MAHOUT LAB

Overview: In this lab we will prepare movie ratings data using Pig and then run the Mahout Item Recommendation algorithm (Amazon style) to determine which movies are most likely to be similarly liked. We will switch off of Cloudera Quickstart VM back to the Hortonworks Sandbox for this lab because the Cloudera VM would give you memory errors.

1) Shut down the Cloudera Quickstart VM. Start your Hortonworks Sandbox VM from the previous labs.

2) The VM window shows some instructions for logging into a command line session. The user/password is root/hadoop.

3) Run the command “mahout” with no parameters to be presented with a list of Mahout programs that can be executed.

4) Grab the Movielens data in the ml-1m folder provided. We'll be using the **ratings.dat** file in this lab to run a recommendation in Mahout. This file has around a million movie ratings. Upload the file to HDFS using Hue File Browser from a web browser.

(continued on next page)

5) The data in ratings.dat looks like:
1::1193::5::978300760
which is userId::movieId::rating::timestamp but Mahout needs it formatted like userId,movieId,rating. We need to fix this. We could do it with a shell command but let's utilize Pig for this!

```
REGISTER piggybank.jar;
A = LOAD 'ratings.dat' USING
org.apache.pig.piggybank.storage.MyRegExLoader('([^\\:]+)::
([^\\:]+)::([^\\:]+)::([^\\:]+)')
      AS (user:int, movie:int, rating:int, timestamp:int);
B = FOREACH A GENERATE user, movie, rating;
Store B into 'mahout_ratings' using PigStorage(',');
```

What's going on in that Pig script? Normally we would load using PigStorage() and pass it the delimeter of '\t' or ','. Unfortunately, our data's delimeter is more than one character, which PigStorage does not support, so we have to use a custom loader from a cool utility called Piggybank (which we REGISTER much like a Java import).

We provide a regex and provide an "AS" to define what the schema looks like. We iterate with a FOREACH to limit our tuple to remove timestamp. Then we store the data into mahout_ratings (stored in /user/root in HDFS) as comma separated files.

6) Run this from the VM console: (it should take 8-10 minutes to complete if it succeeds)

```
mahout recommenditembased --input mahout_ratings --output
recommendations --tempDir /temp –similarityClassname
SIMILARITY_PEARSON_CORRELATION
```

7) Look for the new file(s) in your browser with Hue File Browser under /user/root/recommendations. It should show, for each movie id, a list of recommended movies. This is similar to Amazon's "Users who purchased this item also bought:" recommendations.