

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
INSTITUTO METRÓPOLE DIGITAL

Trabalho Prático da Unidade 1 - Processos e Threads

Breno Ferreira Leonez Leite

Natal - RN
2021

1. Introdução

Uma das principais características dos sistemas operacionais atuais é o paralelismo, ou seja, a capacidade de executar tarefas de maneira paralela dependendo da quantidade de núcleos que o processador utilizado possui. Dessa forma, através do uso de processos e threads, é possível resolver problemas mais rapidamente do que eles seriam caso fossem resolvidos de maneira sequencial. A fim de avaliar as vantagens e desvantagens entre as abordagens sequencial e paralela (utilizando processos e threads), foram implementados programas para realizar multiplicação de matrizes e comparar seus desempenhos.

2. Descrição do projeto

O [projeto](#) possui 4 programas base, cujas funcionalidades estão descritas a seguir:

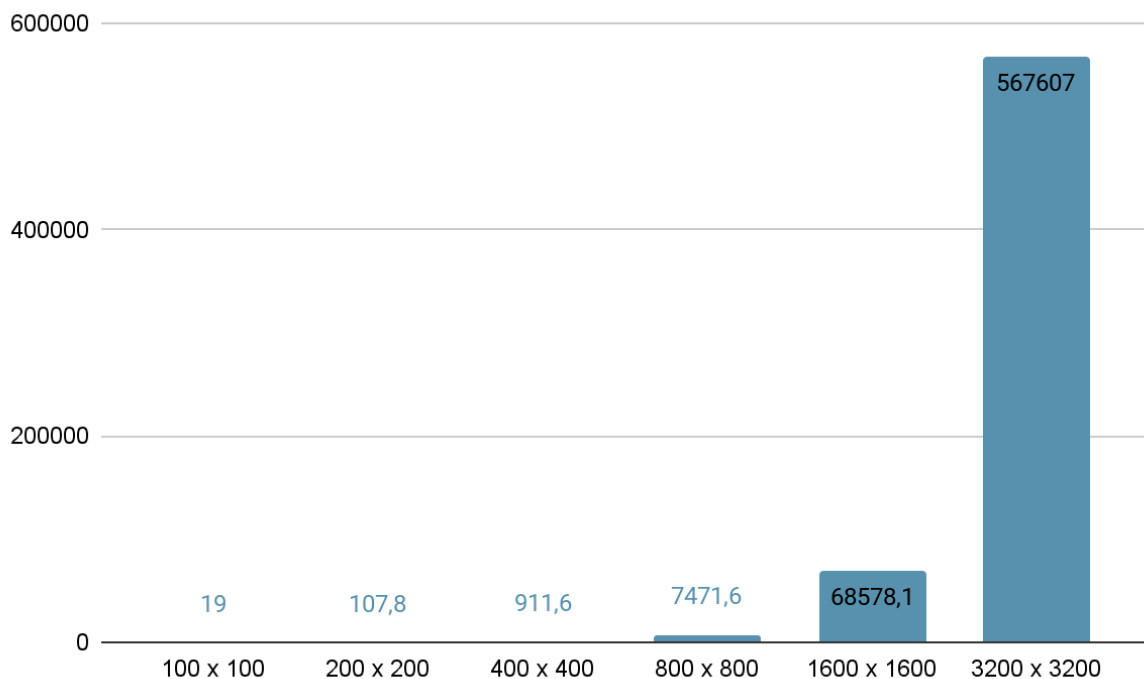
- `auxiliar.cpp`: Recebe 4 argumentos (n_1 , m_1 , n_2 , m_2) via linha de comando, representando dimensões de duas matrizes distintas. Os dois primeiros argumentos serão utilizados para gerar uma matriz aleatória A, de dimensões $n_1 \times m_1$, e os dois últimos argumentos comporão uma matriz aleatória B, de dimensões $n_2 \times m_2$, as quais serão armazenadas em arquivos de texto separadamente.
- `sequencial.cpp`: Recebe como entrada dois arquivos contendo duas matrizes A e B via linha de comando e realiza a multiplicação entre elas, salvando o resultado em um único arquivo de texto.
- `processos.cpp`: Recebe como entrada dois arquivos contendo duas matrizes A e B e um número inteiro P via linha de comando e implementa a multiplicação entre A e B, criando um novo processo para cada P elementos da matriz resultado. Cria $\lceil \frac{n_1 * m_2}{P} \rceil$ arquivos para salvar cada parte do resultado da operação.
- `threads.cpp`: Recebe como entrada dois arquivos contendo duas matrizes A e B e um número inteiro P via linha de comando e implementa a multiplicação entre A e B, criando uma nova thread para cada P elementos da matriz resultado. Cria $\lceil \frac{n_1 * m_2}{P} \rceil$ arquivos para salvar cada parte do resultado da operação.

3. Sequencial vs Paralelo

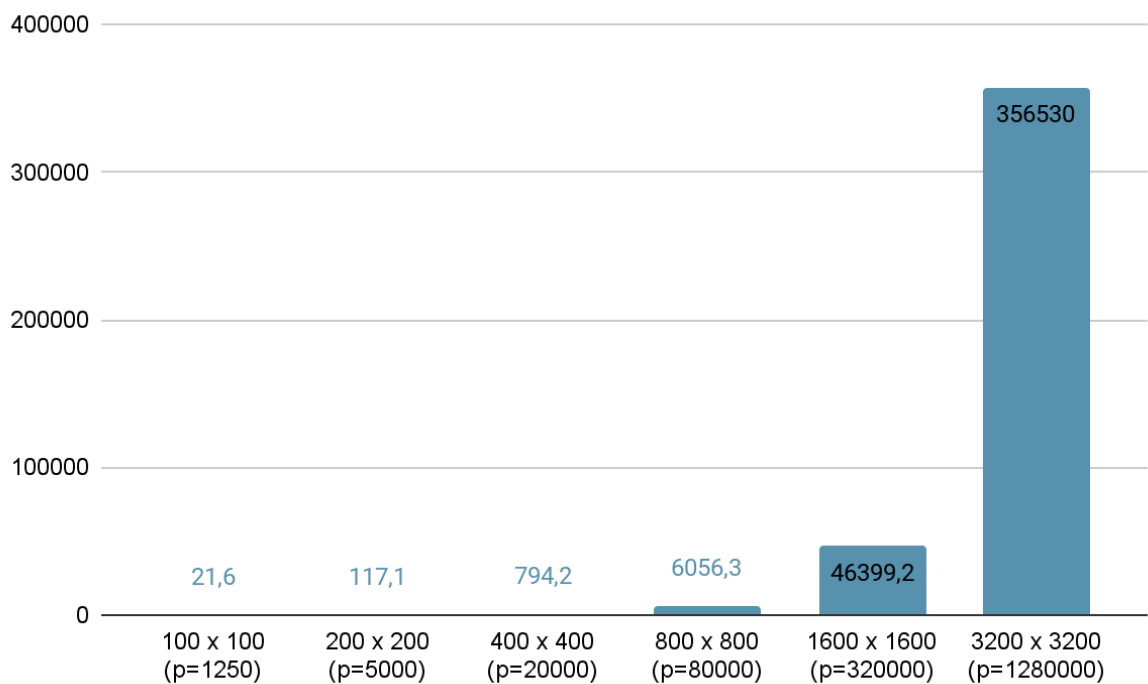
3.1. Experimento E1

Para cada uma das três implementações, o programa foi executado 10 vezes com diversos tamanhos de matrizes A e B. Em seguida, o tempo médio de execução foi calculado utilizando esses valores. Para o caso da implementação sequencial, o tempo considerado foi o localizado no final do arquivo resultado. Para as implementações paralelas, o tempo utilizado no cálculo foi o maior dentre todos os resultados intermediários.

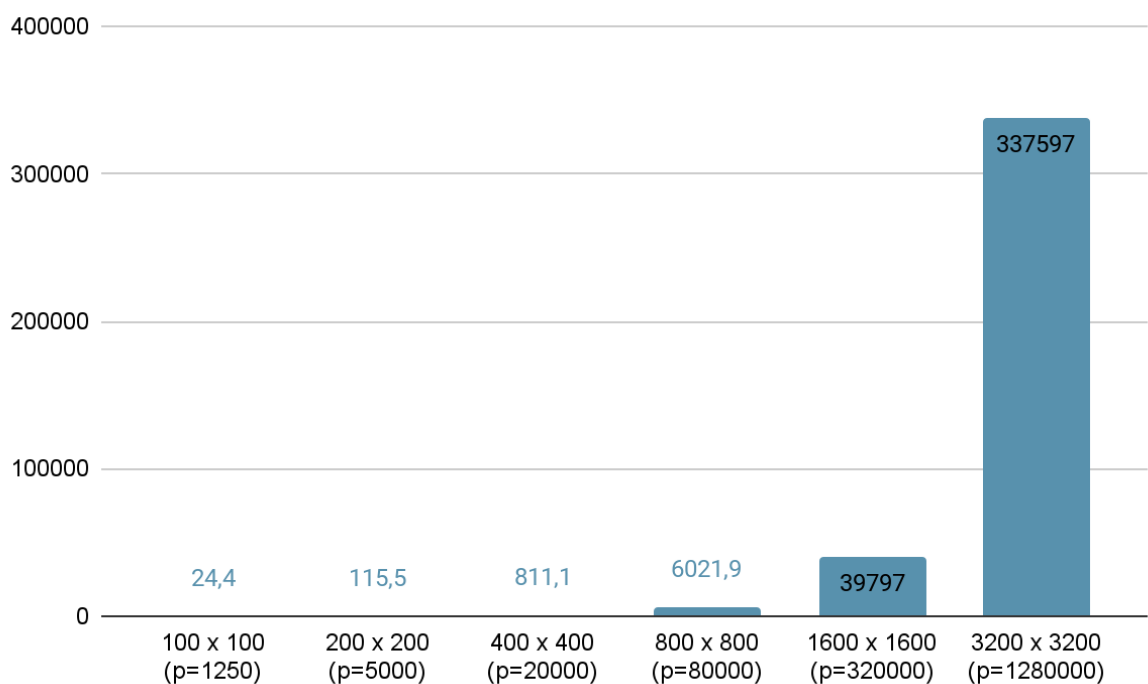
3.1.1. Sequencial



3.1.2. Processos



3.1.3. Threads

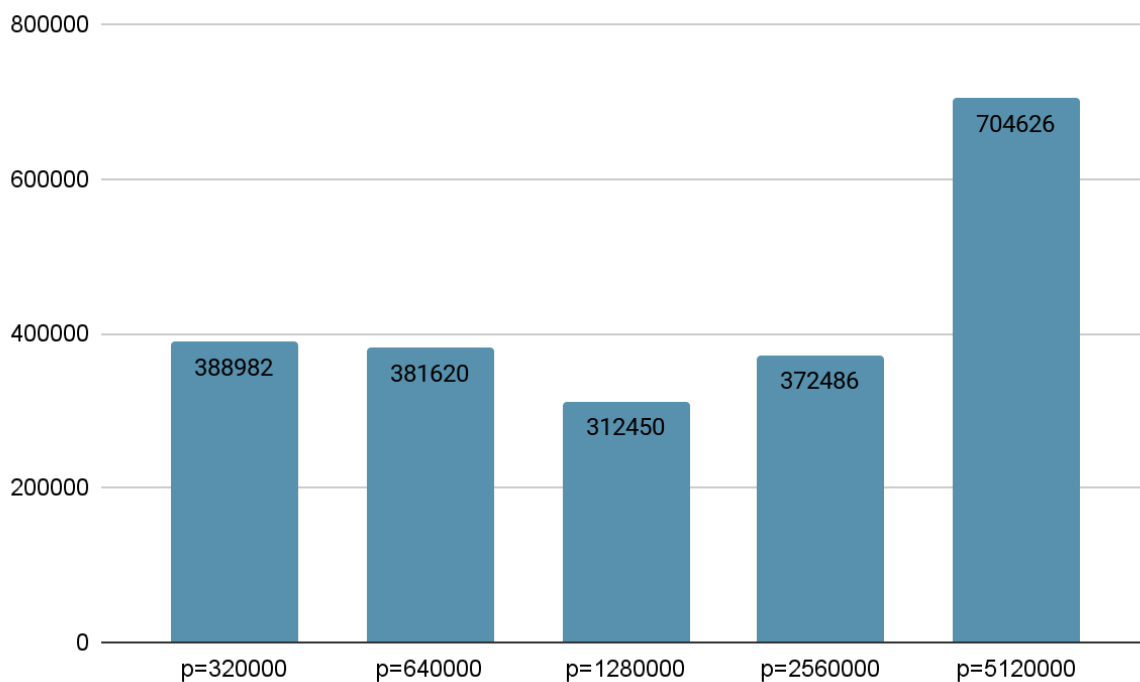


3.2. Experimento E2

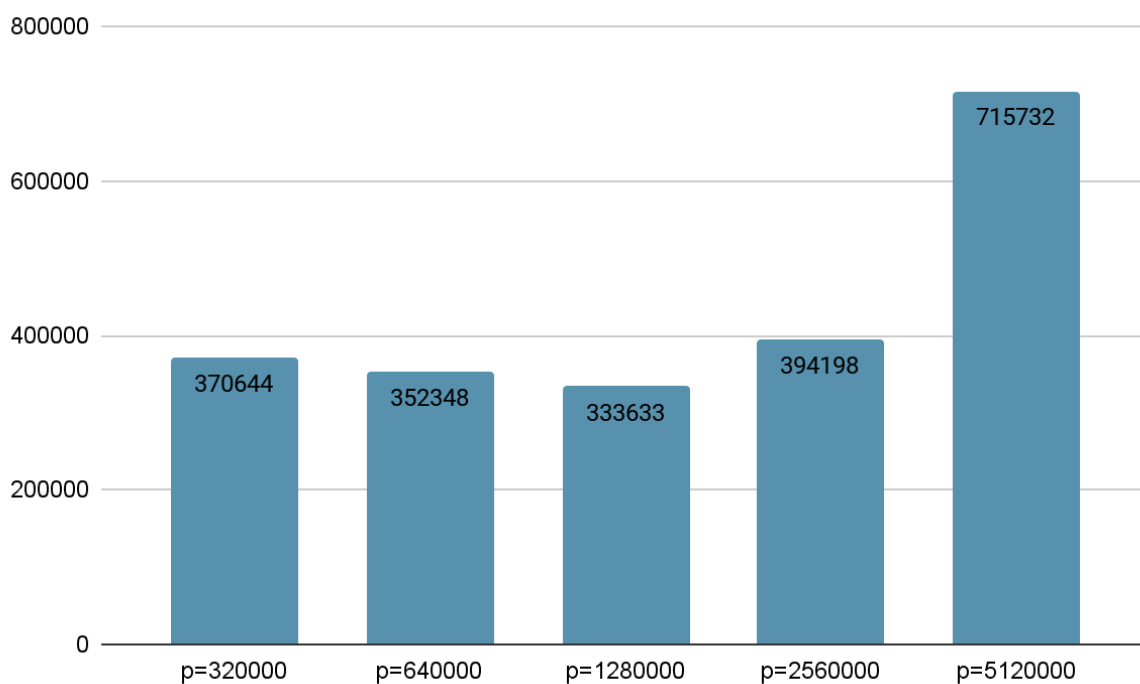
Utilizando as matrizes 3200 x 3200 encontradas no Experimento 1, os programas paralelos foram testados com valores de P iniciados em $\lceil \frac{n1*m2}{8} \rceil$ decrementado-o até que o valor chegue em $\frac{1}{4}$ desse valor. Em seguida, o valor de P foi aumentado até

alcançar $\lceil \frac{n1*m2}{2} \rceil$. Cada valor foi testado 10 vezes e realizada a média do tempo de execução para fins de comparação.

3.2.1. Processos



3.2.2. Threads



4. Discussões

- a) Qual o motivo dos resultados obtidos no experimento E1? O que pode ter causado o comportamento observado?

Com relação a diferença de tempo entre as implementações sequencial e paralela, a diferença de tempo ocorre devido à multiprogramação do SO, que permite que P elementos da matriz resultado sejam calculados de maneira paralela, diminuindo o custo de execução. Entre as soluções paralelas, as threads se demonstram mais rápidas pois consomem menos recursos e são mais fáceis de criar quando comparadas com os processos.

- b) Qual o motivo dos resultados obtidos no experimento E2? O que pode ter causado o comportamento observado?

Analisando os gráficos do experimento E2, é possível perceber que há uma queda de desempenho quando o valor de P aumenta de forma considerável. Isso ocorre pois há menos processos/threads criados calculando um número maior de elementos, fazendo com que o paralelismo não seja implementado de maneira eficiente.

- c) Qual é o valor de P ideal para a multiplicação das matrizes M1 e M2? Justifique sua resposta através dos experimentos realizados.

Através dos gráficos do experimento E2, é possível verificar que o valor ideal de P para as implementações paralelas é $\lceil \frac{n1*m2}{8} \rceil$, executando a multiplicação com menor tempo médio dentre os valores testados.