Trabalho 2 de CLE

Grupo 9 TP2
Diogo Andrade 89265
Francisco Silveira 84802

Introdução

Usando a implementação do primeiro trabalho, substituímos a implementação Multithread para Multiprocessing Passing Message, usando a biblioteca MPI.

Para isso é necessário pelo menos dois processos:

- um dispatcher que é responsável por ler os ficheiros, passar esses dados ao worker e receber o resultado do processamento e guardá-los;
- e um ou mais workers são responsáveis por processar os dados os ficheiros.

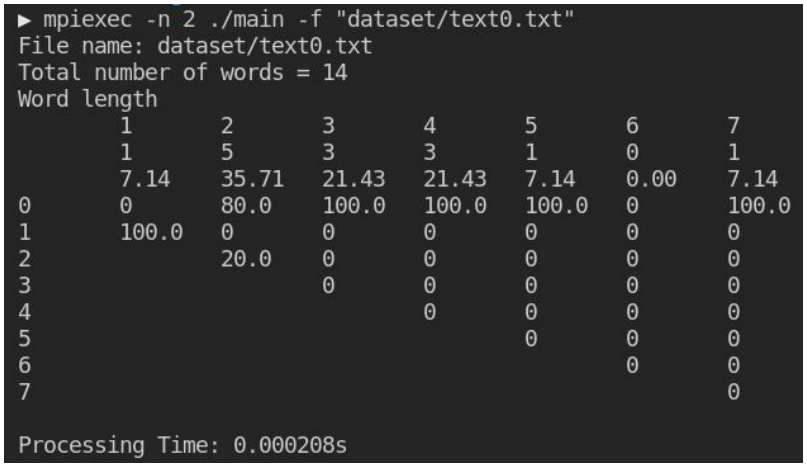
Exercício 1 - Explicação

O workers é responsável por ler e processar um bloco de dados de um ou mais ficheiros. Cada bloco de dados é uma array de tamanho 1024, que contém um conjunto de palavras completas, no caso de não ser possível completar uma palavra, o ponteiro do ficheiro voltará à posição anterior à palavra.

O worker irá contar o número de palavras, como também o número de consoantes tendo em conta o tamanho da palavra.

O acesso aos ficheiros e a junção dos resultados dos vários workers é feito pelo dispatcher.

Exercício 1 - Resultados



```
▶ mpiexec -n 2 ./main -f "dataset/text0.txt"
File name: dataset/text0.txt
Total number of words = 14
Word length
        1       2       3       4       5       6       7
        1       5       3       3       1       0       1
        7.14    35.71   21.43   21.43   7.14    0.00    7.14
0       0       80.0    100.0   100.0   100.0   0       100.0
1       100.0   0       0       0       0       0       0
2               20.0    0       0       0       0       0
3                       0       0       0       0       0
4                               0       0       0       0
5                                       0       0       0
6                                               0       0
7                                                       0

Processing Time: 0.000208s
```

Figura 1: Exemplo de execução.

Número de Workers	Tempo
1	0.013316s
2	0.010563s
4	0.013895s

Figura 2: Teste de tempo (Média de 5 tentativas, com o dataset inteiro).

Exercício 2 - Explicação

O worker é responsável por processar um conjunto de pares de sinais, ou seja, irá calcular a “circular cross-correlation” de cada par de sinais.

Após feitos todos os cálculos, quando imprimido os resultados o output irá dizer se os resultados obtidos são o esperado ou não.

O acesso aos ficheiros e a junção dos resultados dos vários workers é feito pelo dispatcher.

Exercício 2 - Resultados

```
▶ mpiexec -n 2 ./main -f "dataset/newSigVal01.bin
 dataset/newSigVal02.bin dataset/newSigVal03.bin
dataset/newSigVal04.bin"
File name: dataset/newSigVal01.bin
        Result: Expected result founded!
File name: dataset/newSigVal02.bin
        Result: Expected result founded!
File name: dataset/newSigVal03.bin
        Result: Expected result founded!
File name: dataset/newSigVal04.bin
        Result: Expected result founded!
Processing Time: 0.596447s
```

Figura 3: Exemplo de execução.

Número de Workers	Tempo
1	0.605956s
2	0.310584s
4	0.216733s

Figura 4: Teste de tempo (Média de 5 tentativas, com o dataset inteiro).

Conclusão

No primeiro exercício houve uma melhoria nos tempos de execução em relação ao primeiro trabalho, já no segundo isso já não se verificou (nós pensamos que isto acontece porque são necessárias trocas de 4 a 5 mensagens entre o dispatcher e o worker, em cada interação).

No segundo exercício verificou-se que quanto maior o número de workers menores são os tempos de execução, o que faz sentido tendo em conta que o número de trabalhos a realizar.