

# Relatório Final - Crivo de Eratóstenes

**Alunos:** Moab Esdras Leandro Barbosa e Thiago Carlos Silva Pereira

## 1. O Problema

O Crivo de Eratóstenes é um algoritmo e método simples e prático para encontrar números primos até um certo valor limite. Foi criado pelo matemático grego Eratóstenes (a.c. 285-194 a.C.). O algoritmo recebe como entrada uma lista de números inteiros de (2 até  $n$ ) e realiza uma marcação em todos os números não-primos da lista, ao final da execução, todos os números não marcados são primos.

## 2. Algoritmos

### 2.1.

---

#### Algoritmo serial - Crivo de Eratóstenes

---

1. Cria lista de números naturais não-marcados  $2, 3, \dots, n$
  2.  $k \leftarrow 2$
  3. Repete
    - (a) Marca todos os múltiplos de  $k$  entre  $k^2$  e  $n$
    - (b)  $k \leftarrow$  menor número não-marcado  $> k$Até  $k^2 > n$
  4. Os números não-marcados são primos
- 

### 2.2.

---

#### Algoritmo Paralelo com OPENMP - Crivo de Eratóstenes

---

1. Cria lista de números naturais não-marcados  $2, 3, \dots, n$
2.  $k \leftarrow 2$
3. Repetição paralela com `schedule(dynamic)`
  - (a) Marca todos os múltiplos de  $k$  entre  $k^2$  e  $n$
  - (b)  $k \leftarrow$  menor número não-marcado  $> k$Até  $k^2 > n$
4. Os números não-marcados são primos

---

## 2.3.

---

### Algoritmo Paralelo com OPENMPI - Crivo de Eratóstenes

---

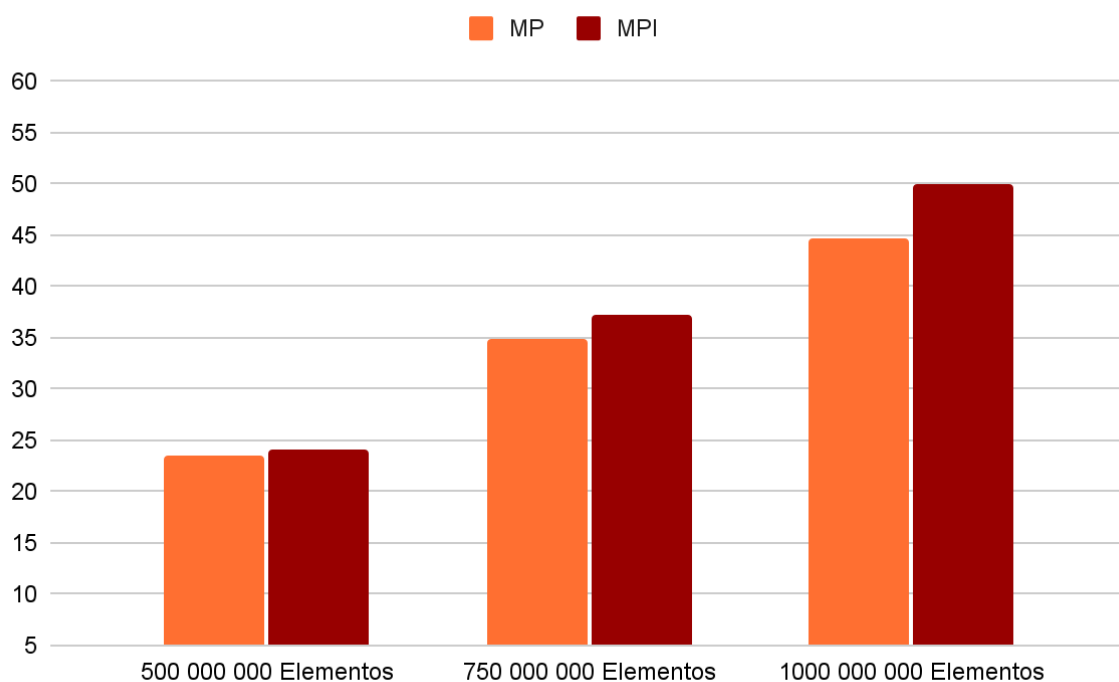
1. Cria lista de números naturais não-marcados  $2, 3, \dots, n$
  2.  $k \leftarrow 2$
  3. rank 0:
    - (a) Marca todos os primos entre  $k$  e  $\text{raiz}(n)$`MPI_Bcast` no intervalo
  4. `MPI_Scatter` na lista
  5. rank 0:  
`MPI_Gather`
- 

## 3. Análise de Desempenho

### 3.1. Resultados

Algoritmo	Quantidade de elementos	Média do tempo (segundos)	Speedup médio	Visão geral
MP	500 000 000	23.553125	1.004445	Esperado
MP	750 000 000	34.940625	0.971201	Inesperado
MP	1000 000 000	44.757812	1.014270	Esperado
MPI	500 000 000	24.191727	0,973602	Inesperado
MPI	750 000 000	37.323741	0,936150	Inesperado
MPI	1000 000 000	50.031897	8,945855	Inesperado

### 3.2. Gráficos



#### 4. Hardware/Software

Hardware/Software	Modelo
CPU	Intel Core i3-7020U CPU 2.30GHz
RAM	16,0 GB
SO	WSL-Ubuntu-20.04-(64 bits)
Linguagem de Programação	C
Compilador	gcc
MPI	MPI 4.1.1

#### 5. Implementação

O código completo pode ser encontrado aqui [Link](#)

#### 6. Dificuldades encontradas

Algumas dificuldades foram encontradas no desenvolvimento do trabalho e a principal delas diz respeito ao uso do WSL, pois houve uma notável perda de desempenho visto que o

algoritmo serial e MP não eram executados em condições semelhantes ao MPI, além disso, o fato de não possuímos um cluster para executar os testes MPI, também afetou a performance do algoritmo. Em alguns testes o speedup do MPI estava abaixo de 1, foram tomadas algumas medidas para contornar o problema, como, executar o serial pelo WSL, que melhorou o speedup geral, mas ainda suspeitamos que se executados em ambiente nativos os algoritmos teriam uma melhor performance.

## **7. Referências**

José Geraldo de Sousa Junior e Lourival Pereira Vieira Neto, Programação Concorrente e Paralela: Algoritmo paralelo para o Crivo de Eratóstenes - [primos.pdf \(puc-rio.br\)](#), 17 de novembro de 2009.

Michael J. Quinn. Parallel Programming in C with MPI and OpenMP, Oregon state university, Higher Education, 2003.