Software Report

Group 33
Andreas Petrou, Marios Vouryias, Amar Aziz

March 23, 2022

1 Introduction

This report describes how to run our **MapReduce** queries and compares their output with the output obtained by running the corresponding queries using **HiveQL**. Both methods have been given the same arguments for a fair comparison.

2 MapReduce Queries

2.1 Query 1.a

```
25      1.0184730687499995E9
91      1.01806099288E9
50      1.0180049995599998E9
92      1.0173554239299997E9
14      1.0164983236999998E9
58      1.0159892535700002E9
32      1.0152762881700002E9
82      1.0144282928900005E9
8       1.0141363358600006E9
94      1.0138409312699999E9
```

Figure 1: Results of Query 1.a using MapReduce

Figure 1 illustrates the results obtained by requesting the top 10 stores with the highest revenue in descending order of *ss_net_paid*. Only records that had *ss_sold_date_sk* between 2451392 and 2451894 were considered. To get the above results, first navigate to the folder where the submitted .java and .jar files are located and run the following command:

hadoop jar querya.jar QueryA 10 2451392 2451894 input/40G/store_sales output/queryA

The first argument is the number of records to be displayed, while the second and third arguments are the start and end dates, respectively. The last two arguments are the input and output file locations. It should be noted that in our case, the input file path for the 40G *store_sales* dataset is **input/40G/store_sales**, which was achieved by the following commands:

hdfs dfs -mkdir /user/$USER/input/40G
hdfs dfs -put /dcs/cs346/tpcds/40G/store_sales.dat input/40G/store_sales

The output file path is up to the user. In our case, it is **output/queryA**.

Finally, to access the output of the MapReduce program and view the actual results, as above, the following command is used:

hdfs dfs -cat output/queryA/out6/part-r-00000

The output of the first job is stored in **output/queryA/out5** while the output of the second job, which is the one we are interested in, is stored in **output/queryA/out6**.

Total Runtime of query: 5 minutes 35 seconds

2.2 Query 1.b

```
8270    74627
47438   73449
25729   73290
770     73144
6151    73123
6031    72981
49022   72909
18254   72726
15991   72707
44312   72656
```

Figure 2: Results of Query 1.b using MapReduce

Figure 2 portrays the results obtained by requesting the top 10 items with the highest quantity (in total) sold in descending order of *ss_quantity*. Only records that had *ss_sold_date_sk* between 2451392 and 2451894 were considered. To get the above results, first navigate to the folder where the submitted .java and .jar files are located and run the following command:

hadoop jar queryb.jar QueryB 10 2451392 2451894 input/40G/store_sales output/queryB

The first argument is the number of records to be displayed, while the second and third arguments are the start and end dates, respectively. The last two arguments are the input and output file locations. It should be noted that in our case, the input file path for the 40G *store_sales* dataset is **input/40G/store_sales**, which was achieved by the following commands:

hdfs dfs -mkdir /user/$USER/input/40G
hdfs dfs -put /dcs/cs346/tpcds/40G/store_sales.dat input/40G/store_sales

The output file path is up to the user. In our case, it is **output/queryB**.

Finally, to access the output of the MapReduce program and view the actual results, as above, the following command is used:

hdfs dfs -cat output/queryB/out6/part-r-00000

The output of the first job is stored in **output/queryB/out5** while the output of the second job, which is the one we are interested in, is stored in **output/queryB/out6**.

Total Runtime of query: 5 minutes 10 seconds

2.3 Query 1.c

```
2451546 2.6901120769E8
2451522 2.1829953532E8
2451544 2.1689601327999997E8
2451537 2.1561832877999994E8
2451521 2.1539236946000004E8
2451533 2.1477279572999984E8
2451532 2.1461869395999995E8
2451851 2.1395338407E8
2451891 2.1391831729E8
2451880 2.1384990861000007E8
```

Figure 3: Results of Query 1.c using MapReduce

Figure 3 shows the results obtained by requesting the top 10 days with the highest total value of *ss_net_paid_inc_tax* in descending order of *ss_net_paid_inc_tax*. Only records that had *ss_sold_date_sk* between 2451392 and 2451894 were considered. To get the above results, first navigate to the folder where the submitted .java and .jar files are located and run the following command:

hadoop jar queryc.jar QueryC 10 2451392 2451894 input/40G/store_sales output/queryC

The first argument is the number of records to be displayed, while the second and third arguments are the start and end dates, respectively. The last two arguments are the input and output file locations. It should be noted that in our case, the input file path for the 40G *store_sales* dataset is **input/40G/store_sales**, which was achieved by the following commands:

hdfs dfs -mkdir /user/$USER/input/40G
hdfs dfs -put /dcs/cs346/tpcds/40G/store_sales.dat input/40G/store_sales

The output file path is up to the user. In our case, it is **output/queryC**.

Finally, to access the output of the MapReduce program and view the actual results, as above, the following command is used:

hdfs dfs -cat output/queryC/out6/part-r-00000

The output of the first job is stored in **output/queryC/out5** while the output of the second job, which is the one we are interested in, is stored in **output/queryC/out6**.

Total Runtime of query: 5 minutes 40 seconds

2.4 Query 2

34	2.0148107721599538E9	9810608
85	2.0168838838699675E9	9779755
94	2.0261370851100123E9	9599785
46	2.01198854285001E9	9589409
62	2.0013569727699218E9	9342076
4	2.0281459483400378E9	9341467
10	2.012362578280094E9	9294113
49	2.0203952882399514E9	9206875
43	2.0214605305900266E9	9059442
106	2.0151720475700278E9	8984077

Figure 4: Results of Query 2 using MapReduce

Figure 4 shows the results obtained by requesting the top 10 stores (and their corresponding total *ss_net_paid*) with the highest floor space in descending order of *s_floor_space*, then descending order of *ss_net_paid*. Only records that had *ss_sold_date_sk* between 2451146 and 2452268 were considered. To get the above results, first navigate to the folder where the submitted .java and .jar files are located and run the following command:

hadoop jar query2.jar Query2 10 2451146 2452268 input/40G/store_sales input/40G/store output/query2

The first argument is the number of records to be displayed, while the second and third arguments are the start and end dates, respectively. The last three arguments are the two input file and output file locations. It should be noted that in our case, the input file path for the 40G *store_sales* dataset is **input/40G/store_sales**, which was achieved by the following commands:

hdfs dfs -put /dcs/cs346/tpcds/40G/store_sales.dat input/40G/store_sales

and the input file path for the 40G *store* dataset is **input/40G/store**, which was achieved by the following command:

hdfs dfs -mkdir /user/$USER/input/40G
hdfs dfs -put /dcs/cs346/tpcds/40G/store.dat input/40G/store

The output file path is up to the user. In our case, it is **output/query2**.

Finally, to access the output of the MapReduce program and view the actual results, as above, the following command is used:

hdfs dfs -cat output/query2/out6/part-r-00000

The output of the first job is stored in **output/query2/out5** while the output of the second job, which is the one we are interested in, is stored in **output/query2/out6**.

Total Runtime of query: 5 minutes 40 seconds

3 Hive Queries

3.1 Query 1.a

```
SELECT ss_store_sk, SUM(ss_net_paid) AS revenue
FROM store_sales_40g
WHERE ss_sold_date_sk_ss_sold_time_sk >= 2451392
  AND ss_sold_date_sk_ss_sold_time_sk <= 2451894
GROUP BY ss_store_sk
ORDER BY revenue DESC
LIMIT 10;
```

Parameters:

- Top Records: 10
- Start Date: 2451392
- End Date: 2451894

```
Total MapReduce CPU Time Spent: 8 minutes 38 seconds 740 msec
OK
+--------------+-----------------------+
| ss_store_sk  |        revenue        |
+--------------+-----------------------+
| 25           | 1.0184730687499993E9  |
| 91           | 1.0180609928800004E9  |
| 50           | 1.0180049995599997E9  |
| 92           | 1.0173554239299997E9  |
| 14           | 1.0164983237000002E9  |
| 58           | 1.0159892535700008E9  |
| 32           | 1.0152762881699997E9  |
| 82           | 1.0144282928899997E9  |
| 8            | 1.0141363358600001E9  |
| 94           | 1.0138409312699999E9  |
+--------------+-----------------------+
10 rows selected (433.462 seconds)
```

Figure 5: Results of Query 1.a using HiveQL

3.2 Query 1.b

```
SELECT ss_item_sk, SUM(ss_quantity) AS number_of_sales
FROM store_sales_40g
WHERE ss_sold_date_sk_ss_sold_time_sk >= 2451392
  AND ss_sold_date_sk_ss_sold_time_sk <= 2451894
GROUP BY ss_item_sk
ORDER BY number_of_sales DESC
LIMIT 10;
```

Parameters:

- Top Records: 10
- Start Date: 2451392
- End Date: 2451894

```
Total MapReduce CPU Time Spent: 10 minutes 45 seconds 170 msec
OK
+-------------+-----------------+
| ss_item_sk  | number_of_sales  |
+-------------+-----------------+
| 8270        | 74627.0         |
| 47438       | 73449.0         |
| 25729       | 73290.0         |
| 770         | 73144.0         |
| 6151        | 73123.0         |
| 6031        | 72981.0         |
| 49022       | 72909.0         |
| 18254       | 72726.0         |
| 15991       | 72707.0         |
| 44312       | 72656.0         |
+-------------+-----------------+
10 rows selected (417.2 seconds)
```

Figure 6: Results of Query 1.b using HiveQL

3.3 Query 1.c

```
SELECT ss_sold_date_sk_ss_sold_time_sk, SUM(ss_net_paid_inc_tax) AS ss_net_paid_inc_tax
FROM store_sales_40g
WHERE ss_sold_date_sk_ss_sold_time_sk >= 2451392
  AND ss_sold_date_sk_ss_sold_time_sk <= 2451894
GROUP BY ss_sold_date_sk_ss_sold_time_sk
ORDER BY ss_net_paid_inc_tax DESC
LIMIT 10;
```

Parameters:

- Top Records: 10
- Start Date: 2451392
- End Date: 2451894

```
Total MapReduce CPU Time Spent: 9 minutes 9 seconds 160 msec
OK
+----------------------------------+-----------------------+
| ss_sold_date_sk_ss_sold_time_sk  |  ss_net_paid_inc_tax  |
+----------------------------------+-----------------------+
| 2451546                          | 2.6901120769E8        |
| 2451522                          | 2.1829953532000005E8  |
| 2451544                          | 2.1689601328000006E8  |
| 2451537                          | 2.156183287799999E8   |
| 2451521                          | 2.1539236945999998E8  |
| 2451533                          | 2.1477279573000002E8  |
| 2451532                          | 2.146186939599999E8   |
| 2451851                          | 2.1395338406999996E8  |
| 2451891                          | 2.1391831729E8        |
| 2451880                          | 2.1384990861000004E8  |
+----------------------------------+-----------------------+
10 rows selected (377.235 seconds)
```

Figure 7: Results of Query 1.c using HiveQL

3.4 Query 2

```
SELECT ss.ss_store_sk, SUM(ss.ss_net_paid) AS total, s.s_floor_space
FROM store_sales_40g AS ss
INNER JOIN store_40g AS s ON ss.ss_store_sk = s.s_store_sk
WHERE ss.ss_sold_date_sk_ss_sold_time_sk >= 2451146
  AND ss.ss_sold_date_sk_ss_sold_time_sk <= 2452268
GROUP BY ss.ss_store_sk, s_floor_space
ORDER BY s.s_floor_space DESC, total DESC
LIMIT 10;
```

Parameters:

- Top Records: 10
- Start Date: 2451392
- End Date: 2451894

```
Total MapReduce CPU Time Spent: 10 minutes 32 seconds 630 msec
OK
+-----------------+----------------------+------------------+
| ss.ss_store_sk  |        total         | s.s_floor_space  |
+-----------------+----------------------+------------------+
| 34              | 2.0148107721599996E9 | 9810608          |
| 85              | 2.0168838838699985E9 | 9779755          |
| 94              | 2.0261370851099997E9 | 9599785          |
| 46              | 2.011988542850002E9  | 9589409          |
| 62              | 2.0013569727700012E9 | 9342076          |
| 4               | 2.0281459483399994E9 | 9341467          |
| 10              | 2.0123625782800014E9 | 9294113          |
| 49              | 2.0203952882399998E9 | 9206875          |
| 43              | 2.0214605305899987E9 | 9059442          |
| 106             | 2.0151720475699997E9 | 8984077          |
+-----------------+----------------------+------------------+
10 rows selected (548.092 seconds)
```

Figure 8: Results of Query 2 using HiveQL

4 Creating External Tables in Hive

External tables were created for both *store_sales* and *store* 40G datasets to be able to query them. This was achieved by the following:

4.1 store_sales_40g

```
CREATE EXTERNAL TABLE store_sales_40g(
    ss_sold_date_sk_ss_sold_time_sk int,
    ss_sold_time int,
    ss_item_sk int,
    ss_customer_sk int,
    ss_cdemo_sk int,
    ss_hdemo_sk int,
    ss_addr_sk int,
    ss_store_sk int,
    ss_promo_sk int,
    ss_ticket_number long,
    ss_quantity int,
    ss_wholesale_cost double,
    ss_list_price double,
    ss_sales_price double,
    ss_ext_discount_amt double,
    ss_ex_sales_price double,
    ss_ext_wholesale_cost double,
    ss_ext_list_price double,
    ss_ext_tax double,
    ss_coupon_amt double,
    ss_net_paid double,
    ss_net_paid_inc_tax double,
    ss_net_profit double
)
ROW FORMAT DELIMITED
 FIELDS TERMINATED BY '|'
LOCATION '/user/cs346id33/input/40G/store_sales/';
```

4.2 store_40g

```
CREATE EXTERNAL TABLE store_40g(
    s_store_sk int,
    s_store_id string,
    s_rec_start_date string,
    s_rec_end_date string,
    s_closed_date_sk string,
    s_store_name string,
    s_number_employees int,
    s_floor_space int,
    s_hours string,
    s_manager string,
    s_market_id int,
    s_geography_class string,
    s_market_desc string,
    s_market_manager string,
    s_division_id int,
    s_division_name string,
    s_company_id string,
    s_company_name string,
    s_street_number int
)
ROW FORMAT DELIMITED
 FIELDS TERMINATED BY '|'
LOCATION '/user/cs346id33/input/40G/store/';
```