

CLE - Assignment 2

Eduardo Santos - 93107

Pedro Bastos - 93150

Problem 1

This problem's solution was made using the MPI library, and consists in:

- The **main** method will process the command line input, and start the **dispatcher**, that signals the workers that there is work to be done, reads chunks of the files and sends them to the workers.
- The **workers** will receive the signal to work and compute the number of words, words starting with a vowel and words ending with a consonant, sending the partial results back to the **dispatcher** via a struct.
- The **dispatcher** will store each partial result, received in a *struct*.
- The above will continue, until there are no more work to be done, and then signal the workers to finish.
- Finally, the **dispatcher** prints the final results of each file.

Problem 2

This problem's solution was made using the MPI library, and consists in:

- The **main** method will process the command line input, and start the **dispatcher**, this will produce chunks of data from the binary matrices files, sending those chunks to the **workers**.
- The **workers** will compute the determinant of the respective matrix (from the chunk), sending the result back to the **dispatcher**.
- The **dispatcher** will store the results received in a *struct*.
- The above will continue, until there are no more work to be done, i.e., there are no more matrices to compute its determinant. When this happens, the **dispatcher** will tell the **workers** to end.
- Finally, the **dispatcher** will print the results.

Execution Times - Problem 1

As we can see, when working with parallel computing, there is a noticeable reduction on the execution time.

Although that is true, there is a slight increase from 4 workers to 8.

A possible explanation is the overhead of computing power, because as the problems are not that complex, not always more processes means less execution time.

CPU: 2,6 GHz Quad-Core Intel i7

# of Workers	Execution time (mean of 10 executions)	Standard deviation
2	0.013380 s	0.001243 s
4	0.008812 s	0.000912 s
8	0.010400 s	0.001012 s

Execution Times - Problem 2

As we can see, when working with parallel computing, there is a noticeable reduction on the execution times, although, as we can observe when comparing the results with **4** vs. **8 workers** from both the **mat128_32.bin** and **mat512_32.bin** files, there is an increase on the time, a possible explanation is that this happens due to an overhead of computing power.

The problems are not that complex, which means that not always more parallel processes means less execution time.

CPU: 2,8 GHz Quad-Core Intel Core i7

File	# of Workers	Execution time (mean of 10 executions)	Standard deviation
mat128_32.bin	2	0.0042711s	0.0003793
mat128_32.bin	4	0.0017748s	0.0001386
mat128_32.bin	8	0.0017954s	0.0002248
mat128_256.bin	2	0.5330135s	0.0172756
mat128_256.bin	4	0.2144752s	0.0046446
mat128_256.bin	8	0.1642803s	0.0082717
mat512_32.bin	2	0.0099704s	0.001398
mat512_32.bin	4	0.004722s	0.0002638
mat512_32.bin	8	0.0050324s	0.0004696
mat512_256.bin	2	1.9918162s	0.0246249
mat512_256.bin	4	0.8061925s	0.0127324
mat512_256.bin	8	0.6073243s	0.0097144