Hadoop MapReduce Lab

CS 410/510: Introduction to Performance Measurement, Modeling, and Analysis

Disclaimer

The data being used comes from actual twitter data and may contain objectionable material. A reasonable attempt has been made to remove offensive tweets, but viewing the raw data is done at your own risk.

Getting Started

Start by downloading the code and sample data from the following site:

```
web.cecs.pdx.edu/~dleblanc/hadoop/wordcount.tar.gz
```

Create a new directory and uncompress the files using the following command:

```
gunzip –c wordcount.tar.gz | tar –xopf –
```

The dash at the end is important, so don't omit it. This should produce a *data* directory with several files, and a mapper and reducer that count the number of times each word appears.

WordCount

For the lab today we will be running our programs locally on a fairly small data set. This type of approach is typical when testing Hadoop programs for deployment. Running a large Hadoop job can often take many hours, so thorough testing is crucial. We will be mimicking the behavior of Hadoop using some simple Linux commands to pipe the data into our programs. The command to run our code on the data is:

```
cat data/* | ./mapper.py | sort –k1,1 | ./reducer.py > result.dat
```

For those of you who aren't already experts in the minutiae of the Linux command line interface these command do the following:

- `cat`: sends the contents of the file to standard out
- `sort`: sorts the input bu a key value and outputs the result to standard out. This mimics the *shuffle and sort* of Hadoop
- `|`: takes the output from the item on the left and uses that as the input for the command on the right
- `>`: takes the output of the item on the left and stores it in the filename provided

To put all that together, the command takes the contents of the data directory and fives that to the mapper. The output from the mapper is sorted and given to the reducer, which combines the data. The final result is then stored in the file *result.dat*.

Once you have run this command take a look at the results file and make sure things are working correctly.

Filter WordCount Results

You'll note that the results from the previous step are case-sensitive contain a large number of special character that cloud the results. Modify the wordCount code to remove any non-alphanumeric character and change all character to lowercase.

Popular Hashtags

Hashtags are a regular feature of Twitter and their frequency can provide some interesting insights into popular culture. Modify the original wordCount code to find the most commonly used hashtags in the sample data.

Counting Words/Characters per Tweet

For this step you need to modify the original wordCount code to determine the distribution of the number of words and characters per tweet. Try to find both pieces of information in a single pass over the data. Think carefully about what the keys should be.

Running Code on Hadoop

The final step of the lab is to make sure you can run code on the Hadoop cluster. The run.sh file has been provided that contains everything needed to execute code on the cluster. In order to make it work you will first need to add the Hadoop bin to the PATH using the following command:

```
export PATH=$PATH:/u/hadoop/karavanic/hadoop-2.5.1/bin
```

It is recommended that you add this line to your .bashrc file. The run.sh file executes the following command:

```
hadoop jar /u/hadoop/karavanic/hadoop-
 2.5.1/share/hadoop/tools/lib/hadoop-streaming-2.5.1.jar \
   -file mapper.py    -mapper mapper.py \
   -file reducer.py   -reducer reducer.py \
   -input /twitter/08/01/* -output /user/**username**/demo-out
```

You will need to replace **username** with your login name before you can run the script. As a simple shell script, typing the command into the terminal will produce the same result. As explained in class, this command allows us to substitute any executables for our mapper and reducer. The input path can be replaced with whatever you will be using for the lab. You will also need to execute the following command, replacing username with your login name, to create a user directory for your results on the cluster.

```
hadoop fs –mkdir /user/username
```

The next step is to choose any single hour in the dataset to run your code on. Try to pick something other than what your classmates are using. Now run your Popular Hashtags code on the tweets from that day.