

# Trabalho prático 1 - Identificação de números primos

Rafael Sachetto Oliveira

29 de setembro de 2021

## 1 Objetivo

O objetivo deste trabalho é a implementação paralela mestre-escravo usando MPI de um algoritmo que identifica números primos em vetor de  $N$  inteiros.

Um vetor exemplo de entrada (entrada.txt) está disponível no portal. O arquivo de entrada deverá ser lido e carregado para memória no processo mestre. Após o processamento, cada processo escravo deverá retornar para o processo mestre quantos divisores exatos possui cada valor armazenado na sua fatia do vetor. Depois que o último escravo terminar, o processo mestre deverá colocar o número de divisores de cada elemento do arquivo de entrada em um arquivo de saída (saída.txt) **na ordem em que os valores originais estavam no arquivo de entrada**. A tomada de tempo das execuções será feita somente no processo mestre após leitura do arquivo de entrada e antes da escrita do arquivo de saída, ou seja, não iremos considerar o tempo de entrada/saída.

## 2 Etapas

A paralelização da aplicação pode ser dividida em 4 partes, as quais devem gerar um relatório técnico. São elas:

1. perfil de desempenho sequencial (usando prof)
2. identificação das oportunidades de paralelização
3. paralelização
4. avaliação dos ganhos da paralelização

## **2.1 Perfil de desempenho sequencial**

O ponto de partida desta tarefa é a execução da aplicação para carga de trabalho proposta para a mesma. Para cada execução deve-se analisar o seu tempo de execução, o tempo de execução em cada uma das suas funções, assim como a variação em consequência da variação de parâmetros e da carga. Mais ainda, deve-se avaliar o uso de recursos computacionais por parte da aplicação, utilizando aplicativos como vmstat. Um dos objetivos da determinação do perfil de desempenho é determinar as porções de código que teriam maior impacto se paralelizadas, por serem as mais demandantes computacionalmente.

## **2.2 Identificação das oportunidades de paralelização**

A identificação das oportunidades de paralelização deve prover as seguintes informações sobre cada uma das oportunidades identificadas:

- qual é a oportunidade, ou seja, qual o arquivo, função e linhas de código que compõem a oportunidade?
- qual a sua significância no perfil sequencial, ou seja, quanto do tempo de execução da aplicação está associado à oportunidade?
- qual a estratégia de paralelização com a sua justificativa (por que você acredita que essa estratégia é escalável e eficiente) e expectativa de eficiência, em termos da fração serial e paralela resultante. A apresentação de pseudo códigos é bem vinda.

## **2.3 Paralelização**

O relatório de paralelização deve documentar todas as atividades de paralelização realizadas no código da aplicação em termos tanto da estratégia utilizada como do código modificado. É uma documentação tradicional de trabalho prático, o que também inclui testes de execução.

## **2.4 Entrega**

Os trabalhos devem ser entregues impreterivelmente até a data marcada no portal didático.