INFSCI 2750: Cloud Computing

Mini Project 2

Objective

The objective of this mini project is to get familiarized with Apache Spark. You will be working on the previously assigned VM's.

Part 1: Setting up Spark: (40 points)

The first task is to configure Spark distribution on top of the previously installed Hadoop cluster. Your setup should make use of YARN for scheduling/running Spark applications. The entire Spark setup should be configured on top of a three node Hadoop cluster.

You can go through the following link for deploying Spark in a standalone mode:
http://spark.apache.org/docs/latest/spark-standalone.html

Once you successfully set up Spark in standalone mode, the next step is to integrate it with the YARN manager.

Here are a few tutorials for integrating Spark with Yarn –
http://spark.apache.org/docs/latest/running-on-yarn.html

Part 2: Developing Spark programs (30 points)

As part of the project you will be working with the '*hetrec2011-lastfm-2k'* data set. In particular you will be working on the *'user_artists.dat'* data file.

The task is to printout the total listening counts of each artist. Your program should print out the listening counts per artist in descending order. In the 'user_artists.dat' file, listening counts for each user-artist pair is indicated by the variable 'weight'.

More details of the dataset could be found here –
http://grouplens.org/datasets/hetrec-2011/

Here are some examples for programming with Spark:
http://spark.apache.org/docs/latest/quick-start.html

Part 3: Developing Spark programs 2 (30 points)

As part of the project you will be working with the log data set which is provided in access_log.zip in the Mini Project 1.

In this project, you need to translate a python program which is running locally to a Spark program, which can be run on the Spark Cluster.

The python program is given within the project file, which is named "LinearRegressionExample.ipynb" which can be opened by jupyter notebook.

The program predicts the trend of accessing the website by a simple linear regression.

What you need to do can be separate to the following:

1. Test the local program (provided in "LinearRegressionExample.ipynb") on your own laptop by running a jupyter notebook, you can check:
    a. Install Anaconda (an all-in-one python environment) for python 3.7: https://www.anaconda.com/distribution/
    b. Run Jupyter notebook:

       https://jupyter.readthedocs.io/en/latest/running.html#running
2. Setup and test your spark cluster with running any of the Spark environment on your cluster (JAVA, Scala, or Python), reference: https://spark.apache.org/docs/latest/quick-start.html
3. Translate the program into a Spark program, you can choose one of the languages you want to use, which can be JAVA, Scala, or Python. The program should include at least the four parts below:
    a. Initial the SparkSession (or SparkContext) running on the Spark cluster:https://spark.apache.org/docs/latest/quick-start.html#self-contained-applications
    b. Load the "access_log" on HDFS into the Spark program as a DataFrame (DataSet): https://spark.apache.org/docs/latest/sql-data-sources-load-save-functions.html
    c. Group the access logs by the timestamp in month:
        i. JAVA: https://spark.apache.org/docs/latest/api/java/index.html?org/apache/spark/sql/Dataset.html
        ii. Scala: https://spark.apache.org/docs/latest/api/scala/index.html#org.apache.spark.sql.Dataset
        iii. Python: https://spark.apache.org/docs/latest/api/python/pyspark.sql.html#pyspark.sql.DataFrame
    d. Use the Spark MLlib to do the simple linear regression on the grouped result and save the model to a output file: https://spark.apache.org/docs/latest/mllib-linear-methods.html#linear-least-squares-lasso-and-ridge-regression

Project Submission: Submit a **single ZIP file** with your *Pitt email ID* as its filename via the CourseWeb system. The package should contain all your source files and a *readme* file that explains how to execute your program. Also include screenshots of your programs output and spark shell. The IP address of the master machine should be clearly visible in the screenshots.