Big Data Computing - 4^{th} Homework Report

Boem Davide, ID: 1176946, davide.boem@studenti.unipd.it
Boscaro Nicola, ID: 1181356, nicola.boscaro.1@studenti.unipd.it
Faccin Dario, ID: 1177736, dario.faccin@studenti.unipd.it*

In this report we described the results obtained by running our code, on *Cloud Veneto*, for the 4^{th} homework.

1 Results

In Table 1 you can see what we have obtained running our code on *Cloud Veneto*[1].

	Cores used by application	**Cores used for each executor**	**numBlocks**	**k**	**Coreset construction (ms)**	**Computation final solution (ms)**	**Average distance**	**Dataset (Approximate size)**
1	20	4	12	10	26519	24	10,5483	
2	20	4	12	20	25519	40	9,9642	
3	20	4	12	30	23942	70	9,7861	
4	20	4	12	40	25159	155	9,6639	
5	20	4	12	50	23727	297	9,5462	all
6	20	4	12	60	30821	449	9,4961	
7	20	4	12	70	25678	726	9,3317	
8	20	4	12	80	27404	1060	9,3204	
9	20	4	12	90	37561	1527	9,2578	
10	20	4	12	100	32148	2122	9,1879	

Table 1: Results obtained on *Cloud Veneto*, using dataset vectors-50-all.txt.bz2 and changing k.

	Cores used by application	**Cores used for each executor**	**numBlocks**	**k**	**Coreset construction (ms)**	**Computation final solution (ms)**	**Average distance**	**Dataset (Approximate size)**
1	20	4	12	10	12636	43	10,2092	
2	20	4	12	20	9732	103	9,5775	
3	20	4	12	30	14897	109	9,2798	
4	20	4	12	40	12534	172	9,0856	
5	20	4	12	50	14074	484	8,9951	2000000
6	20	4	12	60	18627	455	8,9671	
7	20	4	12	70	23357	876	8,9588	
8	20	4	12	80	24393	1141	8,9301	
9	20	4	12	90	30550	1790	8,8520	
10	20	4	12	100	22076	2093	8,8300	

Table 2: Results obtained on *Cloud Veneto*, using dataset vectors-50-2000000.txt.bz2 and changing k.

*Contact email
[1]We decided to round the average distances to 4 decimal places.

2 Plots

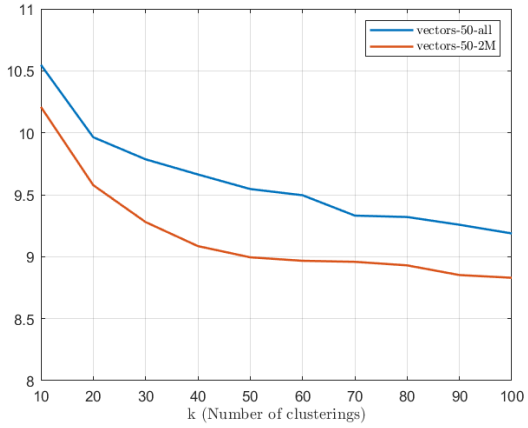
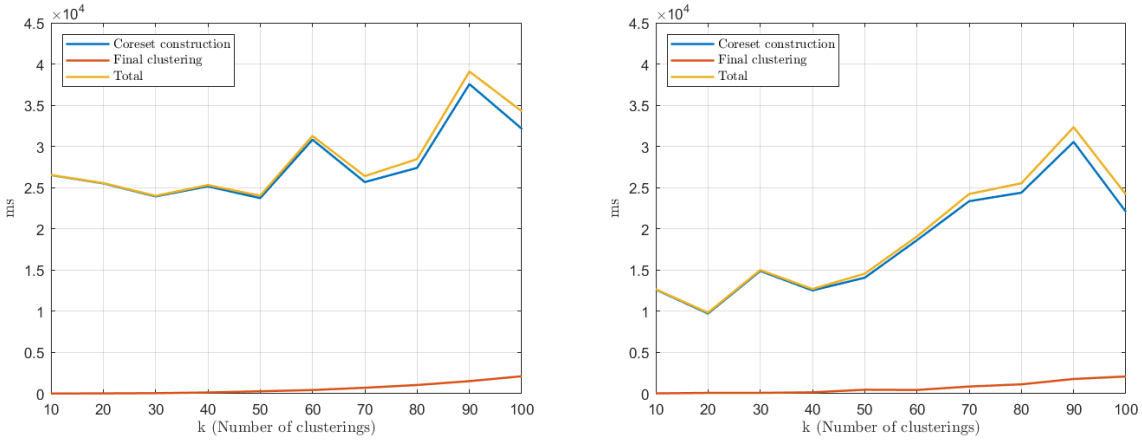


Figure 1: Max diversity distance computed for two datasets.



(a) Times for 5 millions vectors.

(b) Times for 2 millions vectors.

Figure 2: Execution times for two datasets.

3 Conclusions

Looking at the results (see Table 1), we can see that, while the measured time for the *Construction of the coreset* remains almost inside a time interval, the time spent by the program for the *Computation of the final solution* rises proportionally with the values of k and of $numBlocks$.
On the contrary, the *Average distance* starts with bigger values and converges after a while to lower values.
In the case of the total number of cores used by the application (X) we observed that there was a little improvement in the measured time of the *Construction of the coreset*, meanwhile, modifying the number of cores used for each executor (Y) we didn't see any enhancement.