

# Relatorio PPC – Atividade1

## DNA Search



Ciências  
ULisboa

Henrique Barata fc54387

### How did you parallelize the problem?

Para resolver este problema em paralelo, dividi a sequência de ADN pelo numero de cores. Teria de ter atenção para quando esta divisam fosse feita, não separar nenhum padrão que pudesse existir, por exemplo o padrão “ac” na sequencia dividida “aaaccc” em duas partes, “aaa” e “ccc”, não seria contabilizado, mas a função dada (isIn) já resolvia esse problema.

### Did you have concurrency issues? Why or why not?

Para guardar os diferentes resultados das diferentes threads, criei uma matriz onde cada linha seria para cada thread e as colunas seriam as posições onde a frequência de cada padrão seria guardado. No fim somei os valores todos. Assim não encontrei nenhum problema de concorrência, pois este já tinha sido previsto.

### Resultados Obtidos

Threads	1	2	4	8	Media(ns)
Seq (ns)	8450	7957,143	7630	6360	7599,2857
Par (ns)	454700	685490	1007140	2166800	1078532,5
Speed-up	0,018584	0,011608	0,007576	0,002935	
Occupancy	0,018584	0,005804	0,001894	0,000367	

### What was the speed-up with 4 cores? And with 8 cores?

Com o programa a correr sequencialmente, não houve grande diferença, o que foi o esperado devido ao código ter sido corrido apenas com a main thread.

Com o programa a correr em paralelo, em comparação a correr sequencialmente, foi mais demorado. Assim nesta situação não seria indicado correr o programa em paralelo, pois a criação de threads e o resto do código necessário para a sua execução, consome mais tempo de processamento do que o sequencialmente.

Excluiu-se o primeiro resultado, pois deu um valor muito diferente dos outros para 1 core.

## What was the occupancy with 4 cores? And with 8 cores?

Para 4 cores = 0,001894 e para 8 cores =0,000367.

### Função P3Huge

Criei uma função onde aumentei o número de padrões e a dimensão do número de elementos na sequência de ADN, e aí já notei uma diferença significativa e mais rápida no processamento em paralelo. Para este caso já compensa resolver o problema em paralelo, sendo a mais rápida com 4 Threads.

Threads	1	2	4	8
Par (ns)	4385436550	2977022001	2664169600	2776399499