UNIVERSIDADE FEDERAL DE SÃO JOÃO DEL-REI
CIÊNCIA DA COMPUTAÇÃO

[Computação Paralela] Trabalho Prático 1

Lucas Rômulo de Souza Resende

Trabalho Prático 1 de Computação Paralela do curso de Ciência da Computação da Universidade Federal de São João del-Rei.

Prof. Dr. Rafael Sachetto Oliveira

São João del-Rei
19 de outubro de 2022

Sumário

1 Introdução

O objetivo desse trabalho é calcular quantos divisores um número possui, com a finalidade de reconhecer números primos. Numerais primos são números inteiros que possuem apenas dois divisores: 1 e ele próprio.

O programa implementado é baseado no paradigma de comutação paralela e utiliza a biblioteca OpenMPI da linguagem C.

Foram realizados testes em dois ambientes diferentes: o primeiro conta com a utilização de apenas um computador pessoal, enquanto o segundo utiliza vários computadores de um dos laboratórios da universidade.

Os resultados que se obtém, na comparação dos testes, mostram que a paralelização dessa tarefa é vantajosa.

2 Paralelização

A paralelização ocorre utilizando a biblioteca OpenMPI implementada na linguagem C, na fase de computação dos divisores dos valores de entrada.

A biblioteca provê os métodos baseados no paradigma, como Broadcasting, Scatter/-Gather, Send/Reciece, que são utilizados no programa, entre outros.

Os métodos utilizados na implementação do programa são os explicados abaixo:

- Broadcasting: é utilizado para transmitir dados de uma origem para todos os processos que utilizam o mesmo comunicador
- Scatter/Gather: do inglês espalhar/recolher, são métodos que dividem e reunem dados, respectivamente, a partir de um processo mestre utilizando
- Send/Recieve: do inglês enviar/receber, esses métodos são utilizados na troca de dados entre dois processos

3 Descrição do programa

O fluxo do programa ocorre da seguinte forma:

1. O processo mestre carrega o arquivo de entrada em um buffer específico e define N (a quantidade de valores que cada processo vai computar)
2. N é transmitido para todos os processos utilizando o método de Broadcasting
3. O mestre divide os valores para os processos utilizando o método de Scatter
4. Cada processo realiza a computação necessária para os valores recebidos e cronometra o tempo gasto
5. O mestre reúne os resultados no mesmo buffer de entrada utilizando do método Gather e os armazena em um arquivo de saída
6. Os processos comunicam ao mestre o tempo de computação gasto utilizando o método Send
7. O processo mestre recebe os tempos com o método Recieve e os armazena em arquivo

4 Testes

4.1 Arquivos de entrada/saída

O arquivo de entrada utilizado na realização dos testes foi provido pelo próprio professor. Esse arquivo conta com 1440 valores inteiros.

O arquivo de saída armazena a quantidade de divisores que cada valor de entrada possui, com excussão do 1 e dele próprio. Seguindo essa ideia, os números primos podem ser identificados nas saídas com valor 0.

4.2 Computador pessoal

O computador utilizado possui um processador com 8 núcleos, e todos foram utilizados no teste.

Cada processo teve de computar 180 valores, e foram armazenados o tempo total do programa (sem I/O) e o tempo gasto por cada processo na fase da computação.

O gráfico abaixo representa o tempo utilizado de máquina por cada processo:

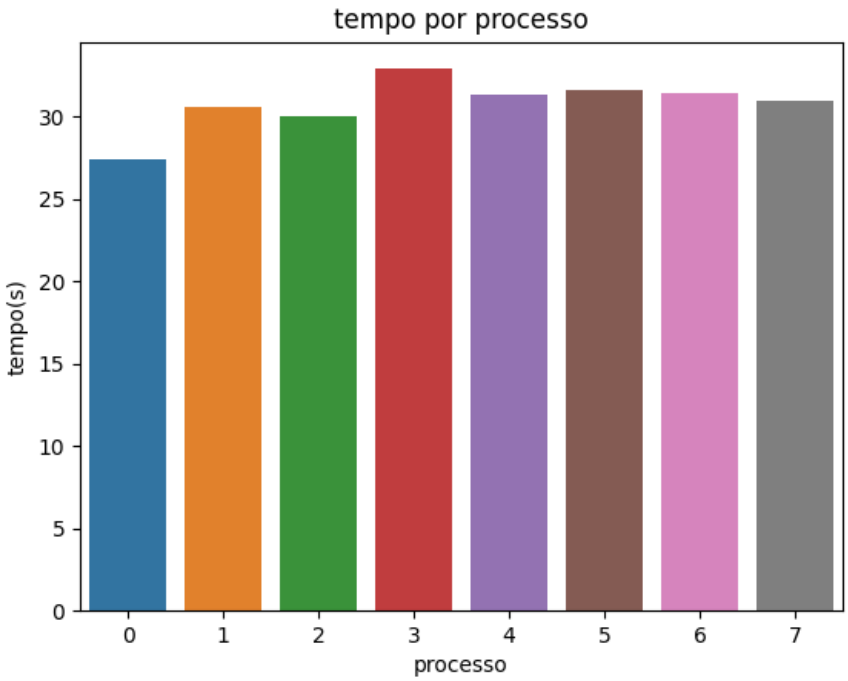

Figura 1: Tempo de computação por processo em computador pessoal

4.3 Laboratório

No teste no laboratório da universidade foram utilizadas 5 máquinas, cada uma com 6 núcleos a disposição, o que resulta em um total de 30 núcleos.

Cada processo teve de computar 80 valores e, assim como no teste no computador pessoal, foram armazenados o tempo total do programa (sem I/O) e o tempo gasto por cada processo na fase da computação.

O gráfico abaixo representa o tempo utilizado de máquina por cada processo:

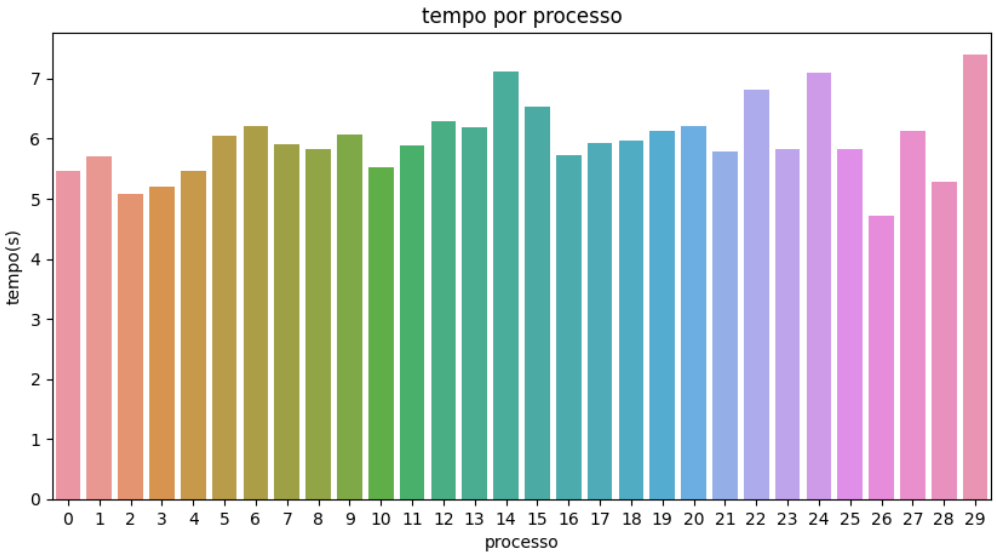

Figura 2: Tempo de computação por processo em laboratório

4.4 Tempo total

O gráfico abaixo representa o tempo total gasto pelo programa, com exceção do tempo de I/O. Esse tempo é medido pelo processo mestre, sendo que a contagem se inicia após a leitura do arquivo de entrada até a reunião dos resultados dos processos (Gather), o que ocorre antes da escrita do arquivo de saída.

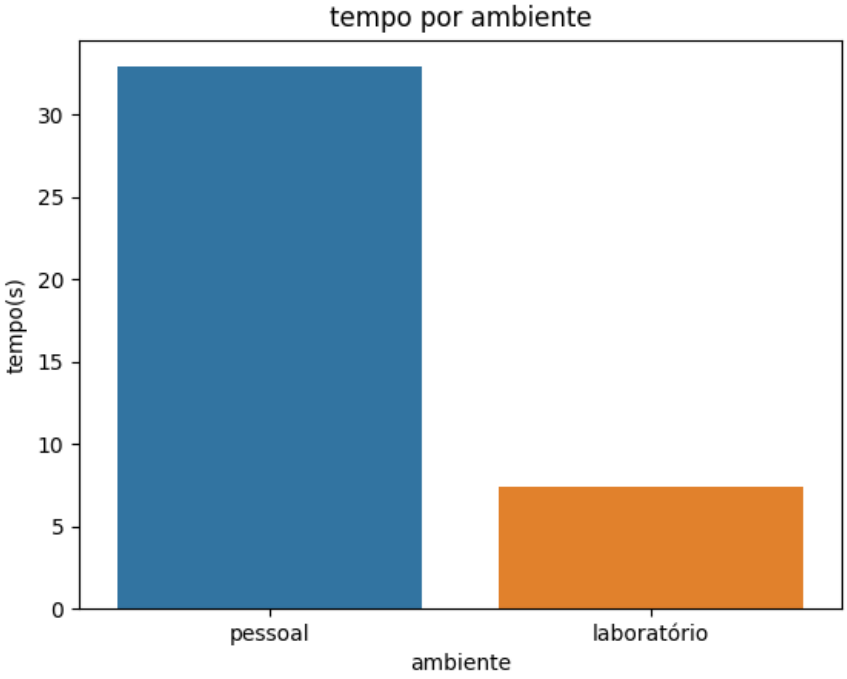

Figura 3: Tempo total de computação por ambiente

5 Conclusão

Pode-se observar que o tempo de execução com 8 processos (computador pessoal) foi maior que o tempo de execução com 30 processos (laboratório).

Se tratando de tempo total, o tempo gasto no computador pessoal foi maior que 30 segundos, enquanto o tempo gasto em laboratório foi menor que 10 segundos, o que dá mais de 20 segundos de diferença.

Em relação ao tempo por processo, o tempo mínimo gasto no computador pessoal foi o do processo 0, que ficou entre 25-30 segundos, enquanto no laboratório o tempo máximo gasto foi de pouco mais de 7 segundos.

Fica claro que, nesse problema, o aumento da quantidade de processos trás um ganho significativo, se pensando no tempo como única métrica.

Um outro ponto a ser observado é a questão de tempo gasto entre os processos.

No computador pessoal o processo que gasta menos tempo leva aproximadamente 27 segundos para realizar sua tarefa, enquanto o que processo que leva mais tempo demora quase 33 segundos, quase 6 segundos de diferença.

Isso pode ser observado também no teste em laboratório. O processo mais rápido levou aproximadamente 4.7 segundos, enquanto o mais demorado levou 7.4 segundos, uma diferença de 2.7 segundos.

Essas diferenças no tempo indicam que os processos que terminam primeiro acabam ficando parados enquanto outros estão computando, o que é uma perda de poder computacional.

Uma ideia para um trabalho futuro é contornar essa perda utilizando uma melhor distribuição de carga.