**HU Extension Assignment 10 E185 Big Data Analytics**

**Issued on: April 27, 2013 Due by 17:25PM, May 03, 2013**

**Problem 1. Attached ZIP file rmr2-master.ZIP in directory docs contains tutorial.HTML file. Please run Word Count example.**

After struggling a lot with the installation procedure of rmr2, we ended up quitting the process and using the Virtual Machine that was kindly provided by the Professor in order to perform the exercises.

We started up by logging into the machine and trying out the rmr2 library on the sample example it provides:

**> library(rmr2)**

Loading required package: Rcpp

Loading required package: RJSONIO

Loading required package: digest

Loading required package: functional

Loading required package: stringr

Loading required package: plyr

Loading required package: reshape2

**> small.ints = to.dfs(1:1000)**

13/05/04 10:57:59 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

13/05/04 10:57:59 INFO compress.CodecPool: Got brand-new compressor

Warning message:

In to.dfs(1:1000) : Converting to.dfs argument to keyval with a NULL key

**> output = mapreduce(**

**+ input = small.ints,**

**+ map = function(k, v) cbind(v, v^2))**

packageJobJar: [/tmp/RtmpYaWiIj/rmr-local-env14e768dc51cb, /tmp/RtmpYaWiIj/rmr-global-env14e7742caaa5, /tmp/RtmpYaWiIj/rmr-streaming-map14e743ef9d3, /var/lib/hadoop-0.20/cache/cloudera/hadoop-unjar1410289754519500592/] [] /tmp/streamjob1234815229337710025.jar tmpDir=null

13/05/04 10:58:13 INFO mapred.FileInputFormat: Total input paths to process : 1

13/05/04 10:58:14 INFO streaming.StreamJob: getLocalDirs(): [/var/lib/hadoop-0.20/cache/cloudera/mapred/local]

13/05/04 10:58:14 INFO streaming.StreamJob: Running job: job\_201305041056\_0001

13/05/04 10:58:14 INFO streaming.StreamJob: To kill this job, run:

13/05/04 10:58:14 INFO streaming.StreamJob: /usr/lib/hadoop-0.20/bin/hadoop job -Dmapred.job.tracker=localhost:8021 -kill job\_201305041056\_0001

13/05/04 10:58:14 INFO streaming.StreamJob: Tracking URL: http://localhost.localdomain:50030/jobdetails.jsp?jobid=job\_201305041056\_0001

13/05/04 10:58:15 INFO streaming.StreamJob: map 0% reduce 0%

13/05/04 10:58:24 INFO streaming.StreamJob: map 50% reduce 0%

13/05/04 10:58:25 INFO streaming.StreamJob: map 100% reduce 0%

13/05/04 10:58:26 INFO streaming.StreamJob: map 100% reduce 100%

13/05/04 10:58:26 INFO streaming.StreamJob: Job complete: job\_201305041056\_0001

13/05/04 10:58:26 INFO streaming.StreamJob: Output: /tmp/RtmpYaWiIj/file14e710ae819

**> result = from.dfs(output)**

**> str(result)**

List of 2

$ key: NULL

$ val: num [1:1000, 1:2] 1 2 3 4 5 6 7 8 9 10 ...

..- attr(\*, "dimnames")=List of 2

.. ..$ : NULL

.. ..$ : chr [1:2] "v" ""

**> head(result$val)**

v

[1,] 1 1

[2,] 2 4

[3,] 3 9

[4,] 4 16

[5,] 5 25

[6,] 6 36

**> tail(result$val)**

v

[995,] 995 990025

[996,] 996 992016

[997,] 997 994009

[998,] 998 996004

[999,] 999 998001

[1000,] 1000 1000000

Given the result above, it seems that the library is correctly installed and that we can proceed to the problems.

In order to run the Problem 1 example, we create an R file with the contents of the word count function from the example:

**[cloudera@localhost Assign10]$ cat Problem1\_WordCount.R**

library(rmr2)

wordcount = function(input, output = NULL, pattern = " "){

wc.map = function(., lines) {

keyval(unlist(strsplit(x = lines, split = pattern)), 1)

}

wc.reduce = function(word, counts ) {

keyval(word, sum(counts))

}

mapreduce(

input = input ,

output = output,

input.format = "text",

map = wc.map,

reduce = wc.reduce,

combine = T)

}

And we upload a file to HDFS from the home directory to try the function out:

**[cloudera@localhost Assign10]$ hadoop fs -put cat-descriptions\_120396.txt .**

**[cloudera@localhost Assign10]$ hadoop fs -ls**

Found 8 items

-rw-r--r-- 1 cloudera supergroup 28194 2013-05-04 11:18 /user/cloudera/cat-descriptions\_120396.txt

drwxr-xr-x - cloudera supergroup 0 2013-04-25 17:58 /user/cloudera/output

-rw-r--r-- 1 cloudera supergroup 11553456 2013-04-10 09:30 /user/cloudera/ratings.csv

drwxr-xr-x - cloudera supergroup 0 2013-04-08 11:26 /user/cloudera/reut2

drwxr-xr-x - cloudera supergroup 0 2013-04-13 19:39 /user/cloudera/reuters

-rw-r--r-- 1 cloudera supergroup 2511 2013-04-21 13:53 /user/cloudera/stocks.txt

drwxr-xr-x - cloudera supergroup 0 2013-04-08 11:03 /user/cloudera/testdata

-rw-r--r-- 1 cloudera supergroup 6 2013-04-10 09:30 /user/cloudera/user-ids.txt

We head back into R to run it:

**> source("Problem1\_WordCount.R")**

Loading required package: Rcpp

Loading required package: RJSONIO

Loading required package: digest

Loading required package: functional

Loading required package: stringr

Loading required package: plyr

Loading required package: reshape2

**> retVal = wordcount("/user/cloudera/cat-descriptions\_120396.txt")**

packageJobJar: [/tmp/Rtmp3tohHQ/rmr-local-env1df1656794af, /tmp/Rtmp3tohHQ/rmr-global-env1df1122e4e26, /tmp/Rtmp3tohHQ/rmr-streaming-map1df162d636f, /tmp/Rtmp3tohHQ/rmr-streaming-reduce1df170b7cccd, /tmp/Rtmp3tohHQ/rmr-streaming-combine1df1296bf3e7, /var/lib/hadoop-0.20/cache/cloudera/hadoop-unjar1496132740791981679/] [] /tmp/streamjob2922333704590015704.jar tmpDir=null

13/05/04 11:20:02 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

13/05/04 11:20:02 WARN snappy.LoadSnappy: Snappy native library not loaded

13/05/04 11:20:02 INFO mapred.FileInputFormat: Total input paths to process : 1

13/05/04 11:20:02 INFO streaming.StreamJob: getLocalDirs(): [/var/lib/hadoop-0.20/cache/cloudera/mapred/local]

13/05/04 11:20:02 INFO streaming.StreamJob: Running job: job\_201305041056\_0002

13/05/04 11:20:02 INFO streaming.StreamJob: To kill this job, run:

13/05/04 11:20:02 INFO streaming.StreamJob: /usr/lib/hadoop-0.20/bin/hadoop job -Dmapred.job.tracker=localhost:8021 -kill job\_201305041056\_0002

13/05/04 11:20:02 INFO streaming.StreamJob: Tracking URL: http://localhost.localdomain:50030/jobdetails.jsp?jobid=job\_201305041056\_0002

13/05/04 11:20:03 INFO streaming.StreamJob: map 0% reduce 0%

13/05/04 11:20:13 INFO streaming.StreamJob: map 50% reduce 0%

13/05/04 11:20:14 INFO streaming.StreamJob: map 100% reduce 0%

13/05/04 11:20:21 INFO streaming.StreamJob: map 100% reduce 33%

13/05/04 11:20:24 INFO streaming.StreamJob: map 100% reduce 100%

13/05/04 11:20:28 INFO streaming.StreamJob: Job complete: job\_201305041056\_0002

13/05/04 11:20:28 INFO streaming.StreamJob: Output: /tmp/Rtmp3tohHQ/file1df14594938d

**> wcResult = from.dfs(retVal)**

**> str(wcResult$val)**

num [1:1762] 1858 2 2 2 1 ...

**> str(wcResult)**

List of 2

$ key: chr [1:1762] "" "39" "56" "a" ...

$ val: num [1:1762] 1858 2 2 2 1 ...

And it seems to bring the desired result.

**Problem 2. Attached ZIP file rmr2-master.ZIP in directory docs contains tutorial.HTML file. Please run Logistic Regression example.**

Similarly to the previous problem, we create a file with the logistic regression function to be run on the program from the tutorial and the sample file.

**[cloudera@localhost Assign10]$ cat Problem2\_Logistic.R**

library(rmr2)

logistic.regression =

function(input, iterations, dims, alpha) {

lr.map =

function(., M) {

Y = M[,1]

X = M[,-1]

keyval(

1,

Y \* X \*

g(-Y \* as.numeric(X %\*% t(plane))))

}

lr.reduce =

function(k, Z)

keyval(k, t(as.matrix(apply(Z,2,sum))))

plane = t(rep(0, dims))

g = function(z) 1/(1 + exp(-z))

for (i in 1:iterations) {

gradient =

values(

from.dfs(

mapreduce(

input,

map = lr.map,

reduce = lr.reduce,

combine = T)))

plane = plane + alpha \* gradient

}

plane

}

And then we run the sample with the data in the example in the document file in R.

**> source("Problem2\_Logistic.R")**

Loading required package: Rcpp

Loading required package: RJSONIO

Loading required package: digest

Loading required package: functional

Loading required package: stringr

Loading required package: plyr

Loading required package: reshape2

**> set.seed(0)**

**> test.size = 10^5**

**> eps = rnorm(test.size)**

**> testdata = to.dfs( as.matrix(**

**+ data.frame(**

**+ y = 2 \* (eps > 0) - 1,**

**+ x1 = 1:test.size,**

**+ x2 = 1:test.size + eps)))**

13/05/04 12:18:24 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

13/05/04 12:18:24 INFO compress.CodecPool: Got brand-new compressor

Warning message:

In to.dfs(as.matrix(data.frame(y = 2 \* (eps > 0) - 1, x1 = 1:test.size, :

Converting to.dfs argument to keyval with a NULL key

**> out = logistic.regression(testdata, 3, 2, 0.05)**

packageJobJar: [/tmp/RtmpqZcIc3/rmr-local-envd1f400c32c6, /tmp/RtmpqZcIc3/rmr-global-envd1f6f469b26, /tmp/RtmpqZcIc3/rmr-streaming-mapd1f7f7a2662, /tmp/RtmpqZcIc3/rmr-streaming-reduced1f239c6c2a, /tmp/RtmpqZcIc3/rmr-streaming-combined1f5d13bcaa, /var/lib/hadoop-0.20/cache/cloudera/hadoop-unjar2522530270540754434/] [] /tmp/streamjob5314243029236120092.jar tmpDir=null

(...)

13/05/04 12:19:59 INFO streaming.StreamJob: Output: /tmp/RtmpqZcIc3/filed1f24dae07a

Deleted hdfs://localhost:8020/tmp/RtmpqZcIc3/filed1f4a415da1

**> out**

x1 x2

[1,] 1961651 1967645

We have a reasonable result from the output, so we declare the example successful.