CSC 555 Mining Big Data

Assignment 4

Due Monday, February 26th

1) Consider a Hadoop job that will result in 79 blocks of output to HDFS.
Suppose that reading a block takes 1 minute and writing an output block to HDFS takes 1 minute. The HDFS replication factor is set to 2.

   a) How long will it take for the reducer to write the job output on a 5-node Hadoop cluster? (ignoring the cost of Map processing, but counting replication cost in the output writing).

   b) How long will it take for reducer(s) to write the job output to 10 Hadoop worker nodes? (Assume that data is distributed evenly and replication factor is set to 1)

   c) How long will it take for reducer(s) to write the job output to 10 Hadoop worker nodes? (Assume that data is distributed evenly and replication factor is set to 2)

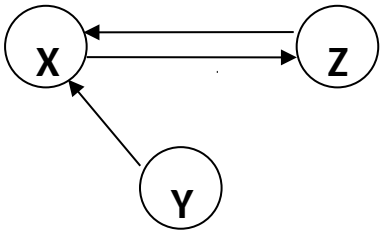
   d) How long will it take for reducer(s) to write the job output to 100 Hadoop worker nodes? (Assume that data is distributed evenly and replication factor is set to 1)

   e) Suppose that replication factor was changed to 3: how long will it take for the reducer to write the job output on a 5-node Hadoop cluster? (same question as a) but with replication of 3).

   You can ignore the network transfer costs as well as the possibility of node failure.

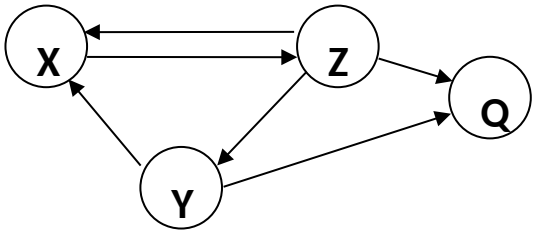
2)

   a) Consider the following graph



   Compute the page rank for the nodes in this graph. If you are multiplying matrices manually, you may stop after 6 steps. If you use a tool (e.g., Matlab, website, etc.) for matrix multiplication, you should get your answer to converge.

   b) Now consider a dead-end node Q:



   What is the page rank of Q?

c) Exercise 5.1.6 from Mining of Massive Datasets

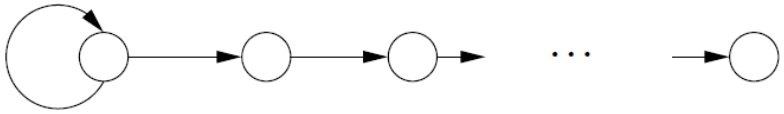


Figure 5.9: A chain of dead ends

Exercise 5.1.6: Suppose we recursively eliminate dead ends from the graph, solve the remaining graph, and estimate the PageRank for the dead-end pages as described in Section 5.1.4. Suppose the graph is a chain of dead ends, headed by a node with a self-loop, as suggested in Fig. 5.9. What would be the PageRank assigned to each of the nodes?

3) Given the input data [(1pm, $5), (2pm, $15), (3pm, $15), (4pm, $20), (5pm, $10), (6pm, $20), (7pm, $30), (8pm, $25), (9pm, $22), (10pm, $30), (11pm, $30), (12pm, $40)]. We will discuss Storm on Wednesday.

a) What will the Hive query "compute average price" return? (yes, this is as obvious as it seems, asked for comparison with part-b)

b) What will a Storm query "compute average price per each 4 hour window" return? (tumbling, i.e., non-overlapping window of tuples, as many as you can fit)

c) What will a Storm query "compute average price per each 4 hour window" return? (sliding, i.e. overlapping window of tuples, moving the window forward 3 hours each time)

4) In this section you will run another custom MapReduce job. You can either use Hadoop streaming or a Java implementation if you prefer.

Implement and run MapReduce to compute: Count number of distinct odd integers in the input file – e.g., {1,2,3,1,5,2,3, 4, 4, 1} => should give you a count of 3 because 1, 3, and 5 are present at least once).
You can generate the input file using this python code linked below (it will create a data file NumbersAndStrings.txt, run with python numStrings.py), it will take about a minute or so. Note that the text file contains numbers and non-numeric data. Your mapper will have to skip over elements that are not numbers.

http://rasinsrv07.cstcis.cti.depaul.edu/CSC555/numStrings.py

Don't forget to submit the code you used, the output and screenshot of Hadoop streaming running.

5) In this section you will practice using HBase and setup Mahout and run the curve-clustering example that we discuss in class on 2/21.

a) Note that HBase runs on top of HDFS, bypassing MapReduce (so only NameNode and DataNode need to be running). You can use your 3-node cluster or the 1-node cluster to run HBase, but specify which one you used.

cd
(Download HBase)
wget http://rasinsrv07.cstcis.cti.depaul.edu/CSC555/hbase-0.90.3.tar.gz
gunzip hbase-0.90.3.tar.gz
tar xvf hbase-0.90.3.tar

cd hbase-0.90.3

(Start HBase service, there is a corresponding stop service and this assumes Hadoop home is set)
bin/start-hbase.sh

(Open the HBase shell – at this point jps should show HMaster)
bin/hbase shell

(Create an employee table and two column families – private and public. Please watch the quotes, if ' turns into ` the commands will not work)


```
create 'employees', {NAME=> 'private'}, {NAME=> 'public'}
put 'employees', 'ID1', 'private:ssn', '111-222-334'
put 'employees', 'ID2', 'private:ssn', '222-333-445'
put 'employees', 'ID3', 'private:address', '123 Fake St.'
put 'employees', 'ID1', 'private:address', '243 N. Wabash Av.'

scan 'employees'
```


Now that we have filled in a few values, add at least 2 columns with at least 5 new values total to the "public" column family (e.g., position, officeNumber). Verify that the table has been filled in properly with scan command and submit a screenshot.

b) Download and setup Mahout:

cd
(download mahout zip package)
wget http://rasinsrv07.cstcis.cti.depaul.edu/CSC555/apache-mahout-distribution-0.11.2.zip
(Unzip the file)
unzip apache-mahout-distribution-0.11.2.zip

set the environment variables (as always, you can put these commands in ~/.bashrc to automatically set these variables every time you open a new connection, source ~/.bashrc to refresh)

export MAHOUT_HOME=/home/ec2-user/apache-mahout-distribution-0.11.2
export PATH=/home/ec2-user/apache-mahout-distribution-0.11.2/bin:$PATH
be absolutely sure you set Hadoop home variable (if you haven't):

Download and prepare synthetic data – it represents a list of 2D curves, represented as a 50-point vector.

Download the synthetic data example:
wget http://rasinsrv07.cstcis.cti.depaul.edu/CSC555/synthetic_control.data
(make a testdata directory in HDFS, the example KMeans algorithm assumes the data lives there by default.
hadoop fs -mkdir -p testdata
(copy the synthetic data over to the testdata directory on HDFS side. You can inspect the contents of the file by running nano synthetic_control.data – as you can see this is a list of 600 vectors, with individual values separated by a space)
hadoop fs -put synthetic_control.data testdata/

Please be sure to report the runtime of any command that includes "time"
time mahout org.apache.mahout.clustering.syntheticcontrol.kmeans.Job

(clusterdump is a built-in Mahout command that will produce the result of KMeans. Output file is written to clusters-10-final because that is where the output is written after 10 iterations. The center points are placed in a separate file, called clusteredPoints)
mahout clusterdump --input output/clusters-10-final --pointsDir output/clusteredPoints --output clusteranalyze.txt

The file clusteranalyze.txt contains the results of the Kmeans run after 10 iterations.

Submit the screenshot of the first page from clusteranalyze.txt (e.g., from more clusteranalyze.txt)

Submit a single document containing your written answers. Be sure that this document contains your name and "CSC 555 Assignment 4" at the top.