Hadoop Assignment

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.

The Data

The data for this assignment consists of every text twitter post made in August of 2014. There are approximately 500 million posts per day, and a total data size of a little over 400GB. Files are sorted into directories by month, day, and hour for the primary data set. Each file contains the posts made for a single minute. A subset of the data has been duplicated in larger files with one file per hour. To examine the file structure you can use the following command:

```
hadoop fs -ls /twitter
```

Running Your Code on Hadoop

The in-class lab contains the instructions needed to run your programs on the cluster. Be sure to test your code as thoroughly as possible on a small dataset locally before running on the full data set. If you are working remotely you will need to connect to one of the machines in the Penguin lab to run your code.

1 Content Searching

For this part of the assignment your will be writing a MapReduce program and a local implementation that finds the percentage of tweets about the movie "Guardians of the Galaxy". Its August 1st release date makes it an interesting candidate. In a real world scenario you'd want to do some content analysis to find out of the posts are generally positive or negative, but that is beyond the scope of this assignment. Come up with your own keywords and phrases to search for, but it must contain at least the following:

- Guardians of the Galaxy
- Groot
- Starlord
- Gamora
- Drax
- Rocket
- Awesome Mix

You need to find out how many times each appears in the data, the percentage of tweets that contain each, and the percentage of posts containing at least one. Make sure you take into account variations in capitalization, punctuation, and spelling. Make sure the local and MapReduce versions are producing the same results before moving to larger volumes of data.

Local Implementation: download the following file as a starting place for the local implementation:

```
web.cecs.pdx.edu/~dleblanc/hadoop/contentSearch.py
```

MapReduce Implementation: start with the wordCount code from the lab for the Hadoop portion.

2 Hadoop vs. Local Implementation

The next part of the assignment is to determine how much data is required before a MapReduce implementation is faster than a reasonably efficient single machine implementation. Determine the running times of each version. Perform a scalability study: apply both implementations to subsets of the data of various sizes and graph your results. Is there a crossover point where it is faster to use Hadoop? Why or why not?

You can use the following command to pipe input from Hadoop into the local version of the program:

```
hadoop fs -cat \twitter\08\01\00\* | ./contentSearch.py
```

This method will be slower than running everything directly from the local system, but the amount of data we need to analyze is likely to make that prohibitive.

3 File Size Comparison

The data for the first two weeks of August has been duplicated as a set of larger files. Instead of the default method of one file per minute, we have created one file per hour. This brings our file sizes from 5-10MB to around 500MB. Since Hadoop is designed around handling large files processing that data is expected to be faster. Run your code from part one on both sets of data for the first two weeks of August. How significant is the difference?