CS5531: Programming Project (Fall 2013)

Praveen Rao

Computer Science Electrical Engineering, UMKC

Email: `raopr@umkc.edu`

The goal of this project is to design, implement, and evaluate a new scheduling algorithm for Big Data processing using Apache Hadoop. This project will be done in groups of three. I encourage you to actively discuss with your peers the issues you may have w.r.t Linux and the project itself.

For this project, you will use IBM SmartCloud Enterprise. You can login via https://www-147.ibm.com/cloud/enterprise/dashboard. I will provide you instructions to use this cloud.

Here is a useful link to learn Linux commands: http://www.ibm.com/developerworks/training/kp/l-kp-command/index.html.

There are three phases in this project and they are worth 8%, 16%, and 8% of the final grade, respectively. This weightage is assigned based on the amount of effort required and the complexity of the phase. Project presentation constitutes 3% of your grade. **In all, this project is worth 35% of your final grade. All the best!**

All phases are due at 11.59 pm on the due date. You will turnin your code/reports via Blackboard.

Note: A regrade is possible for Phase 2, which involves coding. You will be allowed to fix your code within one week after I grade it. However, this will result in a 25% penalty for each test case that failed before.

1 Phase I (Download/Install/Test Apache Hadoop)

(Weightage 8%) **Due: September 20, 2013**

Complete the following tasks on the IBM cloud.

- You should be able to login into the IBM cloud portal. Start 1 master node (or name node) and 5 data nodes on the IBM cloud. Go through the videos listed under "Support" in your IBM account for instructions on how to create instances and login into them using SSH. Here are useful instructions from the cloud portal.

```
Creating Your Cluster
---------------------


You should first launch a Master Node (Image: IBM InfoSphere
```

```
BigInsights Basic - Hadoop 2.0.0.0 64b - PAYG). Once startup is fully
complete, you can copy the public IP address for your Master Node and
launch additional Data Nodes (Image: IBM InfoSphere BigInsights Basic
- Hadoop 2.0.0.0 64b - PAYG). The Data Nodes will add themselves to
the cluster as they come up.



Connect to Your Instance
------------------------


You can SSH to our instance using the standard cloud idcuser account
 with your SSH key. Putty would be an example of an SSH client you
 could use on Windows. The login name is idcuser. Once you are
 connected, you should be able to issue start-all.sh and stop-all.sh
 commands as needed. BigInsights is installed under /mnt/biginsights.
```

- If you click the master node listed under "Control Panel," then you will see the following near the bottom of the page:

```
Web Consoles on Your Instance

BigInsights Web Console
NameNode Status
JobTracker Status
TaskTracker Status
HBase Master UI
HBase Region Server UI
Hive Web Interface
```

  Click on BigInsights Web Console. Select Hadoop as the component and take a screenshot. You should have 6 healthy nodes. [TURNIN]

- Read through the documentation of HDFS shell commands online. Create a directory "cs5531" under /user/idcuser. Download the file data-8.nt.gz from Blackboard. Then uncompress it and copy data-8.nt to /user/idcuser/cs5531. Rename the newly created file to testinput.dat.

- Run the Word Count example provided in /mnt/biginsights/opt/ibm/biginsights/IHC/hadoop-1.0.3-examples.jar. Use /user/idcuser/cs5531/testinput.dat as the input. Store the output in /user/idcuser/cs5531/output. Collect the output and store it in a single local file. [TURNIN]

Complete the following tasks on your personal machine.

- Download hadoop-1.0.3.tar.gz from http://archive.apache.org/dist/hadoop/core/hadoop-1.0.3/.

- Install Eclipse (or any other IDE) on your computer. Create a new Java project and import hadoop-1.0.3.tar.gz as an archive file. Build the project by running Ant Build on build.xml. If there are errors, make sure to fix them. You should have a successful build.

2

- Locate the source code for Fair Scheduler inside src/contrib/fairscheduler. Read through the sources files and understand the big picture. Read through the documentation of Fair Scheduler online.

- Answer the following questions [TURNIN]:

  1. How can you enable Hadoop to use the Fair Scheduler for running jobs?
  2. How can you revert to FIFO scheduling within the Fair Scheduler?

1.1 What to turnin?

Submit all the items that are marked [TURNIN].

2 Phase II

(Weightage 16%) **Due: TBD**

1. You will be implementing three scheduling algorithms in this phase.

   - Scheduling algorithm 1
     Requirement: the job with least number of (map + reduce tasks) should go first; break ties by first examining the priority and then the start time
   - Scheduling algorithm 2
     Requirement: the job with the smallest percentage of map tasks completed should go first; break ties by using Scheduling Algorithm 1
   - Scheduling algorithm 3
     Requirement: the job with the highest number of pending tasks should go first; break ties by using Scheduling Algorithm 2

2. You can create a bash script to automate the invocation of jobs with different priorities, maps, and reduces.

```
#!/bin/bash

if [ $# -ne 1 ];
then
  echo "Usage: run_jobs <# of jobs>"
  exit
fi

hadoop fs -rmr /user/idcuser/output*

# change the number of entries depending on the number of jobs
priorityarray=( NORMAL NORMAL LOW VERY_LOW HIGH HIGH NORMAL )

for ((i=0;i<${1};i++))
do
  # Assuming 20 maps and 5 reduces
  # you can change the priority
  echo "Jobid: " ${i} " Priority: " ${priorityarray[i]}
  hadoop jar /home/idcuser/WordCount.jar org.apache.hadoop.examples.WordCount \
```

```
                /user/idcuser/input /user/idcuser/output${i} 20 5 ${priorityarray[i]} &
    done
```

3. You can modify WordCount.java as follows:

```
 public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();
    String[] otherArgs = new GenericOptionsParser(conf, args).getRemainingArgs();
    if (otherArgs.length != 5) {
      System.err.println("Usage: wordcount <in> <out> <# of map tasks> <# of reduce tasks>" \
                          " <VERY_HIGH|HIGH|NORMAL|LOW|VERY_LOW>");
      System.exit(2);
    }

    conf.set("mapred.map.tasks", otherArgs[2]);
    conf.set("mapred.reduce.tasks", otherArgs[3]);
    conf.set("mapred.job.priority", otherArgs[4]);

    Job job = new Job(conf, "word count");
    job.setJarByClass(WordCount.class);
    job.setMapperClass(TokenizerMapper.class);
    job.setCombinerClass(IntSumReducer.class);
    job.setReducerClass(IntSumReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    FileInputFormat.addInputPath(job, new Path(otherArgs[0]));
    FileOutputFormat.setOutputPath(job, new Path(otherArgs[1]));


    System.exit(job.waitForCompletion(true) ? 0 : 1);
  }
}
```

4. You can create a Makefile to compile the WordCount program.

```
HADOOP_HOME=/home/rpraveen/Downloads/hadoop-0.20.2
all:
mkdir -p WordCount_classes
        javac -classpath ${HADOOP_HOME}/hadoop-0.20.2-core.jar:${HADOOP_HOME}/lib/commons-cli-1.2.jar\
              -d WordCount_classes WordCount.java
        jar -cvf WordCount.jar -C WordCount_classes/ .
```

What to turnin?

1. Your entire source code in tar.gz so that it can be imported as a new project in Eclipse.

2. A 2 page report that describes: (a) your design, (b) your test cases, (c) the challenges your faced, and (d) contributions of each team member.

3 Phase III: Performance Evaluation

(Weightage 8%)

Due Date: TBD

Aim: To understand the performance impact of different scheduling algorithms Fifo, Fairshare, and the three algorithms you implemented in phase II.

You will be given a script/testcase that will start several jobs. You will measure the below metrics.

Metric 1: measure the total running time of each job Time(when a job finished) - Time(when a job was started)

Metric 2: measure the total running time of all jobs

Setup:

Create a Hadoop setup as listed below.

A. 1 master + 1 data nodes

B. 1 master + 2 data nodes

C. 1 master + 4 data nodes

Make a note of the data center that you have used along with the time when you did the experiment. If you have trouble starting many data nodes in the same data center, use multiple data centers. Measure metric 1 for different scheduling algorithms for setups (A), (B), and (C). Measure metric 2 for different scheduling algorithms for setups (A), (B), and (C).

Prepare the following plots.

Plot Type 1 - one plot per job

X-axis: setups (A), (B), and (C)

Y-axis - metric 1

5 trends - one for each scheduling algorithm

Plot Type 2 - only one plot

X-axis: setups (A), (B), and (C)

Y-axis - metric 2

5 trends - one for each scheduling algorithm

Use tools like Excel or Gnuplot.

What to turnin?

All the plots.

4   Project Presentation

(Weightage: 3%)

Each team will present their work and results in class on Wednesday, Dec. 11, 3:30-5:30 p.m.