Universite Paris Dauphine

Big Data

# Single-source shortest path - Djikstra Algorithm

*Students*
Elie Abi Hanna Daher
Bilal El Chami
Badr Erraji

*Professor*
Mr Dario Colazzo

January 30, 2018

# Contents

# 1 Project goal

The goal of the project is to find the shortest paths from a source node to all other nodes in the graph using the Dijkstra's algorithm. The algorithm should be implemented in both Python-Hadoop and Spark.

A long side the implementation, a scalability experiments is needed to check the performance of the algorithm implemented.

# 2 Dijkstra Algorithm

The Dijkstra's algorithm finds the shortest path from source to all other nodes. The dijkstra algorithm is very similar to the BFS algorithm, the only difference is that the distance between neighbors isn't 1, distance can differ from a neighbor to another.

# 3 Implementation

## 3.1 Input

### 3.1.1 Data

The map task should receive the following infromation

- node
- distance
- neighbors data that contains the list of neighbors with their respective distance to the node
- path

So let's take the following example with 1 being the start node :

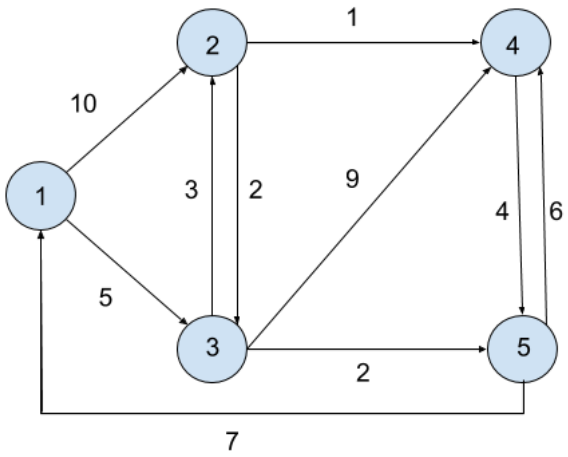


Figure 1: Example of a graph

For the first iteration, the path will be empty. So input data will look like this :

- 1 0 2,10:3,5:
- 2 999 3,2:4,1:
- 3 999 2,3:4,9:5,2:
- 4 999 5,4:
- 5 999 1,7:4,6:

So as you can see, the start node has a distance of 0, and all other node have a distance of 999 which represent an infinite number. The neighbor list contains each neighbor node with their respective distance, the neighbors and the distance are seperated by a "," and neighbors are seperated by ":".

#### 3.1.2 Prepare

Technically, the format we set for the data is very hard to implement to a large graph. Usually the graph is represented by the distance between nodes. So for the graph provided in the previous page, the input data will look like this

- 1 2 10
- 1 3 5
- 2 3 2
- 2 4 1
- 3 2 3
- 3 4 9
- 3 5 2
- 4 5 4
- 5 1 7
- 5 4 6

We created a job MapReduce - called prepare that will format the usual fromat of a graph to the format we are asking for.

### 3.2 Mapper

A map task receive (K,V)

- Key : node
- Value : distance, neighbors data, path

In the first iteration, the path will be empty.
The map task will :

1. emit the node with his information (distance, neighbors data and path)
2. $\forall$ neighbor $\in$ neighbors, it will emit
   - Key : neighbor
   - Value : (node distance + distance of node to the neighbor , node path + neighbor).

The pseudo code of the mapper is as follow :

**Algorithm 1** Mapper

```
1: procedure MAP (node, (distance, neighbors, path))
2:     Emit (node, (distance, neighbors, path))
3:     for all neighbor ∈ neighbors do
4:         dist ← distance + neighbor.distance
5:         p ← path + neighbor.id
6:         Emit (neighbor.id, (dist, p))
```

### 3.3 Reducer

The reduce will gather the possible distance for each node and selects the minimum one and set the path of the minimum one selected

### 3.4 Job Chaining

## 4 Results

### 4.1 Hadoop

### 4.2 Spark

## 5 Performance

## References

[1] *Cloud Computing Lecture 4 - Graph Algorithms with MapReduce.* Jimmy Lin, The iSchool, University of Maryland, February 6, 2008.

# A Appendix example

This an example of appendix