COMPUTAÇÃO DE ALTO DESEMPENHO

2018/2019

Project 2 – Spark

OBJECTIVE

The purpose of this project is to identify the bottleneck in a routing graph and evaluate the impact of reducing the overhead of such bottleneck. **Note**: In this project you cannot use any existing graph processing library, such as GraphX.

You will apply your algorithm to the particular case of the flight's dataset used in Lab2.

IMPLEMENTATION STEPS

1. Build an undirected graph to represent the average delay of the flights between any two airports (the average must consider all flights between the both airports, disregarding the origin and the destination). The graph's nodes are thus the airports, and the edges represent direct routes between airports, labelled with the average delays of the routes' flights.

   Suggestion: represent the graph through an adjacency matrix. See https://spark.apache.org/docs/latest/mllib-data-types.html#distributed-matrix
   You will have to add the following dependency to file build.gradle:

   ```
   implementation 'org.apache.spark:spark-mllib_2.11:2.3.0'
   ```

2. Compute the graph's Minimum Spanning Tree (MST) by implementing the parallel version of Prim's Algorithm (available from CLIP). The MST will be the subgraph of the original with the minimum total edge weight (sum of the average delays). Output the MST and its total edge weight.

3. Identify the bottleneck airport, i.e. the airport with higher aggregated delay time (sum of the delays of all routes going out of the airport) from the ones contained in the complement graph of the MST previously computed.

4. Modify the graph to reduce by a given factor the delay time of all routes going out of the selected airport. This factor will be a parameter of your algorithm (received in the command line) and must be a value in]0, 1[.

5. Recompute the MST and display the changes perceived in the resulting subgraph and on the sum of the total edge weight.

SUBMISSION

The submission will have to include all the developed source code, as well as small report explaining your solution and presenting the execution times collected from the experiments carried out in the cluster.