**Project 2 – Pig Scripts, Hadoop Streaming, RHadoop**

Qian Wang, Shile Zhao

**# Project 2 document**

### First to load datasets into Pig Storage and give the column names

CUSTS = LOAD '/user/hadoop/data/Customers' USING PigStorage(',') AS (custID:INT, custName:CHARARRAY, age:INT, countryCode:INT, salary:FLOAT);

TRANS = LOAD '/user/hadoop/data/Transactions' USING PigStorage(',') AS (transID:INT, custID:INT, transTotal:FLOAT, transNumItems:INT, transDesc:CHARARRAY);

**### Qeury1**

#### Firstly group by transaction dataset by custID column. Secondly, do the count and sum function on grouped dataset

Q1\_gp = GROUP TRANS BY custID;

DESCRIBE Q1\_gp;

Q1 = FOREACH Q1\_gp GENERATE group, COUNT(TRANS.transID), SUM(TRANS.transTotal);

STORE Q1 INTO '/user/hadoop/output/project2/q1';

**# Query2**

# Firstly, join Customers and Transactions by custID

# Secondly, group by the new joined dataset by custID

# Thirdly, do the statistics operations on the grouped dataset

Q2\_join = JOIN TRANS BY custID, CUSTS BY custID;

Q2\_group = GROUP Q2\_join BY CUSTS::custID;

Q2 = FOREACH Q2\_group GENERATE group, Q2\_join.CUSTS::custName, Q2\_join.CUSTS::salary, COUNT(Q2\_join.TRANS::transID), SUM(Q2\_join.TRANS::transTotal), MIN(Q2\_join.TRANS::transNumItems);

STORE Q2 INTO '/user/hadoop/output/project2/q2';

**# Query3**

# Firstly, join Customers and Transactions by custID

# Secondly, group the new joined dataset by custID

# Thirdly, do the statistics operations on the new grouped dataset

Q3\_join = JOIN TRANS BY custID, CUSTS BY custID;

Q3\_group = GROUP Q3\_join BY CUSTS::countryCode;

Q3 = FOREACH Q3\_group GENERATE group, COUNT(Q3\_join.CUSTS::custID), MIN(Q3\_join.TRANS::transTotal), MAX(Q3\_join.TRANS::transTotal);

STORE Q3 INTO '/user/hadoop/output/project2/q3';

**# Query4**

# Firstly, join the two datasets

# Secondly, group the two dataset by custID

# Thirdly, do the statistics operations on new grouped dataset

# Fourthly, sort the result from last step in ascending

# Fifthly, select the first item of the sorting list

Q4\_join = JOIN TRANS BY custID, CUSTS BY custID;

Q4\_group = GROUP Q4\_join BY CUSTS::custID;

Q41 = FOREACH Q4\_group GENERATE Q4\_join.CUSTS::custName, COUNT(Q4\_join.TRANS::transID) AS transNum;

Q42 = ORDER Q41 BY transNum;

Q43 = LIMIT Q42 1;

STORE Q43 INTO '/user/hadoop/output/project2/q4';

**# Query 5**

####!/usr/bin/env python

'''

hadoop jar /usr/share/hadoop/contrib/streaming/hadoop-streaming-1.2.1.jar -input input/Transactions -output out/py -reducer Q5reducer.py -mapper Q5mapper.py -file Customers -file Q5reducer.py -file Q5mapper.py

'''

**#### mapper.py**

####map side join first

import sys

customers = {}

with open('Customers','rb') as f:

for line in f:

record = line.strip().split(',')

if int(record[3].strip())==5:

customers[int(record[0])]=record[1]

for line in sys.stdin:

record = line.split(',')

cid = int(record[1].strip())

if cid in customers.keys():

print "%d,%s,%d" % (cid,customers[cid],1)

**#### reducer.py**

#!/usr/bin/env python

#group

import sys

summary= {}

for line in sys.stdin:

record = line.split(',')

cid = int(record[0])

val = summary.get(int(record[0]), [record[1], 0])

val[1] += int(record[2])

summary[cid] = val

for cid in summary:

print '%d,%s,%d' % tuple([cid] + summary[cid])

**the command line:**

>hadoop jar /usr/share/hadoop/contrib/streaming/hadoop-streaming-1.2.1.jar \

> -input input/Transactions -output out/py \

> -reducer Q5reducer.py -mapper Q5mapper.py \

> -file Customers -file Q5reducer.py -file Q5mapper.py

**### Query 6**

Sys.setenv(HADOOP\_CMD='/usr/share/hadoop/bin/hadoop')

library("rmr2")

library("rhdfs")

hdfs.init()

Q6 <- function(input, output = NULL,pattern = ","){

q6.map<-function(.,lines){

res = NULL

for (line in lines){

res = c(res,unlist(strsplit(line,split=','))[4])

}

return(keyval(res, '1'))

}

q6.reduce<-function(k,v){

keyval(k,toString(sum(as.integer(v))))

}

mapreduce(

input = input,

output = output,

map = q6.map,

reduce = q6.reduce,

combine = FALSE,

input.format=make.input.format("text"))

}

inputPath = '/user/hadoop/input/Customers'

outputPath = '/user/hadoop/out/rhadoop'

Q6(inputPath, outputPath)

results <- from.dfs(outputPath)

x <- results$key

y <- as.integer(results$val)

## Plot the values

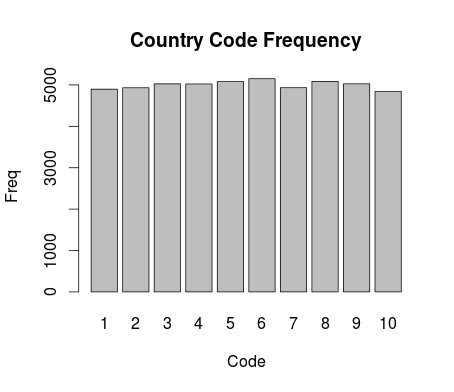
barplot(y, main="Country Code Frequency", xlab="Code", ylab="Freq", names.arg=x)

## Sort the values and Plot them

z = sort(y,index.return = TRUE)$ix

barplot(y[z], main="Sorted Country Code Frequency", xlab="Code", ylab="Freq", names.arg=x[z], col="blue")

**##### unsorted plot:**



**##### sorted plot:**

