

Trabalho prático 1 - Em Grupo (máximo 3 pessoas)

1 Objetivo

O objetivo deste trabalho é a implementação paralela, usando o modelo mestre-escravo com MPI, do algoritmo **Crivo de Eratóstenes** para identificar todos os números primos até um inteiro N .

A aplicação deverá receber como entrada um único número inteiro N (lido da linha de comando). O processo mestre será responsável por dividir o trabalho entre os processos escravos e consolidar os resultados.

2 Descrição do Algoritmo

O Crivo de Eratóstenes funciona da seguinte maneira:

1. Crie uma lista de inteiros consecutivos de 2 até N .
2. Comece com o primeiro primo, $p = 2$.
3. Marque todos os múltiplos de p ($2p, 3p, 4p, \dots$) na lista.
4. Encontre o próximo número na lista que não foi marcado. Este é o próximo número primo. Repita o passo 3 com este novo primo.
5. O processo se repete até que $p^2 > N$.

Todos os números não marcados na lista são primos.

3 Implementação Paralela

O processo mestre (rank 0) deverá ler o inteiro N e determinar a estratégia de divisão de tarefas. Uma abordagem comum é dividir o intervalo de números $[2, N]$ em blocos de tamanho igual e distribuir um bloco para cada processo escravo.

Cada processo escravo será responsável por executar o algoritmo do crivo em sua fatia do vetor. Um desafio importante é que cada escravo precisa da lista de primos até \sqrt{N} para poder eliminar os múltiplos em sua própria fatia. O mestre pode calcular esta lista inicial e enviá-la (Broadcast) para todos os escravos.

Após o processamento, cada escravo deverá retornar sua lista de números primos encontrados para o processo mestre. O mestre, por sua vez, deverá juntar os resultados de todos os escravos e escrever a lista final de primos, em ordem crescente, em um arquivo de saída chamado `primos.txt`.

A tomada de tempo será feita no processo mestre, englobando todo o processo de paralelização, desde a distribuição dos dados até o recebimento do último resultado (sem incluir a escrita no arquivo de saída).

4 Etapas e Relatório Técnico

A implementação e documentação do trabalho devem seguir 4 partes, que devem compor um relatório técnico:

1. **Perfil de desempenho sequencial:** Análise da versão sequencial do Crivo de Eratóstenes usando ferramentas como `gprof`.
2. **Identificação das oportunidades de paralelização:** Descreva os trechos de código com maior custo computacional e como eles podem ser paralelizados.
3. **Paralelização:** Documente a estratégia de paralelização (divisão de dados, comunicação entre processos) e as modificações no código.
4. **Avaliação dos ganhos da paralelização:** Apresente uma análise de desempenho da versão paralela, incluindo métricas como speedup e eficiência, variando o número de processos e o valor de N .

5 Entrega

Os trabalhos devem ser entregues até a data marcada no portal didático. Devem ser entregues um documento PDF contendo o relatório técnico e um arquivo compactado com o código-fonte desenvolvido em C, compilável com `mpicc`.