

Exercício de fixação - GABARITO

**Crivo de Eratosthenes**

A Peneira de Eratosthenes é um algoritmo clássico para encontrar os números primos  $\leq N$  (um número inteiro positivo). O pseudo código sequencial abaixo ilustra o funcionamento da Peneira de Eratosthenes:

1. Crie um vetor de números inteiros  $\text{vet} = \{2, 3, 4, \dots, N\}$ , sendo que nenhum item de  $\text{vet}$  é marcado como primo (i.e., são iniciados com zero);
2. Determine  $K = 2$ , sendo  $K$  o primeiro número não marcado em  $\text{vet}$ ;
3. Repita
  - (a) marque todos os múltiplos de  $K$  entre  $K^2$  e  $N$ ;
  - (b) encontre em  $\text{vet}$  o menor número maior que  $K$  que não está marcado e atribua a  $K$  este novo valor;até  $K^2 > N$
4. Os números não marcados em  $\text{vet}$  são todos números primos.

Exemplo do algoritmo Peneira de Eratosthenes para  $N = 60$ .

a) Criar a lista/o vetor  $\text{vet}$  de números naturais. Nenhum está marcado.

|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| XX | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |

(b)  $K = 2$  (marcar nrs: 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, ... , 60)

|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| XX | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |

(c)  $K = 3$  (marcar nrs: 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57 e 60)

|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| XX | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |

(d)  $K = 5$  (marcar nrs: 25, 30, 35, 40, 45, 50, 55 e 60)

|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| XX | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |

(e)  $K = 7$  (marcar nrs: 49 e 56)

|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| XX | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |

(f) como  $K = 11$ , onde  $K^2 > N$ , então sai do **repita...até**.

Os números primos são os não marcados em vet: 02, 03, 05, 07, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53 e 59.

Faça um projeto de algoritmo paralelo seguindo a metodologia PCAM para o problema da peneira de eratosthenes. Descreva, explicitamente, de forma textual e, se necessário, gráfica, o particionamento, a comunicação, a aglomeração e o mapeamento.

Considere que este algoritmo paralelo deve executar o mais rápido possível, que ele será carregado para execução em um cluster de computadores e que o seu projeto deve usar o particionamento por dados.

Por último, considere que forneceremos este projeto que vocês estão fazendo agora, para outro grupo implementá-lo remotamente. Espera-se que o seu projeto seja suficientemente detalhado para que a outra equipe possa fazer a implementação adequadamente. Fica a dica!

\*\*\*\*\*

Resposta:

Particionamento (versão de particionamento por dados):

O foco da paralelização estará no passo 3a, i.e., na marcação de todos os múltiplos de K entre  $K^2$  e N. Para a versão paralela com particionamento por dados, vet é particionado em N tarefas, sendo que cada tarefa marca vet, caso a divisão de N por K resulte em um resto 0. Isso precisa ser feito a cada iteração, sendo que na primeira delas o valor de  $K == 2$ .

A partir da segunda iteração um novo valor de K precisa ser descoberto (passo 3b), por uma operação de redução, de ordem logarítmica. Como detalhado na seção de comunicação, o novo valor de K precisa ser repassado a todas as tarefas novamente, para que o passo 3a se repita.

Comunicação (versão de particionamento por dados):

Há duas comunicações globais no particionamento por dados acima, a cada iteração. Há uma operação de redução para determinar o novo valor de K e um broadcast para fornecer este novo valor para todas as tarefas ainda em atividade na aplicação, que possuam um valor para o valor de vet maior que  $K^2$ .

Estas comunicações são caras, se comparadas ao custo da possível marcação que cada tarefa deve fazer iterativamente.

Aglomeração (versão de particionamento por dados):

Dada a plataforma alvo deste algoritmo (um cluster de computadores), deve-se agrupar as N tarefas em P processos, onde P equivale ao número de elementos de processamento (ou núcleos) disponíveis. Esta aglomeração permitirá aumentar a granularidade de cada processo e diminuir a comunicação necessária para atingir o objetivo final.

Cada processo receberá um bloco de dados de vet, contendo  $(N / P)$  elementos. Caso não seja possível obter uma divisão exata de  $(N / P)$ , as posições excedentes de vet devem ser

atribuídas ao último processo disponível. Uma alternativa ao resto de  $(N / P)$  é o espalhamento desses elementos excedentes aos primeiros processos disponíveis para processamento.

Com esta aglomeração, a operação de redução e o broadcast não ocorrerão mais em função de  $N$  valores de `vet`, mas sim em função de  $P$  processos.

Mapeamento (versão de particionamento por dados):

Considerando que os nós do cluster possuem um desempenho esperado homogêneo, o mapeamento de  $P$  processos em PROC Elementos de Processamento ocorrerá por meio de uma fila circular (Round-Robin). Neste caso, se  $P == PROC$ , então cada Elemento de Processamento receberá exatamente um processo.

Caso o desempenho dos nós do cluster seja diferente, então o mapeamento dos nós do cluster deve ser dinâmico, determinado em tempo de execução, considerando a heurística de atribuição ao nó com a menor carga de trabalho em andamento (determinada por alguma métrica de desempenho com o número de processos na fila de pronto para execução, uso de CPU, quantidade de bytes trocados em swap de disco, entre outros).