Programming Assignment 1
A Simple Map-Reduce Program

Due on Tuesday February 25 before midnight

Description

The purpose of this project is to develop a simple Map-Reduce program on Hadoop that creates histograms of pixels

This project must be done individually. No copying is permitted. **Note: We will use a system for detecting software plagiarism, called Moss, which is an automatic system for determining the similarity of programs.** That is, your program will be compared with the programs of the other students in class as well as with the programs submitted in previous years. This program will find similarities even if you rename variables, move code, change code structure, etc.

Note that, if you use a Search Engine to find similar programs on the web, we will find these programs too. So don't do it because you will get caught and you will get an F in the course (this is cheating). Don't look for code to use for your project on the web or from other students (current or past). Just do your project alone using the help given in this project description and from your instructor and GTA only.

Platform

You will develop your program on your laptop and then on SDSC Comet. Optionally, you may use IntelliJ IDEA or Eclipse to help you develop your program on your laptop, but you should test your programs on Comet before you submit them.

How to develop your project on your laptop

You can use your laptop to develop your program and then test it and run it on Comet. This step is optional but highly recommended because it will save you a lot of time. Note that testing and running your program on Comet is required.

If you have Mac OSX or Linux, make sure you have Java and Maven installed (on Mac, you can install Maven using Homebrew `brew install maven`, on Ubuntu Linux, use `apt install maven`). If you have Windows 10, you need to install Ubuntu Shell and do: `sudo apt update`, `sudo apt upgrade`, and `sudo apt install openjdk-8-jdk maven`.

To install Hadoop and the project, cut&paste and execute on the unix shell:

```
cd
wget https://archive.apache.org/dist/hadoop/common/hadoop-2.6.5/hadoop-2.6.5.tar.gz
tar xfz hadoop-2.6.5.tar.gz
wget http://lambda.uta.edu/cse6331/project1.tgz
tar xfz project1.tgz
```

You should also set your JAVA_HOME to point to your java installation. For example, on Windows do:

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
```

To test Map-Reduce, go to `project1/examples/src/main/java` and look at the two Map-Reduce examples `Simple.java` and `Join.java`. You can compile both Java files using:

```
mvn install
```

and you can run Simple in standalone mode using:

```
~/hadoop-2.6.5/bin/hadoop jar target/*.jar Simple simple.txt output-simple
```

The file `output-simple/part-r-00000` will contain the results.

To compile and run project1:

```
cd project1
mvn install
rm -rf output
~/hadoop-2.6.5/bin/hadoop jar target/*.jar Histogram pixels-small.txt output
```

The file `output/part-r-00000` will contain the results which must be equal to `solution-small.txt`. After your project works correctly on your

laptop (it produces the same results as the solution), copy it to Comet:

```
cd
rm project1.tgz
tar cfz project1.tgz project1
scp project1.tgz xyz1234@comet.sdsc.edu:
```

where `xyz1234` is your Comet username.

Setting up your Project on Comet

This step is required. If you'd like, you can develop this project completely on Comet. Follow the directions on How to login to Comet at comet.html. Please email the GTA if you need further help.

Edit the file `.bashrc` (note: it starts with a dot) using a text editor, such as `nano .bashrc`, and add the following 2 lines at the end (cut-and-paste):

```
export JAVA_HOME=/lib/jvm/java
alias run='srun --pty -A uot143 --partition=shared --nodes=1 --ntasks-per-node=1 --mem=5G -t 00:05:00 --wait=0 --export=ALL'
export project=/oasis/projects/nsf/uot143/fegaras
```

logout and login again to apply the changes. If you have already developed project1 on your laptop, copy `project1.tgz` from your laptop to Comet. Otherwise, download project1 from the class web site:

```
wget http://lambda.uta.edu/cse6331/project1.tgz
```

Untar it:

```
tar xfz project1.tgz
rm project1.tgz
chmod -R g-wrx,o-wrx project1
```

Go to `project1/examples` and look at the two Map-Reduce examples `src/main/java/Simple.java` and `src/main/java/Join.java`. You can compile both Java files using:

```
run example.build
```

and you can run them in standalone mode using:

```
sbatch example.local.run
```

The file `example.local.out` will contain the trace log of the Map-Reduce evaluation while the files `output-simple/part-r-00000` `output-join/part-r-00000` will contain the results.

You can compile `Histogram.java` on Comet using:

```
run histogram.build
```

and you can run Histogram.java in standalone mode over a small dataset using:

```
sbatch histogram.local.run
```

The file `histogram.local.out` will contain the trace log of the Map-Reduce evaluation while the file `output/part-r-00000` will contain the results. These results must be equal to `solution-small.txt`. To run Histogram.java in distributed mode over a larger dataset, use:

```
sbatch histogram.distr.run
```

The file `histogram.distr.out` will contain the trace log of the Map-Reduce evaluation while the file `output-distr/part-r-00000` will contain the results. These results must be equal to `solution-large.txt`. Note that running in distributed mode will waste at least 10 of your SUs so do this after you make sure that your program works correctly for small data (it produces the same results as the solution).

Project Description: Pixel Histograms

A pixel in an image can be represented using 3 colors: red, green, and blue, where each color intensity is an integer between 0 and 255. In this project, you are asked to write a Map-Reduce program that derives a histogram for each color. For red, for example, the histogram will indicate how many pixels in the dataset have a green value equal to 0, equal to 1, etc (256 values). The pixel file is a text file that has one text line for each pixel. For example, the line

```
23,140,45
```

represents a pixel with red=23, green=140, and blue=45.

You should write one Map-Reduce job in the Java file `src/main/java/Histogram.java`. An empty `src/main/java/Histogram.java` has been provided, as well as scripts to build and run this code on Comet. **You should modify the Histogram.java only**.

To help you, I am giving you the pseudo code:

```
class Color {
    public short type;       /* red=1, green=2, blue=3 */
    public short intensity;  /* between 0 and 255 */
}
map ( key, line ):
  read 3 numbers from the line and store them in the variables red, green, and blue. Each number is between 0 and 255.
  emit( Color(1,red), 1 )
  emit( Color(2,green), 1 )
  emit( Color(3,blue), 1 )

reduce ( color, values )
  sum = 0
  for ( v in values )
      sum += v
  emit( color, sum )
```

In your Java main program args[0] is the file with the pixels (pixels-small.txt or pixels-large.txt) and args[1] is the output directory.

Optional: Use an IDE to develop your project

If you have a prior good experience with an IDE (IntelliJ IDEA or Eclipse), you may want to develop your program using an IDE and then test it and run it on Comet. Using an IDE is optional; you shouldn't do this if you haven't used an IDE before.

On IntelliJ IDEA, go to New→Project from Existing Sources, then choose your project1 directory, select Maven, and then Finish. To compile the project, go to Run→Edit Configurations, use + to Add New Configuration, select Maven, give it a name, use Working directory: your project1 directory, Command line: install, then Apply.

On Eclipse, you first need to install m2e (Maven on Eclipse), if it's not already installed. Then go to Open File...→Import Project from File System, then choose your project1 directory. To compile your project, right click on the project name at the Package Explorer, select Run As, and then Maven install.

Documentation

- The The Map-Reduce API. The API has two variations for most classes: `org.apache.hadoop.mapreduce` and `org.apache.hadoop.mapred`. **You should only use the classes in the package `org.apache.hadoop.mapreduce`**
- The org.apache.hadoop.mapreduce package
- The Job class

What to Submit

You need to submit the following files only:

```
project1/src/main/java/Histogram.java
project1/histogram.local.out
project1/output-distr/part-r-00000
project1/histogram.distr.out
```

Do not submit any other files. Just submit each of these 4 files one-by-one using the following form. These files are automatically uploaded directly into your personal class account for this particular project, so you don't have to include your name or student ID or project number in the file name. You may submit your files as many times as you like, but only the most recently submitted files will be retained and evaluated (newly submitted files replace the old files under the same file name). After you submit the files, please double-check that your submitted files are correct by clicking on the Status link. If you cannot login or have a problem submitting the project using this form, ask the GTA for help.

Submit Programming Assignment #1:

Select a file: Choose File no file selected

Send File

Last modified: 02/07/2020 by Leonidas Fegaras