Assignment of Homework 2 (DEADLINE: May 22, 23:59)

In this homework, you will run a Spark program on the CloudVeneto cluster. As for Homework 1, the objective is to estimate (approximately or exactly) the number of triangles in an undirected graph $G = (V, E)$. More specifically, your program must implement two algorithms:

ALGORITHM 1. The same as Algorithm 1 in Homework 1, so you must recycle method/function MR_ApproxTCwithNodeColors devised for Homework 1, fixing any bugs that we have pointed out to you.

ALGORITHM 2. A 2-round MapReduce algorithm which returns the exact number of triangles. The algorithm is based on node colors (as Algorithm 1) and works as follows. Let $C \geq 1$ be the number of colors and let $h_C(\cdot)$ ne the hash function that assigns a color to each node used in Algorithm 1.

Round 1

- For each edge $(u, v) \in E$ separately create C key-value pairs $(k_i, (u, v))$ with $i = 0, 1, \ldots C - 1$ where each key k_i is a triplet containing the three colors $h_C(u), h_C(v), i$ sorted in non-decreasing order.
- For each key $k = (x, y, z)$ let L_k be the list of values (i.e., edges) of intermediate pairs with key k. Compute the number t_k of triangles formed by the edges of L_k whose node colors, in sorted order, are x, y, z. Note that the edges of L_k may form also triangles whose node colors are not the correct ones: e.g., (x, y, y) with $y \neq z$.

An example of Round 1 is given in this picture.

Round 2. Compute and output the sum of all t_k 's determined in Round 1. It is easy to see that every triangle in the graph G is counted exactly once in the sum. You can assume that the total number of t_k 's is small, so that they can be garthered in a local structure. Alternatively, you can use some ready-made reduce method to do the sum. Both approaches are fine.

Using the cluster

A brief description of the cluster available for the course, together with instructions on how to access the cluster and how to run your program on it are given in this User guide for the cluster on CloudVeneto.

TASK for HW2:

1) Fix bugs (if any) of method/function MR_ApproxTCwithNodeColors written for HW1, which implements **ALGORITHM 1**.

2) Write a method/function MR_ExactTC which implements **ALGORITHM 2**. Specifically, MR_ExactTC must take as input an RDD of edges and the number of colors C, and must return the exact triangle count.

- **Hint (for Java users).** To represent triplets of colors, you can use the scala type Tuple3<Integer,Integer,Integer>, importing scala.Tuple3 at the beginning of your code.
- **Hint.** In Round 1, in order to compute the number of triangles for each key $k = (x, y, z)$ you can run on the set of edges L_k a simple modification of method/function CountTriangles (the one used by MR_ApproxTCwithNodeColors and provided by us for HW1) which before incrementing the count for a new triangle, checks if the colors of its 3 nodes are x, y, z. You can use the the following code for the modified method/function: CountTriangles2.java and CountTriangles2.py. Note that our code receives in input the parameters a,b,p and C that define the hash function.

3) Write a program GxxxHW2.java (for Java users) or **GxxxHW2.py** (for Python users), where xxx is your 3-digit group number (e.g., 004 or 045), which receives in input, as command-line arguments, 2 integers C and R, and a path to the file storing the input graph, and does the following:

- Reads parameters C and R
- Reads the input graph into an RDD of strings (called **rawData**) and transform it into an RDD of edges (called **edges**), represented as pairs of integers, partitioned into 16 partitions, and cached.
- Prints: the name of the file, the number of edges of the graph, C, and R
- Runs R times **MR_ApproxTCwithNodeColors** to get R independent estimates of the number of triangles in the input graph.
- Prints: the median of the R estimates returned by **MR_ApproxTCwithNodeColors** and the average running time of **MR_ApproxTCwithNodeColors** over the R runs.
- Runs R times **MR_ExactTC** to get the exact number of triangles in the input graph
- Prints: the last value returned by **MR_ExactTC** (they are all equal) and the average running time over the R runs.

File (**TO BE ADDED**) shows how to format your output. Make sure that your program complies with this format.

4) Test and debug your program in local mode on your PC *to make sure that it runs correctly. The program must be stand-alone in the sense that it should run without requiring additional files.*

5) Test your program on the cluster using the datasets which have been preloaded in the HDFS available in the cluster. Use various configurations of parameters and report your results using the tables given in this word file: TableHW2-Java.docx (for Java users) and TableHW3-Python.docx (for Python users). (TO BE ADDED)

WHEN USING THE CLUSTER, YOU MUST STRICTLY FOLLOW THESE RULES:

- **To avoid congestion, groups with even (resp., odd) group number must use the clusters in even (resp., odd) days.**
- **Do not run several instances of your program at once.**
- **Do not use more than 16 executors.**
- **Try your program on a smaller dataset first.**
- **Remember that if your program is stuck for more than 1 hour, its execution will be automatically stopped by the system.**

SUBMISSION INSTRUCTIONS. Each group must submit a **zipped folder GxxxHW2.zip**, where xxx is your group number. The folder must contain the program (**GxxxHW2.java** or **GxxxHW2.py**) and a file **GxxxHW2table.docx** with the aforementioned table. Only one student per group must do the submission using the link provided in the Homework 2 section. Make sure that your code is free from compiling/run-time errors and that you comply with the specification, otherwise your grade will be penalized.

*If you have questions about the assignment, contact the teaching assistants (TAs) by email to bdc-course@dei.unipd.it . The subject of the email must be "**HW2 - Group xxx**", where xxx is your group number. If needed, a zoom meeting between the TAs and the group will be organized.*

Last modified: Wednesday, 3 May 2023, 2:46 PM

◂ User Guide for the Cluster provided by Cloud Veneto

Jump to...