**Crivo de Eratóstenes**

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

**Github:** [Link](https://github.com/thcls/Crivo-de-Eratostenes.git)

**Recursos Computacionais(Hardware)**

Intel Core i3-7020U CPU 2.30GHz

16,0 GB - RAM

**Recursos Computacionais(Software)**

Linguagem Utilizada - C

WSL-Ubuntu-20.04-(64 bits)

**1. Descrição do Algoritmo Serial**

O crivo de Eratóstenes foi criado pelo matemático grego Eratóstenes (a.c. 285-194 a.C.), ele é um método simples e prático para determinar os números primos em uma lista de 2 até n elementos. O algoritmo marca todos os números não primos, e ao final os números que não foram marcados são primos, segue abaixo pseudo-código.

**Algoritmo serial - Crivo de Eratóstenes**

1. Cria lista de números naturais não-marcados 2,3,...,n

2. k ← 2

3. Repete

(a) Marca todos os múltiplos de k entre k^2 e n

(b) k ← menor número não-marcado > k

Até k^2 > n

4. Os números não-marcados são primos

O crivo é um método simples porém à medida que n aumenta o algoritmo irá demorar mais tempo para cumprir sua função.

**2. Descrição do Algoritmo Paralelo**

Paralelizar o algoritmo do **crivo de Eratóstenes** é particionar o trabalho do algoritmo serial entre os processos de modo que cada processo seja responsável por marcar os não-primos de sua partição. Assim, teremos como resultado final do algoritmo, a junção da parte dos processos. Se não for divisível por p, cada processo fica com n/p números, se n não for divisível por p, cada processo fica ou com chão(n/p) ou teto(n/p), para que a distribuição não fique desigual.

Nesse algoritmo, o primeiro processo deve possuir pelo menos os primeiros √n números, pois todos os números não-primos são múltiplos de fatores primos menores que √n. Depois que descoberto, o próximo primo é enviado para os outros processos através de um **broadcast,** utilizando a chamada **MPI\_Bcast**. Todos os processos marcam os múltiplos deste primo recebido pelo broadcast feito pelo primeiro processo, e, quando este fator primo for maior que √n, o processo conta quantos primos encontrou e faz uma **MPI\_Reduction** para que o processo 0 totalize os primos.

**Algoritmo paralelo - Crivo de Eratóstenes**

1. Cria lista de números naturais não-marcados 2,3,...,n

2. k ← 2

3. Repete

(a) Marca todos os múltiplos de k entre k^2 e n

(b) k ← menor número não-marcado > k

(c) O processo 0 difunde k para os outros processos

Até k^2 > n

4. Os números não-marcados são primos

5. Reduz para determinar o número de primos

**3. Descrição da análise de desempenho**

A análise será feita utilizando duas máquinas com as especificações apresentadas anteriormente, para valores de n = 500 000000, n = 750 000 000, e n = 1000 000 000. Para cada valor de n serão feitos 100 testes de modo a obter um tempo mais preciso. O tempo de execução de cada teste será obtido através da diretiva **omp\_get\_wtime()** e a média será calculada. Por fim, será construído um gráfico de barras que relaciona n com o tempo de execução em cada algoritmo.