CS 430/530: Parallel Computing

Program 3: Better Inverted Index Using MapReduce
(120 points)

1 Stats Program

In this assignment, your job is to improve the Inverted Index MapReduce example. This will give you a good introduction to Mapreduce..

2 Prerequisites

2.1 Setup at home

Look through the class notes and download Hadoop to your system. Unpack it and configure it to run in pseudo-distributed mode. I would recommend setting it up under the following folder:

`$HOME/hadoop-install/hadoop`

Try out the wordcount example to make sure you are setup correctly.

2.2 Setup in lab

Setup Hadoop in the onyx lab but set it up to run in distributed mode. Run the wordcount example to make sure you can run MapReduce jobs on the onyx cluster.

3 Getting Started

Checkout Hadoop examples (including the inverter index example) from:

`cs430-resources/examples/Hadoop/myExamples`

Setup an Eclipse project for the example code. Now generate a jar file for the project and then run it using your Hadoop install. Here are the commands for running the jar file, copying the output to your local folder and then deleting the output from the Hadoop distributed filesystem. Note that we grabbing some text files from the word-count example.

```
bin/hadoop fs -put ../myExamples/word-count/input input
bin/hadoop jar ../myExamples/inverted-index/inverted-index.jar \
    InvertedIndex /path/to/input /path/to/output
```

```
bin/hadoop fs -get output output
bin/hadoop fs -rm -r output
```

Save this output to another folder since it will help you in verifying correctness of your program.

4 Better Inverted Index

The original inverted index example merely lists all the files that a given word was found in. For example, its output may be as follows:

```
people Scarlet-Letter.txt, Scarlet-Letter.txt, Les-Miserables.txt,
Les-Miserables.txt,
       Les-Miserables.txt, Les-Miserables.txt, Les-Miserables.txt,
perchance    Scarlet-Letter.txt
perhaps    Les-Miserables.txt, Les-Miserables.txt, Les-Miserables.txt,
           Les-Miserables.txt, Les-Miserables.txt, Les-Miserables.txt,
           Les-Miserables.txt, Les-Miserables.txt, Les-Miserables.txt
permit Les-Miserables.txt, Flatland.txt, Flatland.txt
```

As a a first improvement, we would like to count up the number of times the word occurs in a given file so as to reduce the output. This may look like as follows:

```
people  2 Scarlet-Letter.txt,  5 Les-Miserables.txt,
perchance     1 Scarlet-Letter.txt,
perhaps    9 Les-Miserables.txt,
permit  1 Les-Miserables.txt,  2 Flatland.txt,
```

But what we would really like is that the filenames with highest count are listed first. So we want the list of filenames to be reverse sorted by the count. If the counts are the same for two or more filenames, then we will sort by the filenames. So the output for the above example would be as follows.

```
people  5 Les-Miserables.txt, 2 Scarlet-Letter.txt,
perchance     1 Scarlet-Letter.txt,
perhaps    9 Les-Miserables.txt,
permit  2 Flatland.txt, 1 Les-Miserables.txt,
```

5 Running

Start off by getting this assignment working in the pseudo-distributed mode on your own system. Then run it in parallel in the lab with a larger data set.

A larger data set is available on onyx in the folder:

`~amit/cs430/etext-data/etext-all`

It contains 594 books as text files. This will give you a bigger data set to use in the lab to get a sense of the speed of Hadoop. You won't be able to copy it to your home directory but you can load it directly from my folder into Hadoop.

If you get it work on the large data set, then you should pat yourself on the back and go pet a baby elephant!

6 Performance Credit (20 points)

Time the performance on the large data set as the number of nodes is increased. Try a range, say: 2, 4, 8 and 16. Compare your run time with other students in the class and see if you can improve the times.

See the section *Tuning a job* in Chapter 5 of the Hadoop book for more hints on tuning the job.

7 Submission

Submit the Eclipse project for your MapReduce program along with a ready to run jar file at the top level of your submission. Please do not submit any input or output files. Please do not submit your Hadoop install folder either!a Just submit your Eclipse project folder for the program.

Make sure to include a README with your experiences.

Change directory to your assignment directory and execute the following command (on `onyx`) to submit the assignment.

```
submit cathie CS430 PA3
```

This command will pick up all files in the current directory (as well as any subdirectories recursively) and time-stamp them before transferring the combined files to the instructor's account.

Remember to submit your report as a pdf via blackboard. The blackboard due date is for the report only. The code is due ealier.