**BIG DATA COMPUTING 2019-20 – HOMEWORK 3 – GROUP 28**

**(USE ONLY 1 PAGE)**

1. **Required tests.** Do the tests with the parameters indicated in the first 4 columns of the following table. Fill the table reporting, for each test, the following values: **Init** = *time to read the input and create the RDD*;**Ti** = *time of Round i of runMapReduce* (i=1,2), **AvgDist** = *average distance among the solution points*. **Times should be in ms.** If you notice anomalies in the values of a test try to repeat the test 3 times and take the median values*.*

*Some anomalies are to be expected!*

**Table to be filled by Java users**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Dataset** | **K** | **L** | **num-executors** | **Init** | **T1** | **T2** | **AvgDist** |
| Glove2M300d.txt | 100 | 16 | 16 | 21980 | 12544 | 18613 | 29.040733847821866 |
| **Glove2M300d.txt** | **100** | **16** | **8** | 35461 | 22545 | 20626 | 29.040733847821866 |
| **Glove2M300d.txt** | **100** | **16** | **4** | 54876 | 36559 | 21322 | 29.040733847821866 |
| **Glove2M300d.txt** | **100** | **4** | **4** | 65618 | 63583 | 1078 | 29.042145462674455 |

**Table to be filled by Python users**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Dataset** | **K** | **L** | **num-executors** | **Init** | **T1** | **T2** | **AvgDist** |
| **Glove200K300d.txt** | **30** | **16** | **16** |  |  |  |  |
| **Glove200K300d.txt** | **30** | **16** | **8** |  |  |  |  |
| **Glove200K300d.txt** | **30** | **16** | **4** |  |  |  |  |
| **Glove200K300d.txt** | **40** | **16** | **16** |  |  |  |  |

1. **Observations:** *write here a brief comparison among the results of the experiments reported in the table (and perhaps of other experiments that you did on your own), and try to justify the differences that you may have observed.*

We have observed that number of executors and T1 are inversely proportional: a high number of executors allow for a smaller T1 execution time. We think that this is due to a parallelism increase in the execution of Farthest-First Traversal (FFT), since FFT requires O(N\*k) time and more partitions means smaller partitions, so each FFT instance is faster.

With respect of T2, it is less sensible to the variation of the number of executors, since the algorithm works on the coreset returned by FFT, which is sensible smaller than N (the size of the initial pointset).

The obtained AvgDistance is constant among all configuration tested, except for an outlier (L=4, num-executors=4), which provides a slightly different result. In this instance we also noticed a strange T2 value that we think is due to a sensible smaller number of coreset returned by FFT.