CS561 - Project 2

Group 3

Aniket Prafulla Kadam

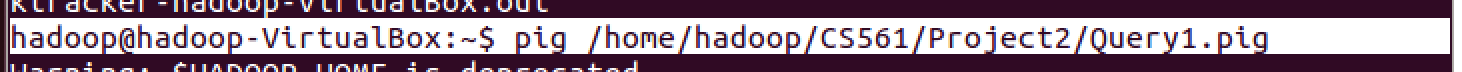
Xinyu Dai

Documentation

**Query1**

The query 1 needs data only from the Transaction.txt file. After reading the dataset from the HDFS, we group the dataset by the custID. For each group key CustomerID, we compute the aggregation count of the transactions as NumTransactions and the aggregation sum of the field of TransTotal as TotalSum. Then we store that output into file on the HDFS.

The running command line for running the pig query is as follows:



Query 1:

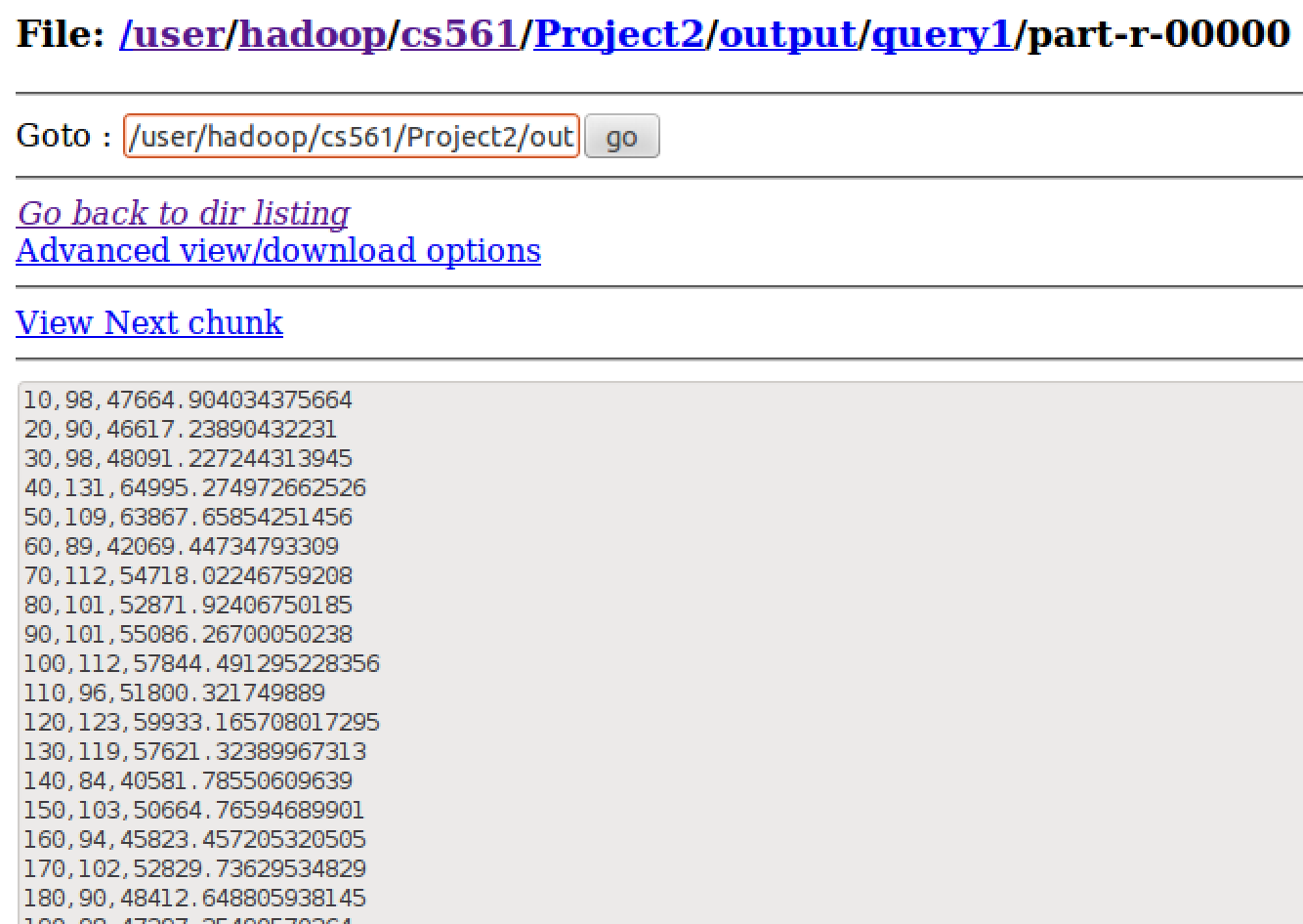
-trans = LOAD '/user/hadoop/cs561/dataset/Transaction.txt' USING PigStorage(',') AS (transID:int, custID:int, transTotal:double, transNumItems:int, transDesc: chararray);

-A = GROUP trans by custID parallel 10;

-B = FOREACH A generate group, COUNT(trans.transTotal) as NumTransactions, SUM(trans.transTotal) as TotalSum;

-store B into '/user/hadoop/cs561/Project2/output/query1' USING PigStorage(',');

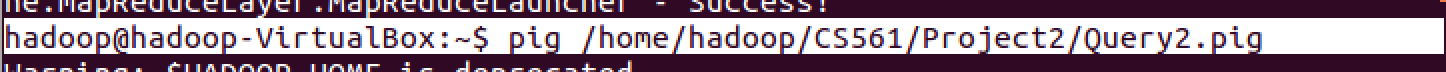
The output of Pig Query 1(snapshot: CustomerID, NumTransactions, TotalSum) is as same as the task3.2 in the Project 1. The total running time is about 2mins, 48sec which is much more slower than the processing time(45sec) in java.



**Query2**

The query 2 needs to join the Customer.txt file and Transaction.txt file based on the custID. After reading the Transaction.txt file from the HDFS, we select the three variables for each row: custID, transTotal and transNumItems which are useful for the generation of NumOfTransaction, TotalSum and MinItems. After selection, we group the selected-dataset by custID and for each group key CustomerID we aggregate the count of transactions and the sum of the field of TransTotal and get the minimum number of items in transactions done by the customer. Then we read the Customer.txt file from the HDFS and generate the custID, name and salary for each row. After that we join the aggregated transaction info and the customer info based on the custID and store the output into file on the HDFS(snapshot in the following figure: CustomerID, Name, Salary, NumOf Transactions, TotalSum, MinItems).

The running command line for running the pig query is as follows:



Query 2:

-trans = LOAD '/user/hadoop/cs561/dataset/Transaction.txt' USING PigStorage(',') AS (transID:int, custID:int, transTotal:double, transNumItems:int, transDesc: chararray);

-A = FOREACH trans generate custID, transTotal, transNumItems;

-B = group A by custID;

-C = foreach B generate group, COUNT(A.transTotal) as NumOfTransaction, SUM(A.transTotal) as TotalSum, MIN(A.transNumItems) as MinItems;

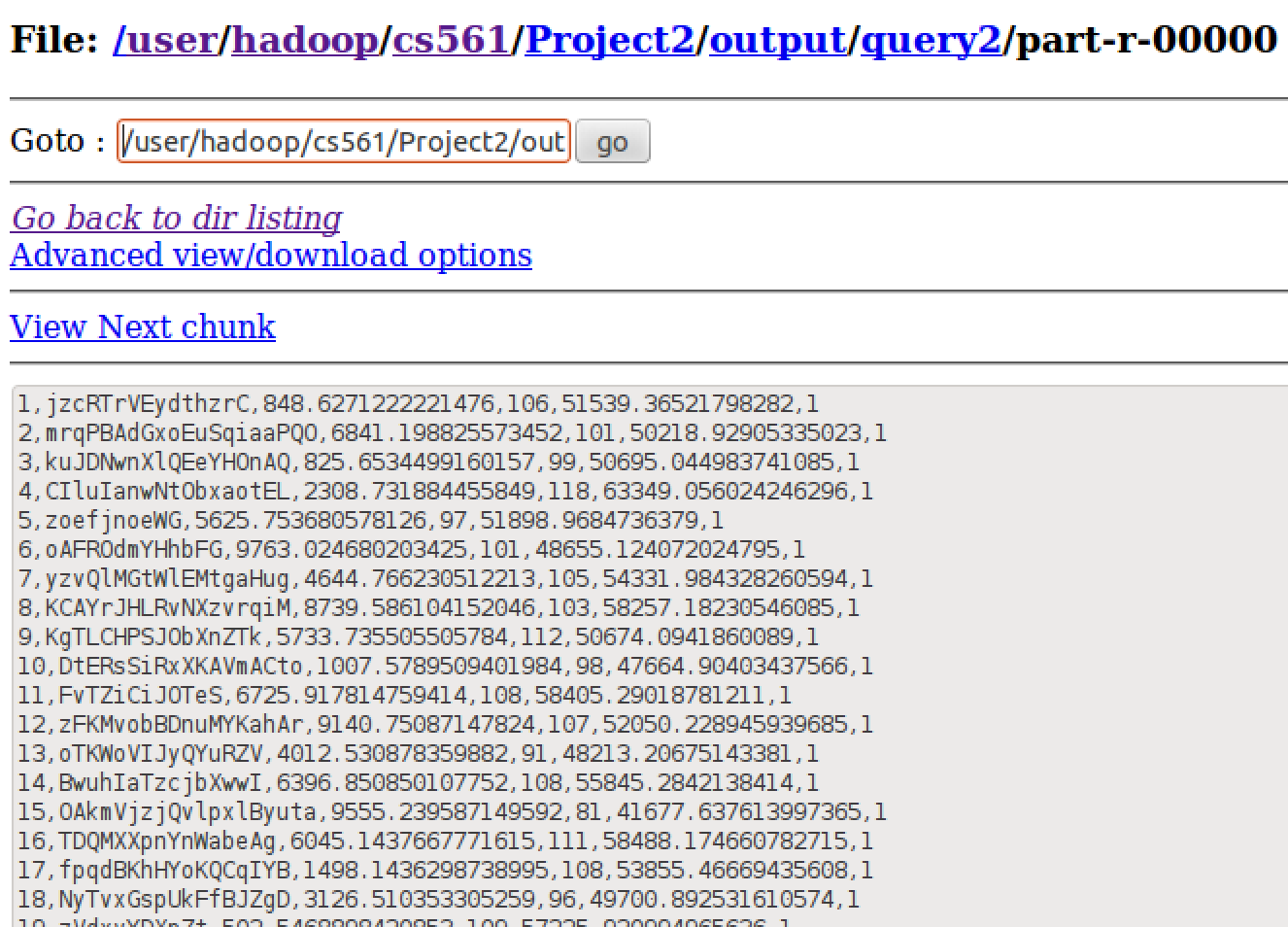
-cust = LOAD '/user/hadoop/cs561/dataset/Customer.txt' USING PigStorage(',') AS (ID:int, name:chararray, age:int, CountryCode:int, salary: double);

-alpha = FOREACH cust generate ID, name, salary;

-D = join C by group, alpha by ID;

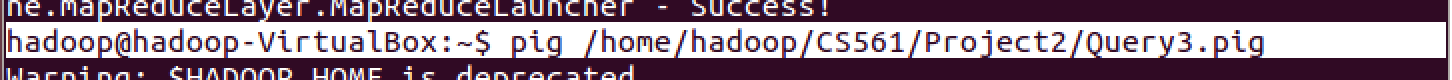
-E = foreach D generate ID, name as Name, salary as Salary, NumOfTransaction, TotalSum, MinItems;

-store E into '/user/hadoop/cs561/Project2/output/query2\_1' USING PigStorage(',');



**Query3**

The query 3 needs information from the two Customer.txt and Transaction.txt file. In the first, we read the two dataset files from the HDFS and join them based on the custID. Then we group the joined dataset by CountryCode. For each group key CountryCode, we count the distinct CustomerID as NumberOfCustomers and calculate the minimum and maximum of the field TransTotal as MinTransTotal and MaxTransTotal. We store the output into file on the HDFS(snapshot in the following figure: CountryCode, NumberOfCustomers, MinTransTotal, MaxTransTotal).

The running command line for running the pig query is as follows:

Query 3:

-trans = LOAD '/user/hadoop/cs561/dataset/Transaction.txt' USING PigStorage(',') AS (transID:int, custID:int, transTotal:double, transNumItems:int, transDesc: chararray);

-A = FOREACH trans generate custID, transTotal;

-cust = LOAD '/user/hadoop/cs561/dataset/Customer.txt' USING PigStorage(',') AS (ID:int, name:chararray, age:int, CountryCode:int, salary: double);

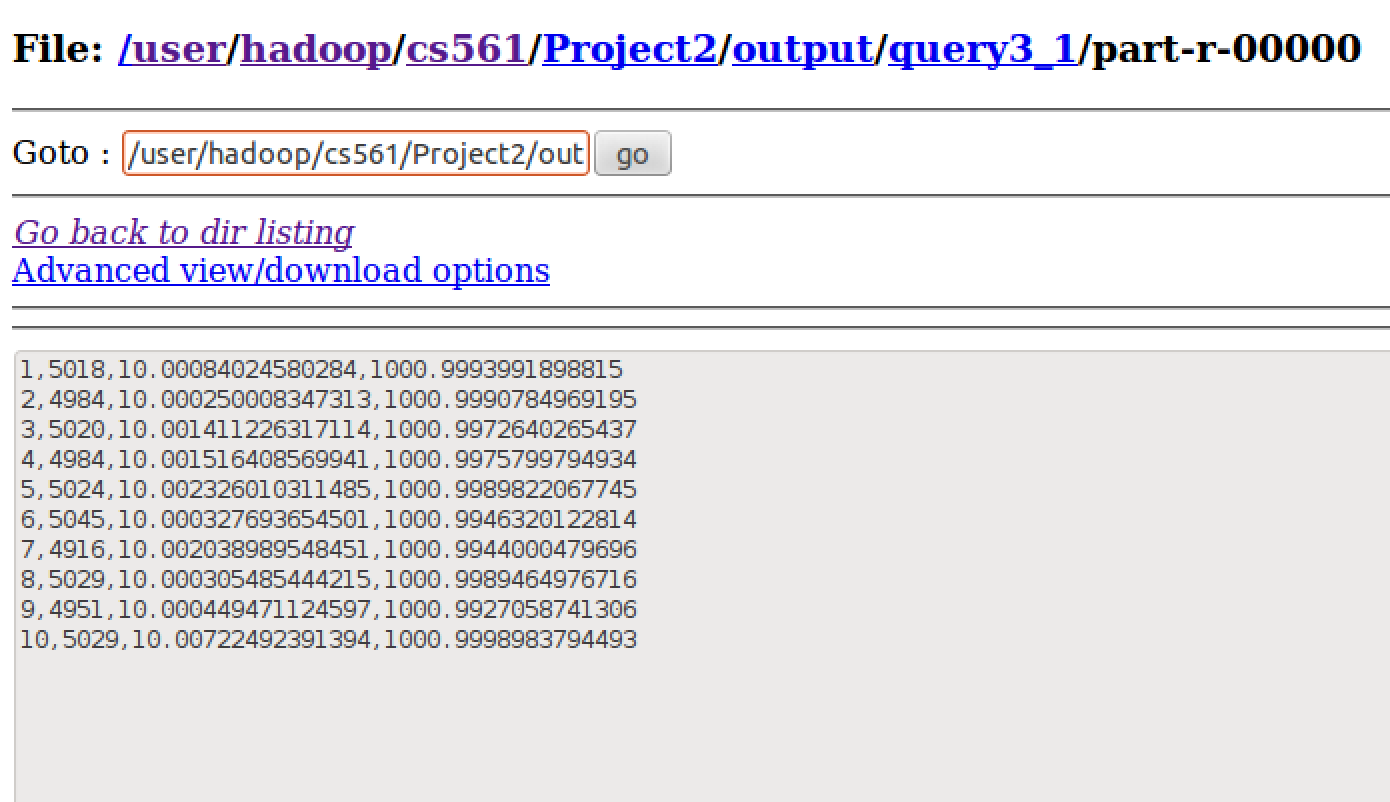
-alpha = FOREACH cust generate ID, CountryCode;

-B = join A by custID, alpha by ID;

-C = group B by CountryCode;

-D = foreach C {E = DISTINCT B.ID; transtotal = B.transTotal; generate group, COUNT(E) as NumberOfCustomers, MIN(B.transTotal) as MinTotal, MAX(B.transTotal) as MaxTotal;};

-store D into '/user/hadoop/cs561/Project2/output/query3' USING PigStorage(',');



**Query4**

The query 4 needs to do join on Customer.txt file and Transaction.txt file based on the custID. At first we group the Transaction dataset by the custID and for each group we aggregate the count of transactions.

**Approach 1: Using the Min function without Rank**

We group the whole dataset to find out the minimum number of transactions and filter the whole dataset to get the row(s) with the minimum number of transactions.

**Approach 2: Using Rank to find the Min**

After aggregating the Transaction dataset, we rank it in ascending order based on the sum on the transactions for each customer and then filter the records for which the rank is 1, i.e. the records with least number of counts. After filtering, only the filtered records are joined to produce the final output

Then we join this output with the Customer.txt file to get the name of these customers. The output is stored in file on HDFS(snapshot in the following figure: CustomerName, MinNumTrans)

The running command line for running the pig query is as follows:



Query 4: Approach 1

-trans = LOAD '/user/hadoop/cs561/dataset/Transaction.txt' USING PigStorage(',') AS (transID:int, custID:int, transTotal:double, transNumItems:int, transDesc: chararray);

-A = group trans by custID;

-B = foreach A generate group, COUNT(trans) as numOfTransactions;

-C = group B ALL;

-D = foreach C generate MIN(B.numOfTransactions) as min;

-E = FILTER B BY numOfTransactions == D.min;

-cust = LOAD '/user/hadoop/cs561/dataset/Customer.txt' USING PigStorage(',') AS (ID:int, name:chararray, age:int, CountryCode:int, salary: double);

-alpha = FOREACH cust generate ID, name;

-F = join E by group, alpha by ID;

-G = foreach F generate name as CustomerName, numOfTransactions as LeastNum;

-store G into '/user/hadoop/cs561/Project2/output/query4\_1' USING PigStorage(',');

Query 4: Approach 2

cust = load '/user/hadoop/cs561/dataset/Customer.txt' using PigStorage(',') as (id,name,age,countrycode,salary);

trans = load '/user/hadoop/cs561/dataset/Transaction.txt' using PigStorage(',') as (transid,custid,transtotal,transnumitems,transdesc);

custTrim = foreach cust generate id,name;

transTrim = foreach trans generate custid;

g = group transTrim by custid;

transCount = foreach g generate group, COUNT(transTrim.custid) as cnt;

r = RANK transCount BY cnt asc;

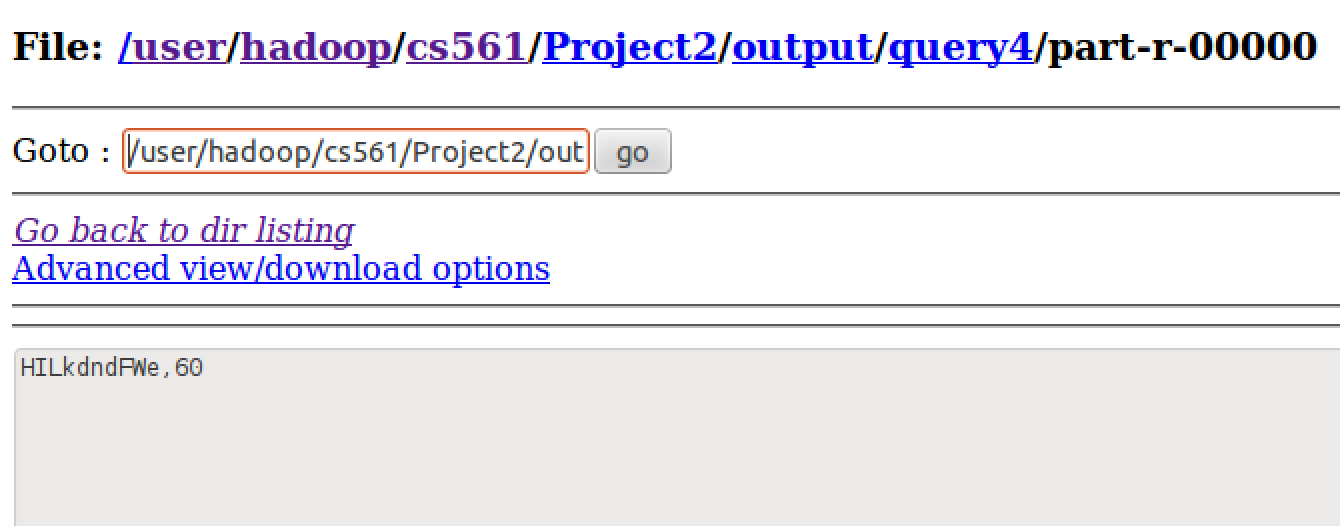
o = FILTER r BY (rank\_transCount == 1);

ot = foreach o generate $1 as cid, $2 as cn;

jcot = join ot by cid, custTrim by id;

oj = foreach jcot generate $3,$1;

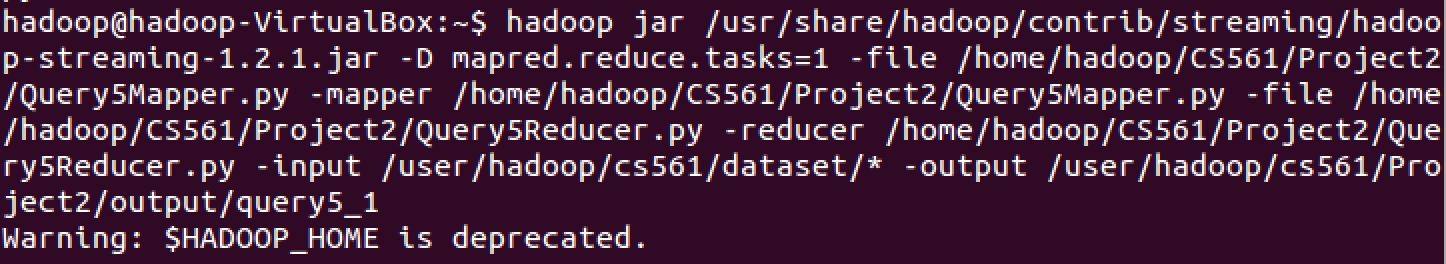
store oj into '/user/hadoop/cs561/output/project2\_4' using PigStorage(',');



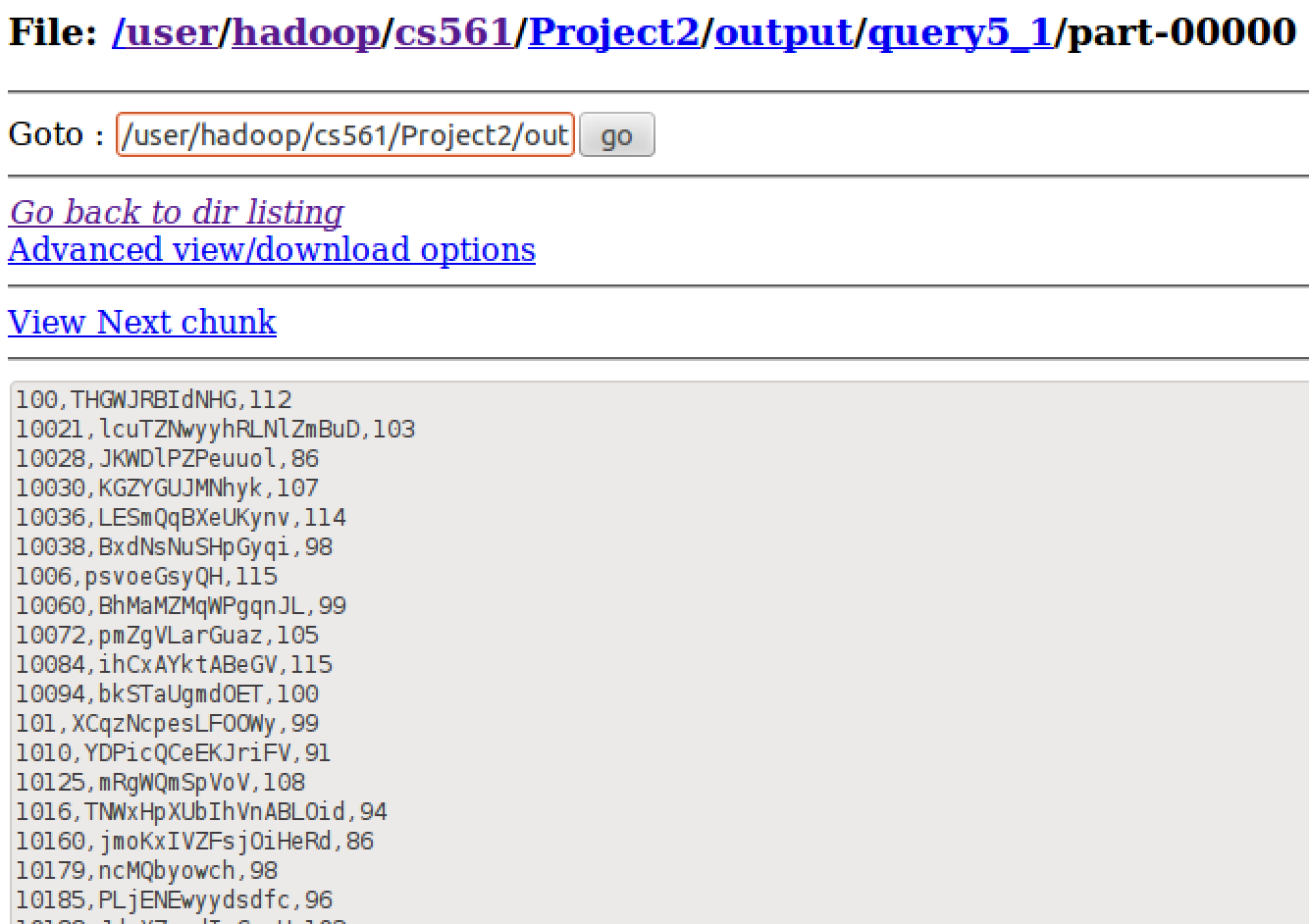
**Query5**

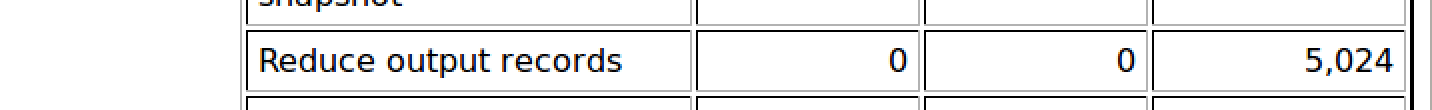
We use Hadoop Streaming in Python to do the query 5. We come up with a mapper and a reducer, while the join is finished in the reducer side. In the mapper, after loading the dataset from the standard input and split the lines into attributes. We first set “-1” for each attribute that we want as default. If the second attribute length is longer than 6, it means this line is from the Customer.txt file and we select the first attribute as custID, second attribute as name and the fourth attribute as countrycode. If not, it means that this line is from the Transaction.txt file and we select the first attribute as tranID and the second attribute as custID. We output the attribute values in the following order: custID, name, countrycode, transID. That means if the line is from the Customer.txt file, it would be like (custID, name, countrycode, “-1”). If the line is from the Transaction.txt file, if would be like (custID, “-1”, “-1”, transID). Thus for the same key custID, the information from the Transaction file would be stored first and the information from the Customer file is after. Then in the reducer, for each custID, we count the number of transactions then if the line is from the Customer file on the same custID and the countrycode equals 5, we would print the output (CustomerID, CustomerName, CountTransactions). Otherwise, we could not print the output.

The running command line for running the Hadoop Streaming is as follows:



The snapshot for the output is as follows and it takes 1mins, 14sec to finish the query and the result is that there are 5024 customers whose countrycode is 5 which matches the result in the query3.





**Query6**

The query 6 is similar as WordCount example. In the mapper side, we generate countrycode as key and one as the value. In the reducer side, we sum up the count for the same key and store the output on HDFS.

The two output figures are as follows:

