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

**(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** |  |  |  |  |
| **Glove2M300d.txt** | **100** | **16** | **8** |  |  |  |  |
| **Glove2M300d.txt** | **100** | **16** | **4** |  |  |  |  |
| **Glove2M300d.txt** | **100** | **4** | **4** | *You will not be able to run this test* | | | |

**Table to be filled by Python users**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Dataset** | **K** | **L** | **num-executors** | **Init** | **T1** | **T2** | **AvgDist** |
| **Glove200K300d.txt** | **30** | **16** | **16** | **623** | **11** | **246887** | **26.308** |
| **Glove200K300d.txt** | **30** | **16** | **8** | **597** | **10** | **278939** | **26.308** |
| **Glove200K300d.txt** | **30** | **16** | **4** | **568** | **10** | **320101** | **26.308** |
| **Glove200K300d.txt** | **40** | **16** | **16** | **561** | **17** | **524175** | **25.75** |

1. **Observations:**

* Average Distance: K and L seem to influence the final result of measuring the average distance. More specifically, with K = 30 and L = 16, the number of executors (as we expected) did not influence the computed average distance. Parallelism only speeds up things, and usually does not provide a different solution. Changing K to 40 (still, as expected) resulted in a decrease of the average distance. This is probably due to the fact that we are selecting a higher number of final points, thus having a higher chance of these points being nearer to one another.
* Elapsed Times: T1 elapsed times only consider invoking the *mapPartitions* function on the input RDD, and since the implementation resides in the Spark framework, we can expect these times to be short and the mapping function to be overall efficient, regardless of the number of executors. There is a slight increase setting K = 40, but it’s to be expected since each partition needs to extract more points to then produce the coreset by collecting the resulting points over the RDD. On the other hand, we can observe how the number of executors affects the computation on T2. The sequential algorithm takes quite some time to compute, and the elapsed times increase with less executors, up to approximately 5 minutes. Again, there’s a spike on the last experiment beacuse of the increased K, taking approximately 9 minutes. This is dictated by the quadratic complexity of the sequential algorithm more than the number of executors (which is raised back to 16).