Project 4

100/100 Points

| 11/2/2022

Attempt 1

REVIEW FEEDBACK
11/1/2022

Attempt 1 Score:
100/100

Add Comment

Unlimited Attempts Allowed

Details

Block Matrix Addition in Spark

Description

The purpose of this project is to develop a Spark program to add two block matrices, as in Project 2.

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/) , 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

As in the previous projects, you will develop your program on SDSC Expanse. Optionally, you may use your laptop or IntelliJ Idea or Eclipse to help you d

Try Again

Using your laptop to develop your project

You may use your laptop to develop your program and then test it and run it on Expanse.

To install the Spark software and the project:

```
cd
wget https://archive.apache.org/dist/spark/spark-3.1.2/spark-3.1.2-bin-hadoop3.2.tgz
tar xfz spark-3.1.2-bin-hadoop3.2.tgz
wget http://lambda.uta.edu/cse6332/project4.tgz
tar xfz project4.tgz
```

Go to project4/examples and look at the Spark example src/main/scala/JoinSpark.scala. You can compile JoinSpark.scala using:

```
cd examples
mvn install
~/spark-3.1.2-bin-hadoop3.2/bin/spark-submit --class JoinSpark target/cse6332-spark-examples-0.1.jar e.txt d.txt output
```

To compile and run project4 on small matrices: (make sure your JAVA_HOME is correct first)

```
cd project4
mvn install
~/spark-3.1.2-bin-hadoop3.2/bin/spark-submit --class Add target/cse6332-project4-0.1.jar M-matrix-small.txt N-matrix-small.txt output
```

The file `output/part-r-00000` will contain the results which must be the same as in `solution-small.txt`.

If you want to use the Scala and the Spark interpreter to learn Scala and Spark, use the following command: (make sure your JAVA_HOME is correct first)

```
~/spark-3.1.2-bin-hadoop3.2/bin/spark-shell
```

After you make sure that the project works correctly for small data on your laptop, copy the files to Expanse and test it on Expanse.

Installing the Project on Expanse

Try Again

This step is required. If you'd like, you can develop this project completely on Expanse. If you have already developed project 4 on your laptop, copy `project4.tgz` from your laptop to Expanse. Otherwise, download project4 using `wget http://lambda.uta.edu/cse6332/project4.tgz`.

Then untar project4:

```
tar xfz project4.tgz
chmod -R g-wrx,o-wrx project4
```

Go to project4/examples and look at the Spark example JoinSpark.scala. You can compile JoinSpark.scala using:

```
run joinSpark.build
```

and you can run it in local mode using:

```
sbatch joinSpark.local.run
```

File join.local.out will contain the trace log of the Spark evaluation and `output/part*` will contain the output.

You can compile `Add.java` on Expanse using:

```
run add.build
```

and you can run it in local mode over the two small matrices using:

```
sbatch add.local.run
```

The result in the directory `output` (in the files `part-00000` and `part-00001`) must be similar to `solution-small.txt`. You should modify and run your programs in local mode until you get the correct result. After you make sure that your program runs correctly in local mode, you run it in distributed mode using:

```
sbatch add.distr.run
```

Try Again

This will write the result in the directory `output-distr` (in the files `part-r-00000` and `part-r-00001`) and must be similar to `solution-large.txt`.

Project Description

For this project, you need to implement Problem 2 using Spark in Scala. **Do not use Hadoop Map-Reduce.** That is, you need to implement block matrix addition in Spark. You will take two sparse matrices as input, convert them to block matrices, and then perform block matrix addition on them. A sparse matrix is a dataset of triples `(i,j,v)`, where `i` and `j` are the indices (of type `Int` in Scala) and `v` is that matrix value (of type `Double` in Scala) at the indices `i` and `j`. A block matrix is a Spark RDD of blocks of type `RDD[((Int,Int),Block)]` in Scala, where the two `Int` are the block coordinates. A `Block` has type `Array[Double]` in Scala and has size `rows*columns` where `rows=columns=100`. A matrix element M_{ij} is stored inside the block with block coordinates (i/rows,j/columns) at the location (i%rows)*rows+(j%columns) inside the block. The block matrix addition of M and N is done by finding blocks from M and N with the same block coordinates and by adding the blocks together using regular matrix addition in Scala.

Your project is to convert two sparse matrices M and N which are read from files to block matrices using the Scala function `createBlockMatrix` and then add them using block matrix addition.

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 IDEA, go to New→Project from Existing Sources, then choose your project4 directory, select Maven, and the Finish. To compile the project, go to Run→Edit Configurations, use + to Add New Configuration, select Maven, give it a name (eg, build), use Working directory: your project4 directory, Command line: install, then Apply. To run your project in local mode, you need to add the line conf.setMaster("local[2]") in the main program before you create SparkContext (you should remove this line before you test your project on Expanse). Go to Run→Edit Configurations, use + to Add New Configuration, select Application, give it a name (eg, run), use the Main class: Add, Program arguments: `M-matrix-small.txt N-matrix-small.txt output`.

On Eclipse, you first need to install **m2e** (https://projects.eclipse.org/projects/technology.m2e) (Maven on Eclipse), if it's not already installed.
Then, inst
Project fro

Try Again

select Run As, and then Maven install. To run your project in local mode, you need to add the line conf.setMaster("local[2]") in the main program before you create SparkContext (you should remove this line before you test your project on Expanse). Right-click on Add.scala→Run As→Run Configurations, select Scala Application, press the New button to create a new configuration, give it a name (eg, run), add the main class Add, and go to Arguments. Add the arguments: `M-matrix-small.txt N-matrix-small.txt output`. Now you can run it in local mode by hitting Run.

Documentation

You can learn more about Scala at:

- **A Scala Tutorial for Java Programmers** (http://docs.scala-lang.org/tutorials/scala-for-java-programmers.html)
- **A Tour of Scala** (https://docs.scala-lang.org/tour/tour-of-scala.html)
- **Scala API** (https://www.scala-lang.org/api/current/scala/collection/immutable/index.html)

You can learn more about Spark at:

- **Spark Quick Start** (http://spark.apache.org/docs/latest/quick-start.html)
- **Spark Programming Guide** (http://spark.apache.org/docs/latest/programming-guide.html)
- **Spark by {Examples}** (https://sparkbyexamples.com/)
- **RDD API** (http://spark.apache.org/docs/latest/api/scala/org/apache/spark/rdd/RDD.html)
- **PairRDDFunctions API** (http://spark.apache.org/docs/latest/api/scala/org/apache/spark/rdd/PairRDDFunctions.html)

What to Submit

As in the previous projects, you need to tar your project4 directory on Expanse, copy project4.tgz from Expanse to your laptop, and submit it using this project page. Make sure that your project4 contains the files:

```
projeect4/src/main/java/Add.java
projeect4/add.local.out
projeect4/add.distr.out
projeect4/output-distr/part-r-00000
projeect4/output-distr/part-r-00001
```

Try Again

Preview Unavailable

project4.tgz

Download

(https://uta.instructure.com/files/23085730/download?download_frd=1&verifier=bMYxx9DxUtruA4MEW8mhjzwSz6Dh3k8kCojQe4Ak)

Try Again