Md Jamiur Rahman Rifat
Zhicheng Luo

Distributed Graphs Processing

BDMA - Semantic Data Management - Lab 2

27 April 2022

Exercise 1:

Superstep 0:

VertexID: 2, VertexValue: 1, Received Message: 2147483647
VertexID: 3, VertexValue: 6, Received Message: 2147483647
VertexID: 1, VertexValue: 9, Received Message: 2147483647
VertexID: 4, VertexValue: 8, Received Message: 2147483647
No message sent from VID2 to VID4
No message sent from VID3 to VID1
Message sent from VID1 to VID2 is 9
No message sent from VID3 to VID4
No message sent from VID2 to VID3

Superstep 1:

VertexID: 2, VertexValue: 1, Received Message: 9 Changed value of the vertex: 9
Message sent from VID2 to VID3 is 9
Message sent from VID2 to VID4 is 9

Superstep 2:

VertexID: 3, VertexValue: 6, Received Message: 9 Changed value of the vertex: 9
VertexID: 4, VertexValue: 8, Received Message: 9 Changed value of the vertex: 9
No message sent from VID3 to VID4
No message sent from VID3 to VID1

9 is the maximum value in the graph

Exercise 2:

In the solution, each node stores the value of the length of shortest path from node 1, which is initialized as *Integer.MAX_VALUE* at the beginning accepting node 1. The message is defined as the value of the source node plus the edge length. It will be sent to the destination node only if the sum is less than the value of the destination node, which means that a shorter path to the destination node from node 1 through the source node is found.

We assume that there is no edge having a negative value or length in the given graph.

We follow the code schema in the Java project and implement the following three functions:

VProg: in the case of the first superstep the vertex value, otherwise sends the minimum of the vertex value and the received message.

sendMsg: if the destination node value is smaller than the current value plus the edge length, which represents the length of a shorter path, we will send the sum to it. Otherwise, we do not send anything.

Merge: in this case, the logic for the function is the same as the vertex program.

The result is as below:

```
Minimum cost to get from A to D is 9
Minimum cost to get from A to C is 2
Minimum cost to get from A to A is 0
Minimum cost to get from A to E is 5
Minimum cost to get from A to F is 20
Minimum cost to get from A to B is 4
```

Exercise 3:

In this exercise, we extend the solution of exercise 2 to compute the length of the shortest path as well as record all the nodes in it.

We extend the value of nodes as a 2-item tuple. The first item is the length of the shortest path from node 1, the same as exercise2. The second item is a list of all nodes in the shortest path which the program has discovered so far.

Accordingly, in each superstep, when we find a shorter path to a node, we not only update the length, but the list of nodes in the path. More specifically, in the function **sendMsg**, the message includes not only the length of a shorter path, but the node lists of it, which is concatenated by the node lists of the shortest path to the source node and the destination node.

The result is as below:

```
Minimum cost to get from A to D is [A, C, E, D] with cost 9
Minimum cost to get from A to F is [A, C, E, D, F] with cost 20
Minimum cost to get from A to B is [A, B] with cost 4
Minimum cost to get from A to A is [A] with cost 0
Minimum cost to get from A to E is [A, C, E] with cost 5
Minimum cost to get from A to C is [A, C] with cost 2
```

Exercise 4:

In this exercise we got the top 10 most relevant articles from the Wikipedia dataset. The graphX implementation of pagerank algorithm requires two parameter value one is the damping factor and the other is maximum iteration. To the best of our knowledge there is no straightforward way to calculate these values therefore we have taken a small search space (for the limitation of our computation capacity) and try to run the algorithm by different values of the damping factor and number of iterations. The combination at which we are getting the less deviation of the sum of the values of page rank of the top 10 articles, we are considering that combination value (the damping factor and maxiteration) as the optimal one.

percent change with iteration 9 damp factor 0.75 value 0.0034756136048683943
percent change with iteration 10 damp factor 0.75 value 0.0023169174817417155
percent change with iteration 11 damp factor 0.75 value 6.283993633341976E-4
percent change with iteration 12 damp factor 0.75 value 5.170841179172552E-4
percent change with iteration 13 damp factor 0.75 value 1.1680319881073486E-4
percent change with iteration 14 damp factor 0.75 value1.2289215479847231E-4
percent change with iteration 8 damp factor 0.85 value 0.05893320355204478
percent change with iteration 9 damp factor 0.85 value 0.015742044066214548
percent change with iteration 10 damp factor 0.85 value 0.010879178284041384
percent change with iteration 11 damp factor 0.85 value 0.003979593062433455
percent change with iteration 12 damp factor 0.85 value 0.0032522897774500217
percent change with iteration 13 damp factor 0.85 value 0.0010635262240938116
percent change with iteration 14 damp factor 0.85 value 0.0010330504589702288
percent change with iteration 8 damp factor 0.9 value 0.015567556798163955
percent change with iteration 9 damp factor 0.9 value 0.03462279121919512
percent change with iteration 10 damp factor 0.9 value 0.024067981381421095
percent change with iteration 11 damp factor 0.9 value 0.010049363788162619
percent change with iteration 12 damp factor 0.9 value 0.00820660463107004
percent change with iteration 13 damp factor 0.9 value 0.0030680297215926744
percent change with iteration 14 damp factor 0.9 value 0.002938991385027324
best iteration = 13 best damping factor = 0.75

id	name	pagerank
8830299306937918434	University of Cal...	2979.566632708042
1746517089350976281	Berkeley, California	1466.880525725874
8262690695090170653	Uc berkeley	362.5724717214202
7097126743572404313	Berkeley Software...	192.57459673480076
1735121673437871410	George Berkeley	180.59421308969954
8494280508059481751	Lawrence Berkeley...	175.07637525390592
6990487747244935452	Busby Berkeley	104.66810652882634
1164897641584173425	Berkeley Hills	81.90638960645269
5820259228361337957	Xander Berkeley	69.26939614374034
1630393703040852365	Berkeley, CA	62.832346028421995