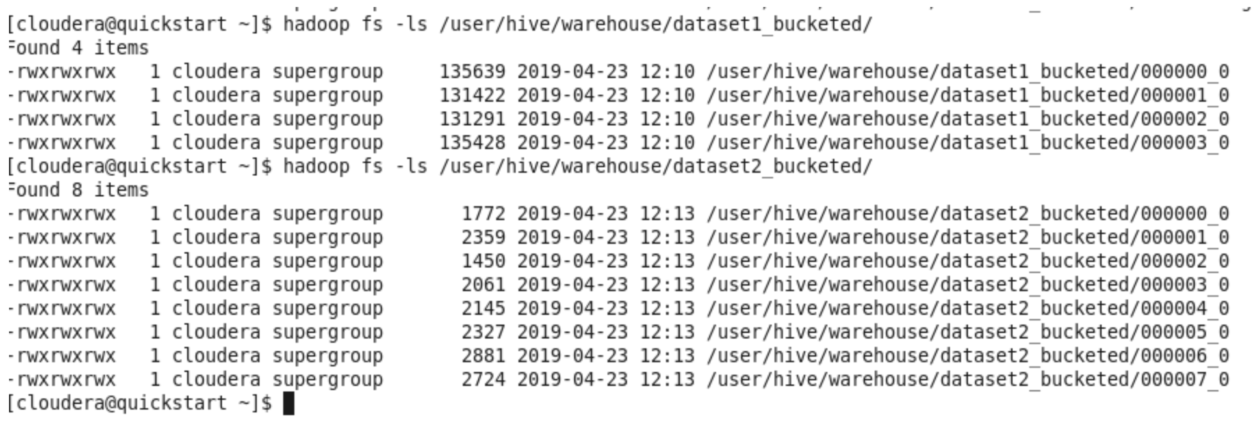
```
CREATE TABLE IF NOT EXISTS dataset2_bucketed (eid int,first_name String,
last_name String) clustered by(first_name) into 8 buckets row format
delimited fields terminated BY ',' ;

insert into dataset2_bucketed select * from dataset2;
```



```
hive> set hive.enforce.bucketing=true;
hive> CREATE TABLE IF NOT EXISTS dataset1_bucketed ( eid int,first_name String, last_name String, email String, gender String, ip_address String) clustered by(first_nam
e) into 4 buckets row format delimited fields terminated BY ',';
OK
Time taken: 0.159 seconds
hive> CREATE TABLE IF NOT EXISTS dataset2_bucketed (eid int,first_name String, last_name String) clustered by(first_name) into 8 buckets row format delimited fields ter
minated BY ',' ;
OK
Time taken: 0.143 seconds
hive> insert into dataset1_bucketed select * from dataset1;
Query ID = cloudera_20190423120808_6040c6a6-2c03-4faf-a9a0-d9aecf5da03a
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 4
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
```

Note: While running insert queries, we can also observe number of reducers is equal to number of buckets.

c) Check in HDFS to ensure bucketed files are created as expected.



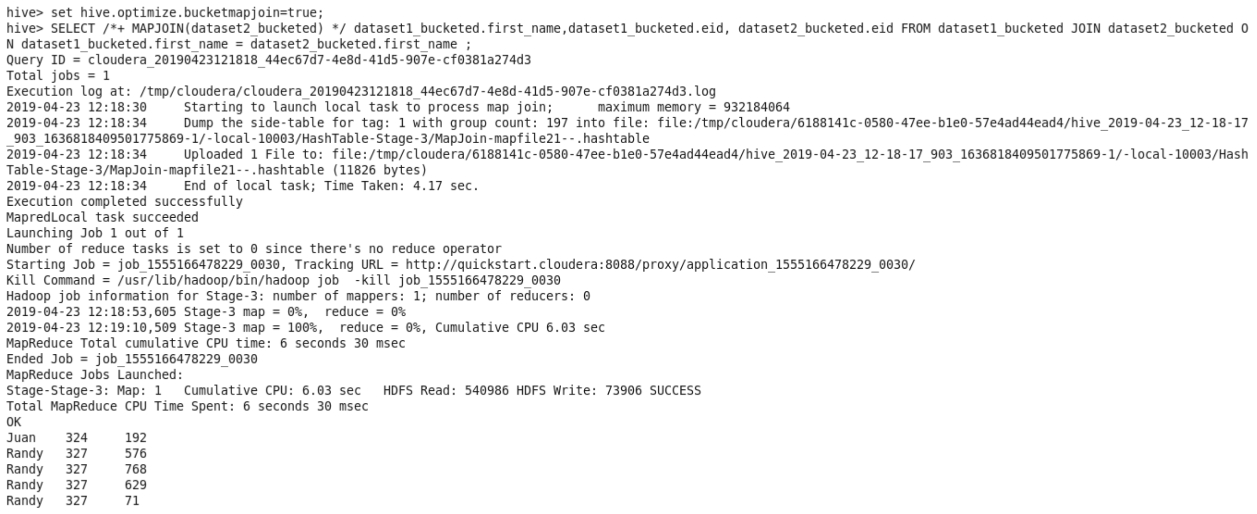
```
[cloudera@quickstart ~]$ hadoop fs -ls /user/hive/warehouse/dataset1_bucketed/
Found 4 items
-rwxrwxrwx   1 cloudera supergroup     135639 2019-04-23 12:10 /user/hive/warehouse/dataset1_bucketed/000000_0
-rwxrwxrwx   1 cloudera supergroup     131422 2019-04-23 12:10 /user/hive/warehouse/dataset1_bucketed/000001_0
-rwxrwxrwx   1 cloudera supergroup     131291 2019-04-23 12:10 /user/hive/warehouse/dataset1_bucketed/000002_0
-rwxrwxrwx   1 cloudera supergroup     135428 2019-04-23 12:10 /user/hive/warehouse/dataset1_bucketed/000003_0
[cloudera@quickstart ~]$ hadoop fs -ls /user/hive/warehouse/dataset2_bucketed/
Found 8 items
-rwxrwxrwx   1 cloudera supergroup       1772 2019-04-23 12:13 /user/hive/warehouse/dataset2_bucketed/000000_0
-rwxrwxrwx   1 cloudera supergroup       2359 2019-04-23 12:13 /user/hive/warehouse/dataset2_bucketed/000001_0
-rwxrwxrwx   1 cloudera supergroup       1450 2019-04-23 12:13 /user/hive/warehouse/dataset2_bucketed/000002_0
-rwxrwxrwx   1 cloudera supergroup       2061 2019-04-23 12:13 /user/hive/warehouse/dataset2_bucketed/000003_0
-rwxrwxrwx   1 cloudera supergroup       2145 2019-04-23 12:13 /user/hive/warehouse/dataset2_bucketed/000004_0
-rwxrwxrwx   1 cloudera supergroup       2327 2019-04-23 12:13 /user/hive/warehouse/dataset2_bucketed/000005_0
-rwxrwxrwx   1 cloudera supergroup       2881 2019-04-23 12:13 /user/hive/warehouse/dataset2_bucketed/000006_0
-rwxrwxrwx   1 cloudera supergroup       2724 2019-04-23 12:13 /user/hive/warehouse/dataset2_bucketed/000007_0
[cloudera@quickstart ~]$
```

d) For performing Bucket-Map join, we need to set this property in the Hive shell.

```
set hive.optimize.bucketmapjoin = true;
```

e) Run below query

```
 SELECT /*+ MAPJOIN(dataset2_bucketed) */
dataset1_bucketed.first_name,dataset1_bucketed.eid, dataset2_bucketed.eid
FROM dataset1_bucketed JOIN dataset2_bucketed ON dataset1_bucketed.first_name
= dataset2_bucketed.first_name ;
```

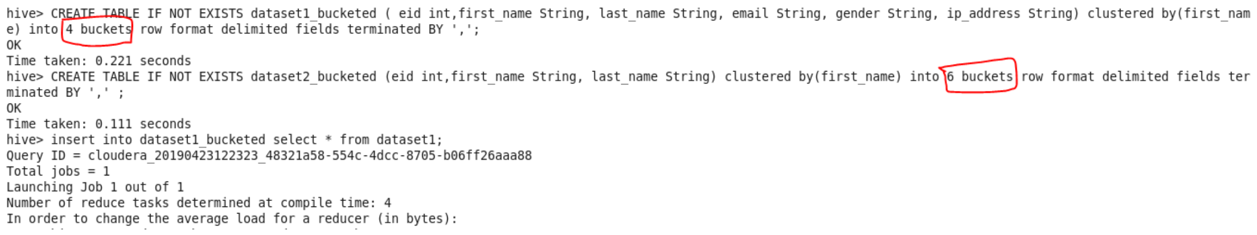


```
hive> set hive.optimize.bucketmapjoin=true;
hive> SELECT /*+ MAPJOIN(dataset2_bucketed) */ dataset1_bucketed.first_name,dataset1_bucketed.eid, dataset2_bucketed.eid FROM dataset1_bucketed JOIN dataset2_bucketed O
N dataset1_bucketed.first_name = dataset2_bucketed.first_name ;
Query ID = cloudera_20190423121818_44ec67d7-4e8d-41d5-907e-cf0381a274d3
Total jobs = 1
Execution log at: /tmp/cloudera/cloudera_20190423121818_44ec67d7-4e8d-41d5-907e-cf0381a274d3.log
2019-04-23 12:18:30     Starting to launch local task to process map join;      maximum memory = 932184064
2019-04-23 12:18:34     Dump the side-table for tag: 1 with group count: 197 into file: file:/tmp/cloudera/6188141c-0580-47ee-b1e0-57e4ad44ead4/hive_2019-04-23_12-18-17
_903_1636818409501775869-1/-local-10003/HashTable-Stage-3/MapJoin-mapfile21--.hashtable
2019-04-23 12:18:34     Uploaded 1 File to: file:/tmp/cloudera/6188141c-0580-47ee-b1e0-57e4ad44ead4/hive_2019-04-23_12-18-17_903_1636818409501775869-1/-local-10003/Hash
Table-Stage-3/MapJoin-mapfile21--.hashtable (11826 bytes)
2019-04-23 12:18:34     End of local task; Time Taken: 4.17 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1555166478229_0030, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1555166478229_0030/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1555166478229_0030
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 0
2019-04-23 12:18:53,605 Stage-3 map = 0%,  reduce = 0%
2019-04-23 12:19:10,509 Stage-3 map = 100%,  reduce = 0%, Cumulative CPU 6.03 sec
MapReduce Total cumulative CPU time: 6 seconds 30 msec
Ended Job = job_1555166478229_0030
MapReduce Jobs Launched:
Stage-Stage-3: Map: 1   Cumulative CPU: 6.03 sec   HDFS Read: 540986 HDFS Write: 73906 SUCCESS
Total MapReduce CPU Time Spent: 6 seconds 30 msec
OK
Juan    324     192
Randy   327     576
Randy   327     768
Randy   327     629
Randy   327     71
```

Time taken to complete this query will be much less than map join we run without buckets.

Let's see what will happen if the buckets of dataset1 are not in multiples of buckets of dataset2.

a) Drop tables and recreate them with new number of buckets.



```
hive> CREATE TABLE IF NOT EXISTS dataset1_bucketed ( eid int,first_name String, last_name String, email String, gender String, ip_address String) clustered by(first_nam
e) into 4 buckets row format delimited fields terminated BY ',';
OK
Time taken: 0.221 seconds
hive> CREATE TABLE IF NOT EXISTS dataset2_bucketed (eid int,first_name String, last_name String) clustered by(first_name) into 6 buckets row format delimited fields ter
minated BY ',' ;
OK
Time taken: 0.111 seconds
hive> insert into dataset1_bucketed select * from dataset1;
Query ID = cloudera_20190423122323_48321a58-554c-4dcc-8705-b06ff26aaa88
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 4
In order to change the average load for a reducer (in bytes):
```

b) Run the query again and observe the log. It won't throw any error however takes more time than bucket map join. If you don't see much time difference, that might be because our datasets are very small.

c) Recreate the dataset1_bucketed with 4 & dataset2_bucketed with 8 buckets.

Sort Merge Bucket Map Join

If the tables being joined are sorted and bucketized on the join columns and have the same number of buckets, a sort-merge join can be performed. The corresponding buckets are joined with each other at the mapper.

Here we have 4 buckets for dataset1 and 8 buckets for dataset2. Now, we will create another table with 4 buckets for dataset2.

For performing the SMB-Map join, we need to set the following properties:

```
Set hive.input.format=org.apache.hadoop.hive.ql.io.BucketizedHiveInputFormat;

set hive.optimize.bucketmapjoin = true;

set hive.optimize.bucketmapjoin.sortedmerge = true;
```

To perform this join, we need to have the data in the bucketed tables sorted by the join column. Now, we will re-insert the data into the bucketed tables by using sorting the records.

```
insert overwrite table dataset1_bucketed select * from dataset1 sort by
first_name;
```

The above command will overwrite the data in the old table and insert the data as per the query.

So now the data in the dataset1_bucketed table is sorted by first_name.

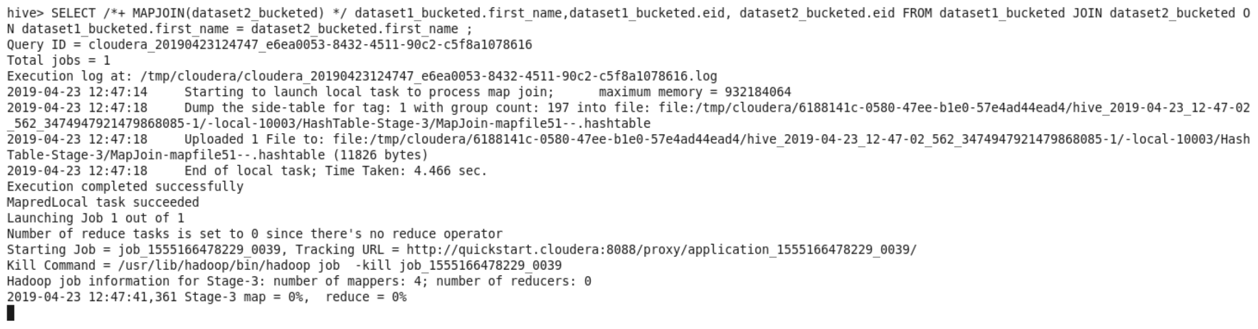
Now, let us perform the SMB-Map join on the two tables with 4 buckets in one table and 8 buckets in one table.

We will now overwrite the data into the dataset2_bucketed table, using the following command:

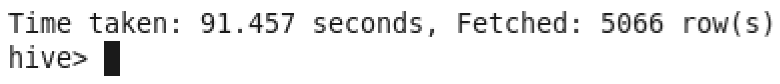
```
insert overwrite table dataset2_bucketed select * from dataset2 sort by
first_name;
```

Now, let us perform the join between tables having 4 buckets and 8 buckets.

```
SELECT /*+ MAPJOIN(dataset2_bucketed) */
dataset1_bucketed.first_name,dataset1_bucketed.eid, dataset2_bucketed.eid
FROM dataset1_bucketed JOIN dataset2_bucketed ON dataset1_bucketed.first_name
= dataset2_bucketed.first_name ;
```



```
hive> SELECT /*+ MAPJOIN(dataset2_bucketed) */ dataset1_bucketed.first_name,dataset1_bucketed.eid, dataset2_bucketed.eid FROM dataset1_bucketed JOIN dataset2_bucketed O
N dataset1_bucketed.first_name = dataset2_bucketed.first_name ;
Query ID = cloudera_20190423124747_e6ea0053-8432-4511-90c2-c5f8a1078616
Total jobs = 1
Execution log at: /tmp/cloudera/cloudera_20190423124747_e6ea0053-8432-4511-90c2-c5f8a1078616.log
2019-04-23 12:47:14     Starting to launch local task to process map join;      maximum memory = 932184064
2019-04-23 12:47:18     Dump the side-table for tag: 1 with group count: 197 into file: file:/tmp/cloudera/6188141c-0580-47ee-b1e0-57e4ad44ead4/hive_2019-04-23_12-47-02
_562_3474947921479868085-1/-local-10003/HashTable-Stage-3/MapJoin-mapfile51--.hashtable
2019-04-23 12:47:18     Uploaded 1 File to: file:/tmp/cloudera/6188141c-0580-47ee-b1e0-57e4ad44ead4/hive_2019-04-23_12-47-02_562_3474947921479868085-1/-local-10003/Hash
Table-Stage-3/MapJoin-mapfile51--.hashtable (11826 bytes)
2019-04-23 12:47:18     End of local task; Time Taken: 4.466 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1555166478229_0039, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1555166478229_0039/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1555166478229_0039
Hadoop job information for Stage-3: number of mappers: 4; number of reducers: 0
2019-04-23 12:47:41,361 Stage-3 map = 0%,  reduce = 0%
```

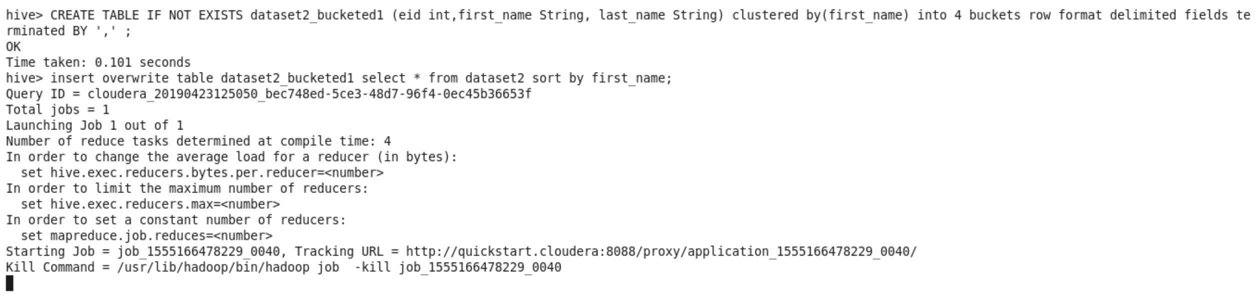


```
Time taken: 91.457 seconds, Fetched: 5066 row(s)
hive>
```

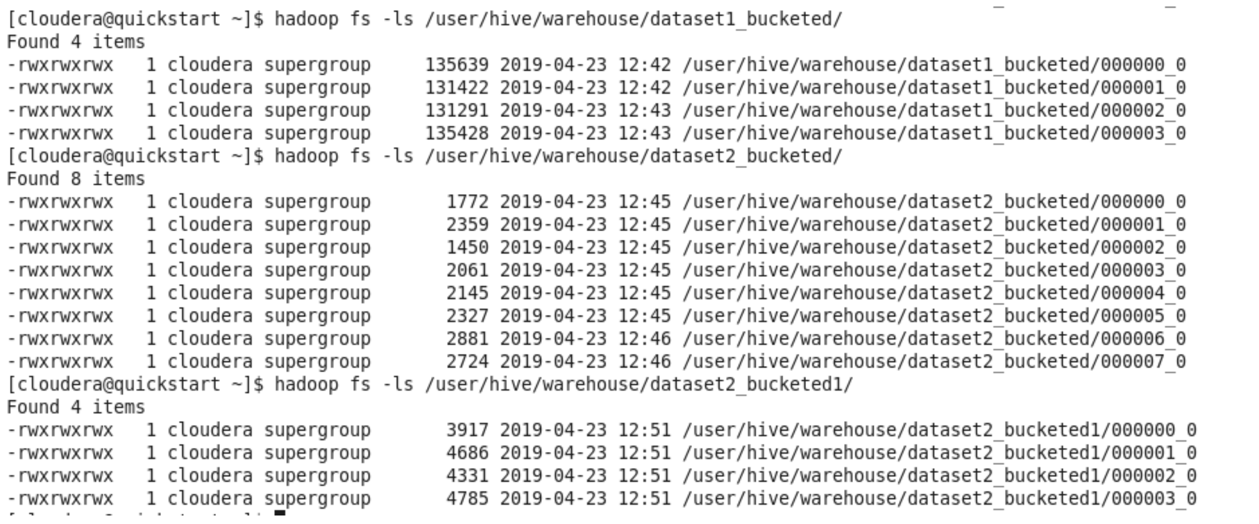
To perform SMB-Map join, we need to have the same number of buckets in both the tables with the bucketed column sorted.

Now, we will create another table for dataset2 having 4 buckets and will insert the data that is sorted by first_name.

```
CREATE TABLE IF NOT EXISTS dataset2_bucketed1 (eid int,first_name String,
last_name String) clustered by(first_name) into 4 buckets row format
delimited fields terminated BY ',' ;

insert overwrite table dataset2_bucketed1 select * from dataset2 sort by
first_name;
```



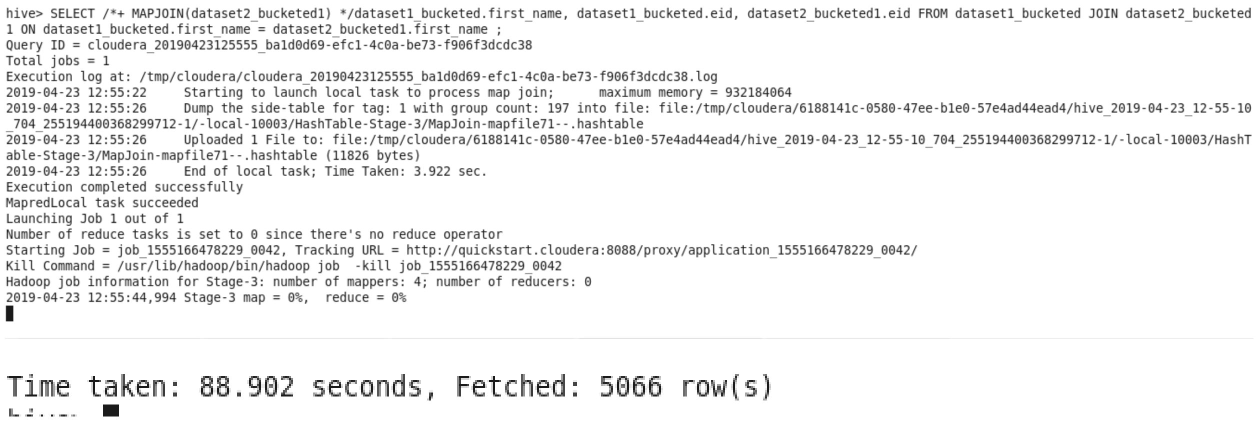
```
hive> CREATE TABLE IF NOT EXISTS dataset2_bucketed1 (eid int,first_name String, last_name String) clustered by(first_name) into 4 buckets row format delimited fields te
rminated BY ',' ;
OK
Time taken: 0.101 seconds
hive> insert overwrite table dataset2_bucketed1 select * from dataset2 sort by first_name;
Query ID = cloudera_20190423125050_bec748ed-5ce3-48d7-96f4-0ec45b36653f
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 4
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_1555166478229_0040, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1555166478229_0040/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1555166478229_0040
```



```
[cloudera@quickstart ~]$ hadoop fs -ls /user/hive/warehouse/dataset1_bucketed/
Found 4 items
-rwxrwxrwx   1 cloudera supergroup     135639 2019-04-23 12:42 /user/hive/warehouse/dataset1_bucketed/000000_0
-rwxrwxrwx   1 cloudera supergroup     131422 2019-04-23 12:42 /user/hive/warehouse/dataset1_bucketed/000001_0
-rwxrwxrwx   1 cloudera supergroup     131291 2019-04-23 12:43 /user/hive/warehouse/dataset1_bucketed/000002_0
-rwxrwxrwx   1 cloudera supergroup     135428 2019-04-23 12:43 /user/hive/warehouse/dataset1_bucketed/000003_0
[cloudera@quickstart ~]$ hadoop fs -ls /user/hive/warehouse/dataset2_bucketed/
Found 8 items
-rwxrwxrwx   1 cloudera supergroup       1772 2019-04-23 12:45 /user/hive/warehouse/dataset2_bucketed/000000_0
-rwxrwxrwx   1 cloudera supergroup       2359 2019-04-23 12:45 /user/hive/warehouse/dataset2_bucketed/000001_0
-rwxrwxrwx   1 cloudera supergroup       1450 2019-04-23 12:45 /user/hive/warehouse/dataset2_bucketed/000002_0
-rwxrwxrwx   1 cloudera supergroup       2061 2019-04-23 12:45 /user/hive/warehouse/dataset2_bucketed/000003_0
-rwxrwxrwx   1 cloudera supergroup       2145 2019-04-23 12:45 /user/hive/warehouse/dataset2_bucketed/000004_0
-rwxrwxrwx   1 cloudera supergroup       2327 2019-04-23 12:45 /user/hive/warehouse/dataset2_bucketed/000005_0
-rwxrwxrwx   1 cloudera supergroup       2881 2019-04-23 12:46 /user/hive/warehouse/dataset2_bucketed/000006_0
-rwxrwxrwx   1 cloudera supergroup       2724 2019-04-23 12:46 /user/hive/warehouse/dataset2_bucketed/000007_0
[cloudera@quickstart ~]$ hadoop fs -ls /user/hive/warehouse/dataset2_bucketed1/
Found 4 items
-rwxrwxrwx   1 cloudera supergroup       3917 2019-04-23 12:51 /user/hive/warehouse/dataset2_bucketed1/000000_0
-rwxrwxrwx   1 cloudera supergroup       4686 2019-04-23 12:51 /user/hive/warehouse/dataset2_bucketed1/000001_0
-rwxrwxrwx   1 cloudera supergroup       4331 2019-04-23 12:51 /user/hive/warehouse/dataset2_bucketed1/000002_0
-rwxrwxrwx   1 cloudera supergroup       4785 2019-04-23 12:51 /user/hive/warehouse/dataset2_bucketed1/000003_0
```

Now, we have two tables with 4 buckets and the joined column sorted. Let us perform the join query again.

```
SELECT /*+ MAPJOIN(dataset2_bucketed1) */dataset1_bucketed.first_name,
dataset1_bucketed.eid, dataset2_bucketed1.eid FROM dataset1_bucketed JOIN
dataset2_bucketed1 ON dataset1_bucketed.first_name =
dataset2_bucketed1.first_name ;
```



```
hive> SELECT /*+ MAPJOIN(dataset2_bucketed1) */dataset1_bucketed.first_name, dataset1_bucketed.eid, dataset2_bucketed1.eid FROM dataset1_bucketed JOIN dataset2_bucketed
1 ON dataset1_bucketed.first_name = dataset2_bucketed1.first_name ;
Query ID = cloudera_20190423125555_ba1d0d69-efc1-4c0a-be73-f906f3dcdc38
Total jobs = 1
Execution log at: /tmp/cloudera/cloudera_20190423125555_ba1d0d69-efc1-4c0a-be73-f906f3dcdc38.log
2019-04-23 12:55:22     Starting to launch local task to process map join;      maximum memory = 932184064
2019-04-23 12:55:26     Dump the side-table for tag: 1 with group count: 197 into file: file:/tmp/cloudera/6188141c-0580-47ee-b1e0-57e4ad44ead4/hive_2019-04-23_12-55-10
_704_255194400368299712-1/-local-10003/HashTable-Stage-3/MapJoin-mapfile71--.hashtable
2019-04-23 12:55:26     Uploaded 1 File to: file:/tmp/cloudera/6188141c-0580-47ee-b1e0-57e4ad44ead4/hive_2019-04-23_12-55-10_704_255194400368299712-1/-local-10003/HashT
able-Stage-3/MapJoin-mapfile71--.hashtable (11826 bytes)
2019-04-23 12:55:26     End of local task; Time Taken: 3.922 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1555166478229_0042, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1555166478229_0042/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1555166478229_0042
Hadoop job information for Stage-3: number of mappers: 4; number of reducers: 0
2019-04-23 12:55:44,994 Stage-3 map = 0%,  reduce = 0%

Time taken: 88.902 seconds, Fetched: 5066 row(s)
```

We can observe time taken by SMB map join is less than simple bucket join.

Questions to Sumith:

Setting size of the small file to very low size doesn’t caused any exception.

Giving hint alone is not working without setting hive.auto.convert.join = true.