



Project 1

| 9/27/2022

100/100 Points

Attempt 1

**REVIEW FEEDBACK**

9/22/2022

Attempt 1 Score:
100/100

Add Comment

Unlimited Attempts Allowed

▼ Details

A Simple Map-Reduce Program

Description

The purpose of this project is to develop a simple Map-Reduce program on Hadoop that analyzes data from Netflix.

This project must be done individually. No copying is permitted. **Note: We will use a system for detecting software plagiarism, called [Moss](http://theory.stanford.edu/~aiken/moss/) (<http://theory.stanford.edu/~aiken/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 Expanse \(https://uta.instructure.com/courses/118779/pages/sdsc-expanse\)](https://uta.instructure.com/courses/118779/pages/sdsc-expanse)


Optionally,
before you

[Try Again](#)

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 Expanse. This step is optional but highly recommended because it will save you a lot of time. Note that testing and running your program on Expanse is required.

If you have Mac OS 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`.

On Windows 10, you need to install [Windows Subsystem for Linux \(WSL 2\)](https://docs.microsoft.com/en-us/windows/wsl/install-win10)  [_\(https://docs.microsoft.com/en-us/windows/wsl/install-win10\)_](https://docs.microsoft.com/en-us/windows/wsl/install-win10) and then Ubuntu 20.04 or 22.04 LTS. It's OK if you have WSL 1 or an older Ubuntu. Then, open a unix shell (terminal) on WSL2 and do:

```
sudo apt update
sudo apt upgrade
sudo apt install openjdk-8-jdk maven
```

To install Hadoop and the project on Mac, Linux, or Windows WSL2, cut&paste and execute on the unix shell:

```
cd
wget https://archive.apache.org/dist/hadoop/common/hadoop-3.2.2/hadoop-3.2.2.tar.gz
tar xzf hadoop-3.2.2.tar.gz
wget http://lambda.uta.edu/cse6332/project1.tgz
tar xzf project1.tgz
```

You should also set your JAVA_HOME to point to your java installation. For example, on Windows 10 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:

```
cd
cd project1/examples
mvn install
```

and you c:

[Try Again](#)

```
~/hadoop-3.2.2/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
cd project1
mvn install
rm -rf temp output
~/hadoop-3.2.2/bin/hadoop jar target/*.jar Netflix small-netflix.csv temp output
```

The file `output/part-r-00000` must contain the same results as in file `small-solution.txt`. (The file `temp/part-r-00000` contains the temporary results from the first map-reduce job). After your project works correctly on your laptop (it produces the same results as the solution), copy it to Expanse:

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

where `xyz1234` is your Expanse username.

Setting up your Project on Expanse

This step is required. Follow the directions on How to login on Expanse at [SDSC Expanse](https://uta.instructure.com/courses/118779/pages/sdsc-expense) (<https://uta.instructure.com/courses/118779/pages/sdsc-expense>). Please email the GTA if you need further help.

First, you need to allow password-less login to local host (without it, you can't run Map-Reduce). Login on Expanse. Then do

```
ssh localhost
```

using your password and exit using control-D. Then do:

```
ssh-keygen
```

[Try Again](#)

(press enter at each line). Then do:

```
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys  
chmod og-wx ~/.ssh/authorized_keys
```

Now you should be able to ssh localhost without a password.

Then, edit the file .bashrc (note: it starts with a dot - you can see it using `ls -a`) using a text editor, such as nano .bashrc, and add the following lines at the end (cut-and-paste):

```
module load gcc openjdk  
module load slurm sdsc  
SW=/expance/lustre/projects/uot182/fegaras  
alias run='srun --pty -A uot182 --partition=shared --nodes=1 --ntasks-per-node=1 --mem=2G -t 00:05:00 --wait=0 --export=ALL'
```

logout and login again to apply the changes. If you have already developed project1 on your laptop, copy project1.tgz from your laptop to Expanse. Otherwise, download project1 from the class web site using `wget http://lambda.uta.edu/cse6332/project1.tgz`. Then untar it using:

```
tar xzf 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 output of the Map-Reduce evaluation.

You can also

[Try Again](#)

```
run netflix.build
```

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

```
sbatch netflix.local.run
```


The results generated by your program will be in the directory output. Your results must be the same as in the file small-solution.txt.

You should develop and run your programs in standalone mode until you get the correct result. After you make sure that your program runs correctly in standalone mode, you run it in distributed mode using:

```
sbatch netflix.distr.run
```

This will process the data on the large dataset large-netflix.csv (located in /expanse/lustre/projects/uot182/fegaras/netflix/combined_data_1.txt) and will write the result in the directory output-distr. These results should be similar to the results in the file large-solution.txt. Note that running in distributed mode will use up at least 10 of your SUs. So do this a couple of times only, after you make sure that your program works correctly in standalone mode. You can check your SUs using: `expanse-client user`

Project Description

In this project, you are asked to do some simple statistical analysis using Map-Reduce. You will use a real dataset from Netflix. The dataset contains movies and, for each movie, the users' ratings (from 1 to 5). See [Netflix Prize data](https://www.kaggle.com/datasets/netflix-inc/netflix-prize-data)  (https://www.kaggle.com/datasets/netflix-inc/netflix-prize-data). The large Netflix dataset is located on Expanse at /expanse/lustre/projects/uot182/fegaras/netflix/combined_data_1.txt.

First, for each different user, you need to calculate the average rating this users gives to movies. Then, you need to construct a histogram of the frequency distribution of these average ratings from all users. There should be 41 intervals in the histogram: the first is the number of users whose average rating is between 1.0 and 1.1 (exclusive), the second is for those between 1.1 and 1.2 (exclusive), ..., those between 4.9 and 5.0 (exclusive), and those exactly 5.0.

To help yo

Try Again

```
map ( key, line ):  
    if the line doesn't end with ":"  
        read 2 integers from the line into the variables user and rating (delimiter is comma ",")  
        emit( user, rating )  
  
reduce ( user, ratings ):  
    count = 0  
    sum = 0  
    for n in ratings  
        count++  
        sum += n  
    emit( user, (int)(sum/count*10) )
```

The second Map-Reduce is:

```
map ( user, line ):  
    read 2 integers from the line into the variables user and rating (delimiter is tab "\t")  
    emit( rating, 1 )  
  
reduce ( rating, values ):  
    sum = 0  
    for v in values  
        sum += v  
    emit( rating/10.0, sum )
```

You should write the two Map-Reduce jobs in the Java file `src/main/java/Netflix.java`. An empty `src/main/java/Netflix.java` has been provided, as well as scripts to build and run this code on Expanse. **You should modify the `Netflix.java` only.** In your Java main program, `args[0]` is the netflix file, `args[1]` is the intermediate directory (output of job1 and input of job2), and `args[2]` is the output directory. All the input and output file formats must be text formats. Do not use binary formats. See slide 13 in [bigdata-l03.pdf](#) (<https://uta.instructure.com/courses/118779/files/21361934?wrap=1>) to see how to chain two jobs together. It will be easier to do this project if you finish the first map-reduce first and check the results in the temporary directory `temp` and then finish the second map-reduce.

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 Expanse. Using an IDE is optional; you shouldn't do this if you haven't used an IDE before.

On IntelliJ
project, gc

Try Again

directory, Command line: install, then Apply.

On Eclipse, you first need to install [m2e](https://projects.eclipse.org/projects/technology.m2e) (https://projects.eclipse.org/projects/technology.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](https://hadoop.apache.org/docs/r3.2.2/api/index.html) (https://hadoop.apache.org/docs/r3.2.2/api/index.html). 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](https://hadoop.apache.org/docs/r3.2.2/api/org/apache/hadoop/mapreduce/package-summary.html) (https://hadoop.apache.org/docs/r3.2.2/api/org/apache/hadoop/mapreduce/package-summary.html)
- The [Job class](https://hadoop.apache.org/docs/r3.2.2/api/org/apache/hadoop/mapreduce/Job.html) (https://hadoop.apache.org/docs/r3.2.2/api/org/apache/hadoop/mapreduce/Job.html)

How to submit your project

On Expanse, make sure that the following files in your project1 directory exist and are correct:

```
src/main/java/Netflix.java
netflix.local.out
output/part-r-00000
output-distr/part-r-00000
netflix.distr.out
```

Archive your project1 directory on Expanse using:

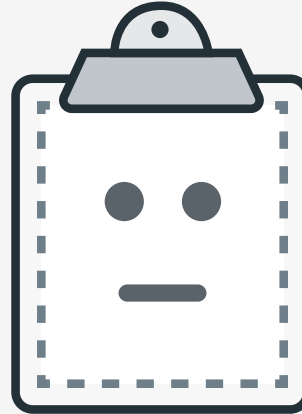
```
tar cfz project1.tgz project1
```

On your laptop, copy project1.tgz from Expanse to your laptop:

```
scp xyz1234@login.expanse.sdsc.edu:project1.tgz ./
```

Finally, up

Try Again



Preview Unavailable

project1.tgz

↓ [Download](#)

[.https://uta.instructure.com/files/22481011/download?download_frd=1&verifier=Xp6xKX3LE0rbdh92dgeG9XdUowT86UrGGTxAtvUZ\)](https://uta.instructure.com/files/22481011/download?download_frd=1&verifier=Xp6xKX3LE0rbdh92dgeG9XdUowT86UrGGTxAtvUZ)

Try Again