

# Relatório EP1

MAC0219 – Programação Concorrente e Paralela

Thiago José Benitez Pena - N.USP 6847829

## Testes

O computador utilizado nos testes foi um AMD Ryzen 5 1600 Six-Core Processor, com 6 núcleos físicos, 12 núcleos lógicos, 16 GB de memória RAM e SSD.

Os cálculos de média e desvio padrão das medições foram calculada com a biblioteca Pandas da linguagem Python. Para o cálculo dos intervalos de confiança, calculados no Excel, adotamos  $\alpha = 0.01$ .

## Apresentação e Análise de Medições

### Regiões

A Figura 1 mostra o tempo de execução em relação às diferentes regiões com os demais parâmetros fixados: 1 *thread*, tamanho da entrada = 4096 e sem I/O e alocação de memória. Não houve diferença significativa entre os modelos OpenMP e PThreads com esses parâmetros.

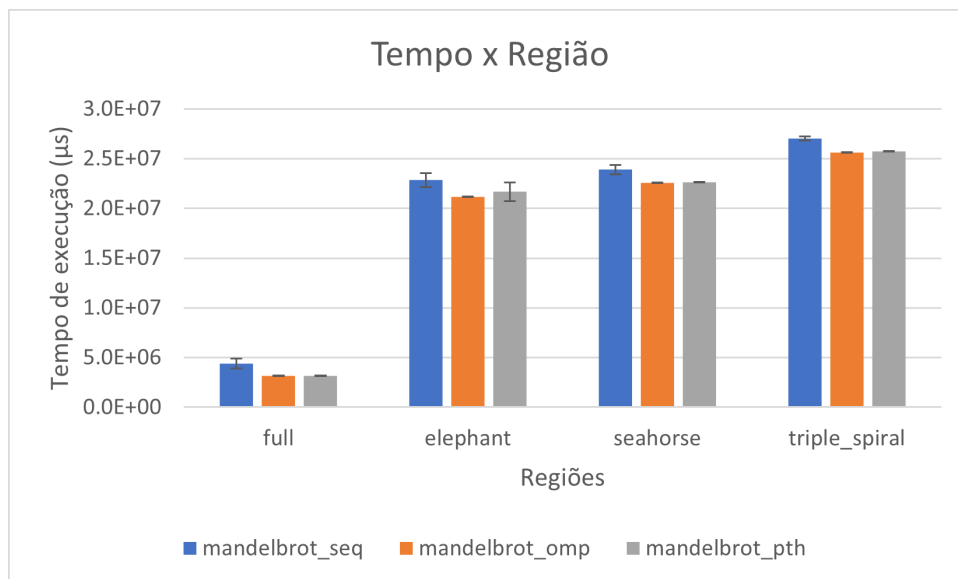


Figura. 1: Tempo x Região

Podemos notar que a região *full* é executada com tempo menor que as demais, exceto para *elephant*, onde não houve diferença significativa. Nota-se também que os tempos para o modelo sequencial foram ligeiramente maiores que os demais, mesmo considerando o intervalo de confiança.

## I/O e Alocação de Memória

A Figura 2 mostra o tempo de execução em relação a presença ou não de I/O e alocação de memória, com os demais parâmetros fixados: 1 *thread*, tamanho da entrada = 8192 e na região *full*.

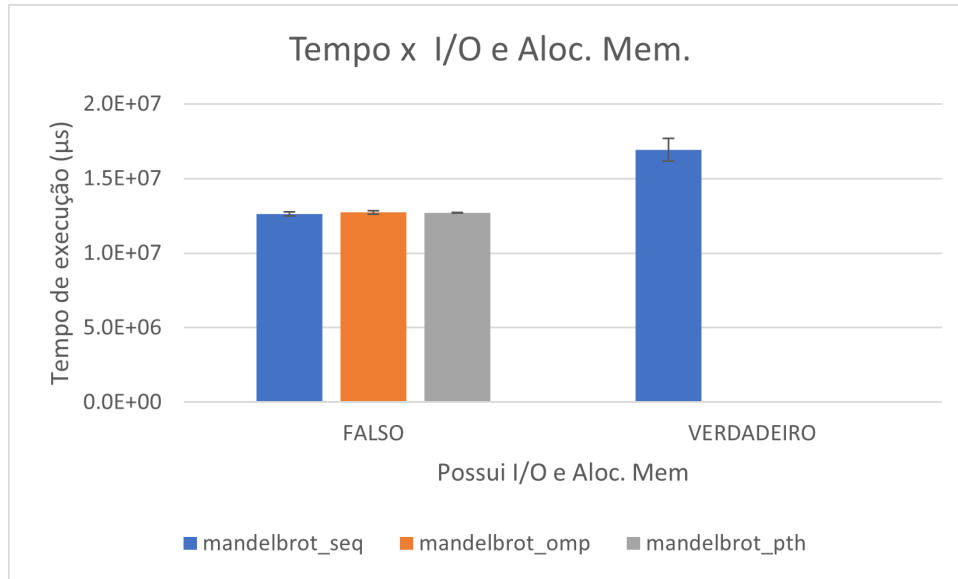


Figura. 2: Tempo x I/O e Alocação de Memória

Nota-se que o tempo é significativamente maior quando há I/O e alocação de memória, como era esperado.

## Número de Threads

A Figura 3 mostra o tempo de execução em relação a presença ou não de I/O e alocação de memória, com os demais parâmetros fixados: tamanho da entrada = 8192, na região *triple\_spiral* e sem I/O e alocação de memória.

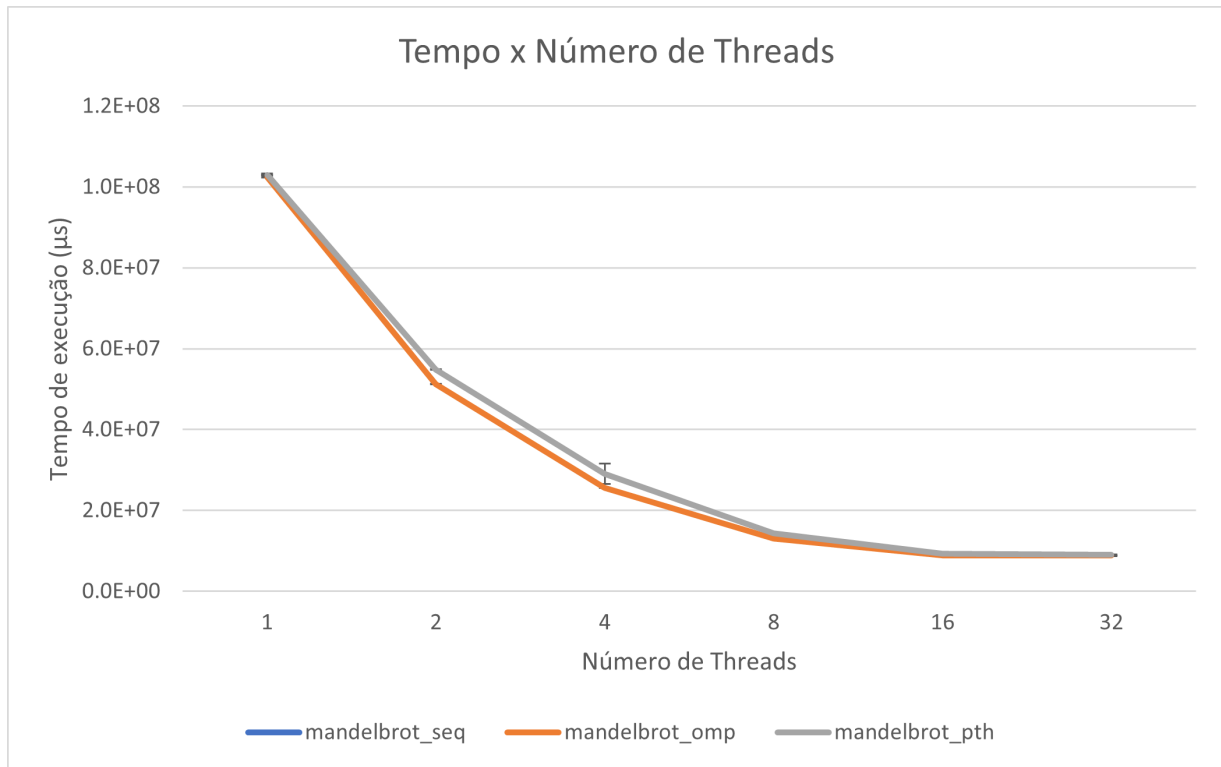


Figura. 3: Tempo x Número de Threads

O modelo sequencial só possui ponto com 1 *thread*, no qual o tempo é semelhante aos demais. Nota-se que o tempo de execução é reduzido drasticamente com o aumento de número de *threads*, estabilizando a partir de 16 *threads*. A performance do modelo OpenMP foi um pouco melhor em alguns casos, porém, na maior parte não há diferença estatisticamente significativa.

## Tamanho de Entrada

A Figura 4 mostra o tempo de execução em relação ao tamanho da entrada, com os demais parâmetros fixados: 1 *thread*, tamanho da entrada = 8192, na região *seahorse* e sem I/O e alocação de memória.

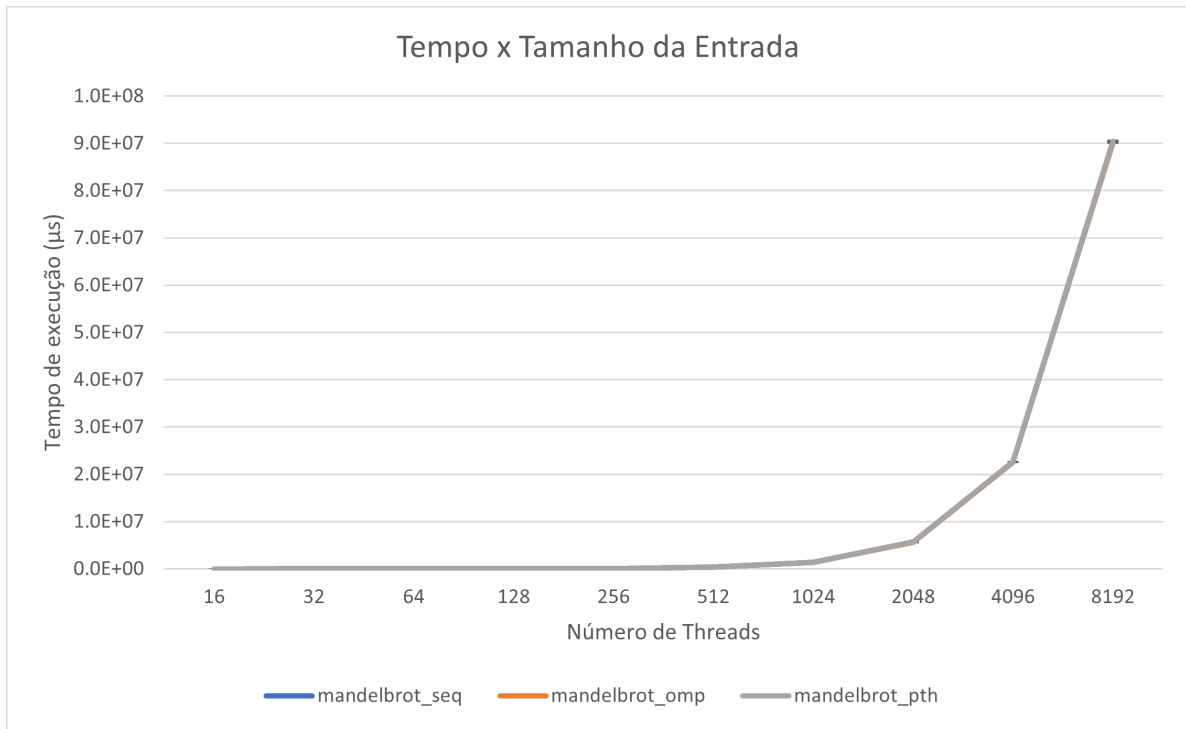


Figura. 4: Tempo x Tamanho da Entrada

Nota-se como o tempo cresce exponencialmente no gráfico, lembrando que o tamanho da entrada também cresce exponencialmente no eixo das abscissas. A performance entre os modelos com 1 *thread* ficou tão próxima que parece haver apenas uma curva no gráfico.

## Outras análises

Nesta seção, é feita a análise dos tempos de execução variando tanto o tamanho da entrada, quanto o número de *threads* para os modelos OpenMP, na Figura 5 e PThreads, Figura 6. Mantivemos fixa a região em *full*.

Essas duas figuras possuem um formato bastante semelhante e evidenciam como a crescente melhora de performance em relação a 1 *thread* com o aumento do tamanho de entrada. Nota-se que com o modelo PThreads, curiosamente não houve um grande aumento de performance ao passar de 2 para 4 *threads*

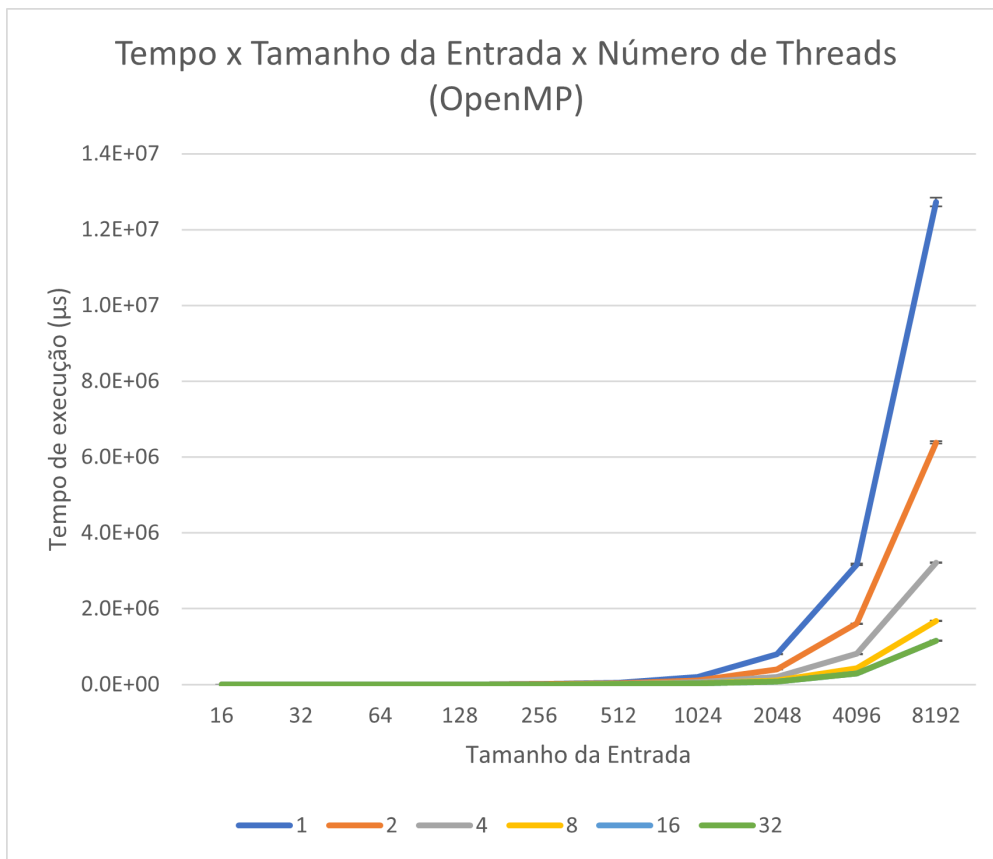


Figura. 5: Tempo x Tamanho da Entrada x Número de Threads (OpenMP)

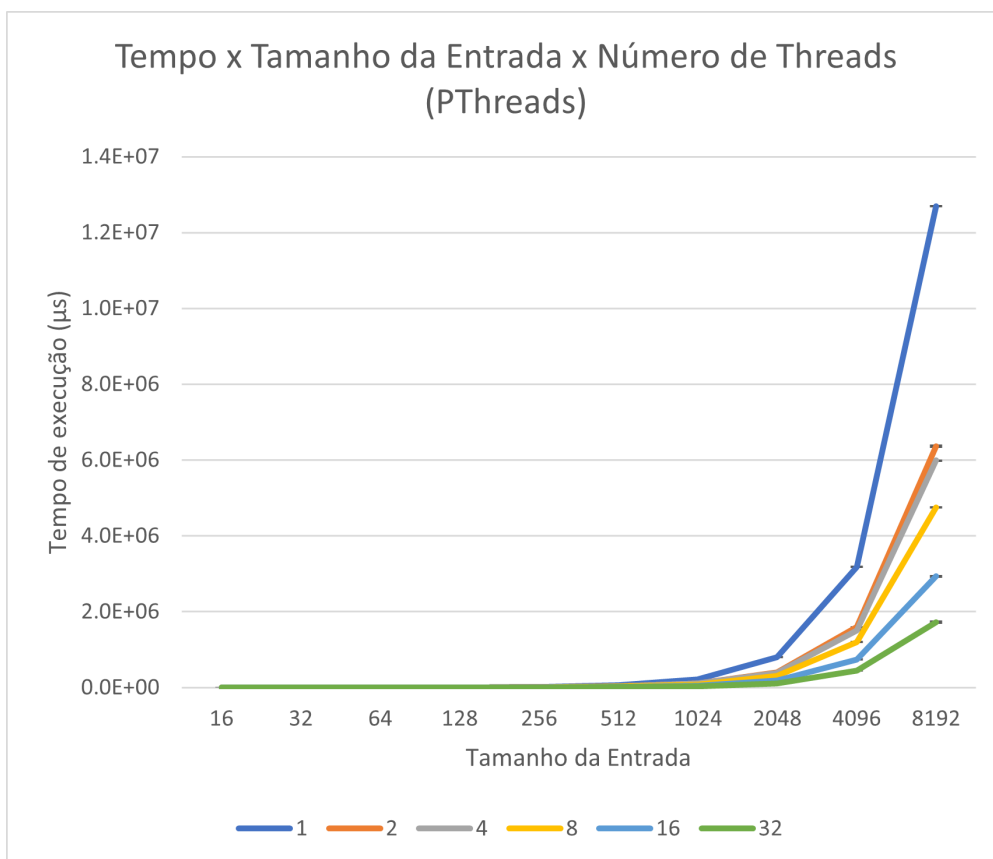


Figura. 6: Tempo x Tamanho da Entrada x Número de Threads (PThreads)