A Graph Processing Program using GraphX

Description

The purpose of this project is to develop a graph processing program using Pregel on Spark GraphX.

This project must be done individually. No copying is permitted. **Note: We will use a system for detecting software plagiarism, called**[**Moss (Links to an external site.)**](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 Comet. Optionally, you may use your laptop to help you develop your program, but you should test your programs on Comet before you submit them.

Setting up your Project

Login into Comet and download and untar project8:

wget http://lambda.uta.edu/cse6331/project8.tgz

tar xfz project8.tgz

chmod -R g-wrx,o-wrx project8

Look at the example example/src/main/scala/SSSPExample.scala

Project Description

You are asked to re-implement Project #5 (Graph Processing) using Pregel on Spark GraphX. That is, your program will partition your graph into 10 clusters. An empty project8/src/main/scala/Partition.scala is provided, as well as scripts to build and run this code on Comet. **You should modify Partition.scala only**. **You should use the pregel method from the [GraphX Pregel API (Links to an external site.)](https://spark.apache.org/docs/latest/graphx-programming-guide.html" \l "graph-builders" \t "_blank) only to write your code**. Your main program should take the text file that contains the graph (small-graph.txt or large-graph.txt) as an argument and print the results to the output. The stopping condition is when the number of repetition reaches 5.

You can compile Partition.scala using:

run partition.build

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

sbatch partition.local.run

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 partition.distr.run

This will work on the moderate-sized graph and will print the results to the output.

The following pseudo-code to do graph clustering using Pregel:

1. Read the input graph and construct the RDD of edges
2. Use the graph builder Graph.fromEdges to construct a Graph from the RDD of edges
3. Access the VertexRDD and change the value of each vertex to be the -1 except for the first 5 nodes (these are the initial cluster number)
4. Call the Graph.pregel method in the GraphX Pregel API to calculate the new cluster number for each vertex and propagate this number to the neighbors. For each vertex, this method changes its cluster number to the max cluster number of its neighbors only if the current cluster number is -1.
5. Group the graph vertices by their cluster number and print the partition sizes (you can use Spark RDD methods like in Project 5)

Optional: Use your laptop to develop your project

If you'd prefer, you may use your laptop to develop your program and then test it and run it on Comet.

To install the project:

cd

wget http://lambda.uta.edu/cse6331/project8.tgz

tar xfz project8.tgz

To compile and run project8:

cd project8

mvn install

~/spark-1.5.2-bin-hadoop2.6/bin/spark-submit --class Partition --master local[2] target/cse6331-project8-0.1.jar small-graph.txt

Look at the example example/src/main/scala/SSSPExample.scala. You can compile and run it using:

cd example

mvn install

~/spark-1.5.2-bin-hadoop2.6/bin/spark-submit --class SSSPExample --master local[2] target/cse6331-project8-0.1.jar

Documentation

You can learn more about GraphX at:

* [GraphX Programming Guide (Links to an external site.)](https://spark.apache.org/docs/latest/graphx-programming-guide.html)
* [GraphX Pregel API (Links to an external site.)](https://spark.apache.org/docs/latest/graphx-programming-guide.html#pregel-api)
* [Graph Builders (Links to an external site.)](https://spark.apache.org/docs/latest/graphx-programming-guide.html#graph-builders)

What to Submit

Zip and submit your project8 directory, which must contain the following files:

project8/src/main/scala/Partition.scala

project8/partition.local.out

project8/partition.distr.out