

**UNIVERSIDADE FEDERAL DE SÃO CARLOS**

Centro de Ciências Exatas e de Tecnologia

Departamento de Computação

**Programação Paralela e Distribuída**

**Exercício Programa - EP1**

**Professor:** Hermes Senger

**Aluno:** Ivan Duarte Calvo, 790739

São Carlos, 7 de Dezembro de 2022

## 1. Metodologia

Os códigos *pi\_seq.c*, *pi\_pth.c* e *pi\_omp.c* foram submetidos para funcionamento no *cluster*, e os resultados foram obtidos através da saída padrão presente no arquivo *.out*. Após a obtenção dos resultados foram construídas as tabelas presentes na próxima seção e também calculados os valores de Speedup e Eficiência através das seguintes equações, onde  $n$  representa o número de threads utilizadas:

$$Speedup(n) = \frac{t_{seq}}{t_n}$$

$$Eficiencia(n) = \frac{Speedup(n)}{n}$$

## 2. Resultados

### a. Sequencial

nº de threads	tempo de execução
1	3,401355

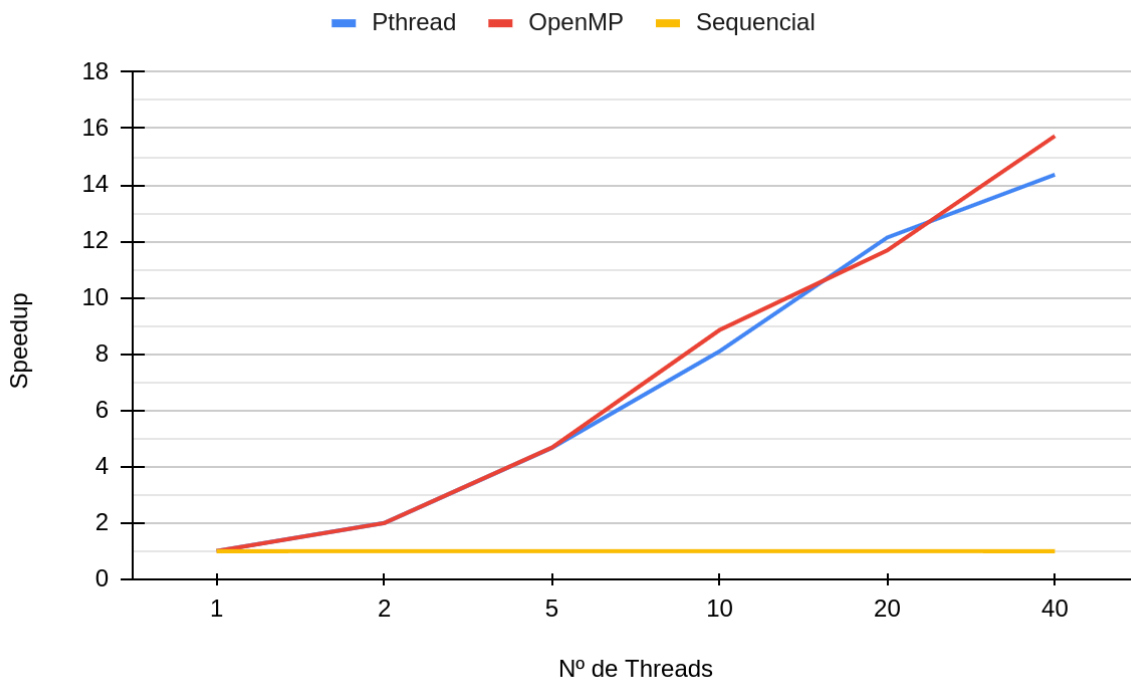
### b. Pthread

nº de threads	tempo de execução	Speedup	Eficiência
1	3,397986	1	1
2	1,697032	2,002311094	1,001155547
5	0,727662	4,669731276	0,9339462553
10	0,419893	8,092504519	0,8092504519
20	0,280027	12,13449417	0,6067247087
40	0,236643	14,35912324	0,3589780809

### c. OpenMP

nº de threads	tempo de execução	Speedup	Eficiência
1	3,393491	1	1
2	1,699927	1,996256898	0,998128449
5	0,725537	4,677212878	0,9354425756
10	0,383445	8,850007172	0,8850007172
20	0,290585	11,67813549	0,5839067743
40	0,215612	15,73887817	0,3934719543

### 3. Gráfico do speedup



### 4. Discussão dos resultados

Observando os resultados obtidos pode-se notar claramente a grande vantagem da programação paralela em relação à programação sequencial. Especialmente quando se utiliza todo o potencial de um nó do cluster, utilizando 40 threads, se obtém um ganho de performance de aproximadamente 15 vezes a performance com a programação sequencial.

É também interessante observar a pouca variação dos tempos de execução e ganhos de performance entre as implementações com OpenMP e Pthread. Ao menos para esse caso específico, é justificável concluir que a implementação com o OpenMP se mostra mais vantajosa, visto que não apresenta nenhum decréscimo de performance ao mesmo tempo em que auxilia com uma implementação consideravelmente menos complexa.