EE 451: Parallel and Distributed Computation

PA3 — Spring 2021

Due date: Friday 5th March 2021 11:59 PM

1. **Login to the USC Discovery Cluster**

   - The host is: discovery.usc.edu
   - The username and password are the same as your USC account. You are able to use ssh to login to the cluster (remember to replace YOUR_USC_NET_ID with your actual usc net ID.):

     ```
     ssh YOUR_USC_NET_ID@discovery.usc.edu
     ```

   - **Do not** run your program in the login node
   - After login, use the 'srun' command to run your program on a remote node. For example:

     ```
     srun -n 1 ./<your executable>
     ```

2. **Spark Examples**

   To run a Spark python program, for example, the '`pi.py`' (the provided example used to calculate the pi (π) value), follow the steps:

   1. Login to the Discovery cluster
   2. Load the JDK needed by the Spark framework

      ```
      module load openjdk/1.8.0_202-b08
      ```

   3. Install Spark:

      ```
      wget https://ftp.wayne.edu/apache/spark/spark-2.4.7/spark-2.4.7-bin-hadoop2.7.
          tgz
      tar xvf spark-2.4.7-bin-hadoop2.7.tgz
      mv spark-2.4.7-bin-hadoop2.7 spark
      ```

   4. Run the example (Note that directly copying the command from the PDF file sometimes introduces some unnecessary space characters (depending on the PDF reader). You may need to manually delete these spaces so that the command can work correctly.):

      ```
      srun -n 1 ./spark/bin/spark-submit ./spark/examples/src/main/python/pi.py
      ```

   5. Expected output: There is a lot of console output of the Spark framework. If the program runs correctly, you can find a line similar to:

      ```
      Pi is roughly 3.145080
      ```

3. **Introduction**

   The objective of this assignment is to gain experience with programming using the MapReduce programming model [1] in Apache Spark Cluster programming framework [2]. Apache Spark supports SCALA, python and java as programming languages. This assignment uses python as the programming language. If you use any other language, please provide detailed instructions for running the program in your submission.

4. **K-means Clustering [20 points]**

   Based on the discussion slides, complete the `Map` (`mapToCluster`) and `Reduce` (`updatemeans`) functions (you are only allowed to modify these two functions) of `kmeans.py` to implement the K-means clustering algorithm under the MapReduce programming model [15 points]. Run the program and submit the produced output file (`kmeans_output`)[5 points] and your code.

   Note: In your K-means implementation, if a data point has multiple closest cluster centers, you should select the first one (the cluster center with the smallest index).

   You can use the following command to run your kmeans implementation (`data.txt` and `means.txt` are the provided standard inputs.):

   ```
   srun -n 1 ./spark/bin/spark-submit kmeans.py data.txt means.txt
   ```

5. **Triangle Counting [30 points]**

   Based on the discussion slides, write a program which uses the map/reduce functions in Apache spark to count the number of triangles in a graph. The input graph and the description of its format is provided in the file named: `p2p-Gnutella06.txt`.

   A python helper program named `readgraph.py` is provided which reads the input file and populates the nodes and edges to help you get started (you can run it using: `python readgraph.py`).

   Your program (Please name it as `trianglecounting.py`) should produce an output file (Name the output file as '`triangle_output`'.) which contains the number of triangles to which each vertex belongs to. [25 points]. Run the program and submit the code and the output file produced. [5 points].

   You should be able to run your program using the following command:

   ```
   srun -n 1 ./spark/bin/spark-submit trianglecounting.py p2p-Gnutella06.txt
   ```

6. **Submission**

   - Code: '`kmeans.py`' and the output file '`kmeans_output`'. (20 pts)
   - Code: '`trianglecounting.py`' and the output file '`triangle_output`'. (30 pts)

- Report: Write clearly how to compile and run your code. Report the screenshots of the execution on the Discovery cluster.

You may discuss the algorithms. However, the programs have to be written individually. Submit the code, produced files and the report via Blackboard. Make sure your program is runnable.

Note: You must use the Spark MapReduce programming model to implement the K-means and triangle-counting algorithms. Implementation using pure Python code without invoking the Spark framework is NOT acceptable.

References

[1] "MapReduce, http://static.googleusercontent.com/media/research.google.com/en/us/archive/mapreduce-osdi04.pdf

[2] "Apache Spark," https://spark.apache.org/