

**Universidade Federal de São Carlos**  
**Programa de Pós-Graduação em Ciência da Computação**

**CCO-129-7 Introdução à Computação de Alto  
Desempenho (2021/2)**

---

**Exercício Programa - EP 2**

**OpenHPC UFSCar**

---

**Alcides Mignoso e Silva      760479**

**Submission Date : 24/02/2022**

# 1 Atividade

O objetivo desta atividade é paralelizar a solução da equação de transferência de calor de Laplace pelo método de Jacobi utilizando **pthread**s. O código base utilizado foi o existente em <https://github.com/HPCSys-Lab/HPC-101/tree/main/examples/laplace> e este relatório visa discutir valores de tempo de execução, speedup e eficiência das execuções.

As execuções foram realizadas utilizando matrizes com número de linhas e colunas igual a 2\_000, e com 1, 2, 5, 10, 20 e 40 threads, em um computador com um processador Intel i7-10510U (8 threads) e 16gb de memória RAM.

## 2 Resultados

### 2.1 Tabelas

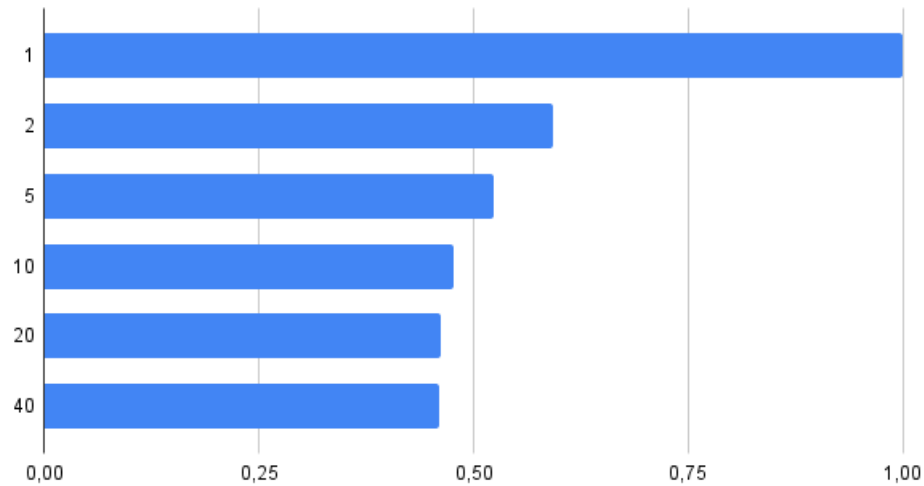
A tabela abaixo trás os tempos de execução do programa (em segundos) na versão sequencial e na paralelizada.

N. Threads	Sequencial	Pthread
1	86,7438	12,3731
2	86,7438	6,4107
5	86,7438	6,4100
10	86,7438	6,4153
20	86,7438	6,3785
40	86,7438	6,4053

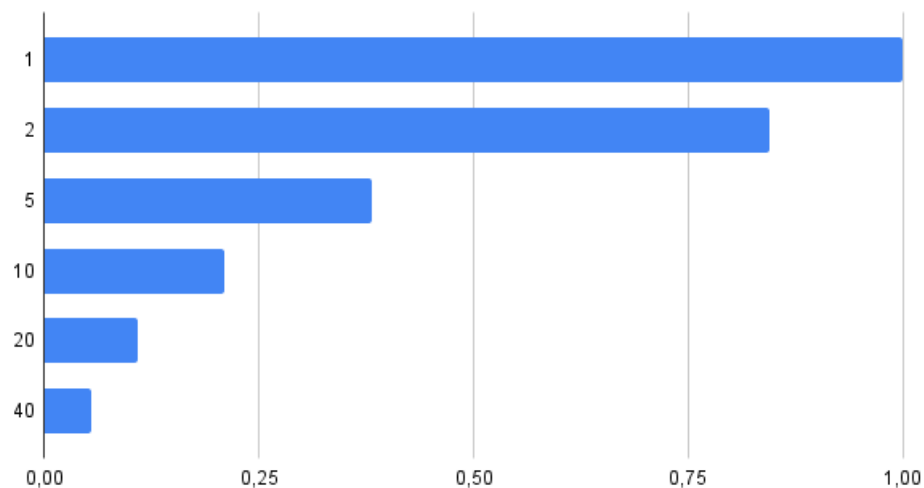
## 2.2 Gráficos

Os gráficos abaixo ilustram valores de speedup e eficiência do programa (horizontal) em relação ao número de threads utilizadas na execução (vertical).

Speedup - Laplace



Eficiência - Laplace



## 2.3 Discussão

Como pode-se perceber através do gráfico de speedup e da tabela de tempos de execução, a utilização de mais de 2 threads não impacta fortemente o tempo de execução do programa e consequentemente não altera bruscamente o speedup. Isso se dá por conta do overhead adicionado na criação de novas threads adicionado ao overhead de dividir e reduzir os resultados obtidos pelas novas threads.

Uma consequência desses overheads é a diminuição da eficiência quando novas threads são adicionadas, dado que os tempos de execução acabam não sendo reduzidos significativamente. Caso novas threads continuassem a serem adicionadas, é esperado que o tempo de execução comece a crescer por conta dos overheads já mencionados, ao invés de diminuir - paralelamente a isso a eficiência continuaria a cair.