



**Ciências  
ULisboa**

Faculdade  
de Ciências  
da Universidade  
de Lisboa

## RELATÓRIO DO PRIMEIRO EXERCÍCIO DE PROGRAMAÇÃO PARALELA E CONCORRENTE

- Como foi paralelizado o problema? Primeiramente fiz a divisão da lista de caracteres correspondente à sequência total DNA em partes iguais, sendo essas partes definidas a partir do número de Threads utilizadas para fazer o processamento. Com isto, foi atribuída a cada Thread uma das partes resultantes e o mesmo conjunto de tarefas, começando assim por percorrer cada elemento da parte que lhe foi atribuída onde cada um desses elementos passa por um processo de avaliação em que verificamos, para cada padrão da lista 'patterns', se existe alguma subsequência, com início no elemento avaliado, que corresponde ao padrão em questão. Se sim, incrementamos a contagem da quantidade de padrões identificados na sequência DNA referente a esse mesmo padrão.
- Existiu problema de concorrência? Não, porém, foi identificado que quando, em cada Thread, ao incrementar, em uma lista de resultados única, o número de padrões identificados na parte da sequência atribuída a essa Thread, se pode observar a inconsistência de valores imprevisíveis que não correspondem ao resultado esperado. Isto acontece, possivelmente, devido a cada Thread aceder ao espaço de memória e fazer a sua alteração "por cima" de um valor desatualizado. Para evitar o uso da palavra-chave "synchronized", decidi resolver este problema ao fazer o incremento em espaços de memória distintos para cada Thread em uma matriz bidimensional, onde cada linha corresponde aos resultados obtidos em cada Thread. No final bastou somar cada coluna dessa mesma matriz e colocar o resultado em cada elemento de uma matriz unidimensional.

- Como foi o speed-up com 8 e 64 núcleos?

$$\begin{aligned} \text{(8 núcleos) speed-up} &= T_{\text{sequencial}} / T_{\text{paralelo}} \\ &= 250200,2 / 4651263,167 \approx 0,05379 \end{aligned}$$

$$\begin{aligned} \text{(64 núcleos) speed-up} &= T_{\text{sequencial}} / T_{\text{paralelo}} \\ &= 250272,0333 / 34454654,03 \approx 0,00726 \end{aligned}$$

O speed-up demonstra um resultado extremamente negativo, o que significa que a paralelização não é justificada com fins de aumentar a velocidade de processamento do problema. O programa ficou apenas muito mais lento e desnecessariamente mais complexo.

A minha conclusão é que, pela simplicidade do problema e/ou pelas listas de sequências a serem testadas terem uma quantidade reduzida de elementos (incluindo o exemplo de 9000 elementos), a solução sequencial não tem grandes problemas em resolver o problema. Devido à criação de Threads ser demasiado custosa e pouco recompensadora para um problema em que a solução sequencial não tem problemas de desempenho, a solução paralelizada tem uma notável perda de desempenho à medida que é aumentada a criação de Threads, sendo este o fator maioritário para a divergência de resultados.

- Como foi a ocupância com 8 e 64 núcleos?

$$\begin{aligned} \text{(8 núcleos) Ocupância} &= \text{speed-up} / \#\text{núcleos} \\ &= 0,05379 / 8 \approx 0,00672375 \approx 0,67\% \end{aligned}$$

$$\begin{aligned} \text{(64 núcleos) Ocupância} &= \text{speed-up} / \#\text{núcleos} \\ &= 0,00726 / 64 \approx 0,0001134375 \approx 0,0113\% \end{aligned}$$

NOTA: Cálculos feitos a partir dos resultados dos testes com a sequência grande DNA.

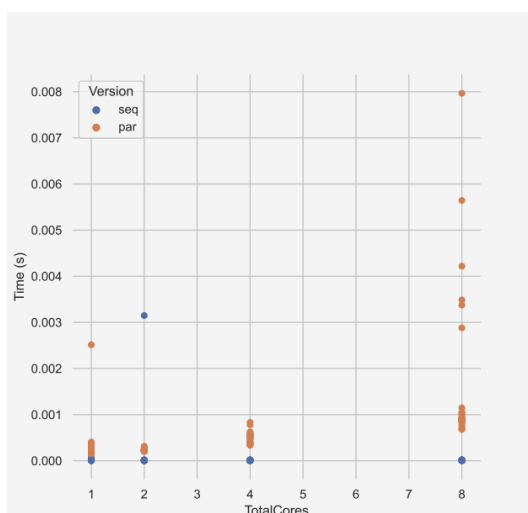


Figura 1 - Resultados com a sequência pequena de DNA

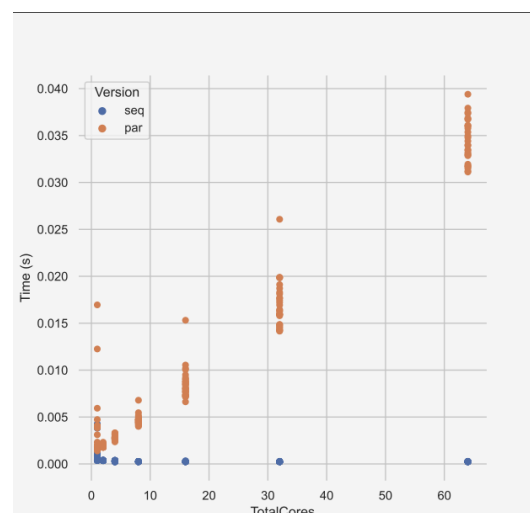


Figura 2 - Resultado com a sequência grande de DNA

