



UniAtenas

Avaliação de checagem optativa: MPI e Clusterização

Prof. Me. Luis Vinicius

Objetivo

Executar, analisar e modificar um programa MPI que estima o valor de π por meio do método de Monte Carlo (MMC) distribuído entre múltiplos processos.

1 Instruções gerais

Os arquivos com a resolução (códigos e respostas em texto) devem ser enviados compactados para `luis.professor@uniatenas.edu.br` e também, opcionalmente, **commitados** no seu repositório **forkado** do repositório oficial da disciplina: github.com/LuisVCSilva/clusterizacao_servidores_2025.

Um pouco sobre o MMC

O método de Monte Carlo foi originalmente proposto como uma técnica para resolver problemas de integração e simulação estatística durante a construção da bomba atômica usando números aleatórios. Um dos trabalhos mais influentes nesse contexto foi publicado por Nicholas Metropolis et. al. em 1953, no qual introduziram métodos computacionais eficientes para cálculos de estado de equilíbrio em sistemas físicos [?].

O método de Monte Carlo (MMC) para estimar o valor de π consiste em gerar pontos aleatórios dentro de um quadrado de lado 1 e calcular a razão entre a quantidade de pontos que caem dentro do quarto de círculo inscrito nesse quadrado.

Mais formalmente, considere um quadrado de lado 1 no primeiro quadrante do plano cartesiano, com o canto inferior esquerdo na origem. O quarto de círculo tem raio 1, definido pela equação:

$$x^2 + y^2 \leq 1$$

A área do quarto de círculo é $\pi/4$ e a área do quadrado é 1. Assim, a razão entre os pontos dentro do círculo e o total de pontos gerados aproxima $\pi/4$.

Sejam N o número total de pontos gerados e M o número de pontos dentro do círculo, a estimativa de π é dada por:

$$\pi \approx 4 \times \frac{M}{N}$$



No algoritmo paralelo, a tarefa de gerar e contar os pontos é dividida entre os processos, e os resultados são agregados para calcular a estimativa final. A figura abaixo ilustra como o método aproxima a constante Pi.

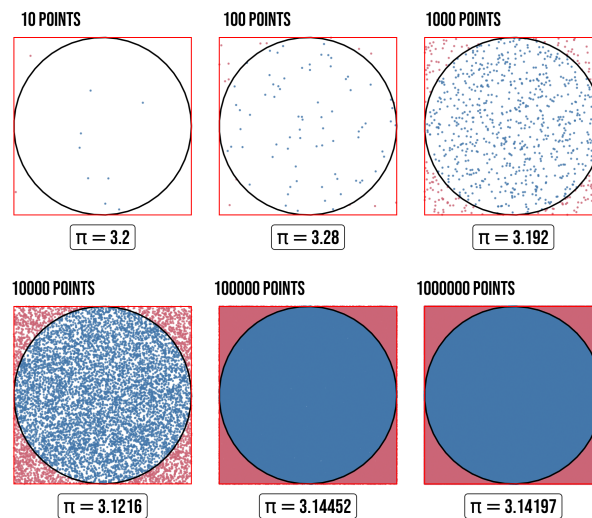


Figura 1: Exemplo ilustrativo do método Monte Carlo: pontos gerados aleatoriamente dentro do quadrado e a área do quarto de círculo (área sombreada). Retirado de <https://mauriciocely.github.io/blog/2020/08/05/estimating-pi-using-the-monte-carlo-method/>

Arquivo base

O código que você deve utilizar está neste diretório:

```
monte_carlo.c
```

Compile com:

```
mpicc monte_carlo.c -o monte_carlo
```

Execute com:

```
mpirun -np 4 ./monte_carlo
```

Parte 1 — Execução básica

1.1. Saída esperada

Execute o programa com 4 processos. Copie aqui a saída do terminal:

[COLE AQUI SUA SAÍDA]



Parte 2 — Análise de funcionamento

2.1. O que cada processo faz neste código?

Resposta:

2.2. Por que o valor estimado de π varia a cada execução?

Resposta:

2.3. Qual o papel da função `MPI_Reduce` nesse código?

Resposta:

Parte 3 — Modificação

3.1. Modifique o código para que cada processo imprima a quantidade de pontos gerados e quantos caíram dentro do círculo.

O processo 0 ainda deve calcular e exibir o valor estimado de π com base na soma global dos acertos.

Faça um **commit** com sua **modificação** e anexe abaixo o código completo.



3.2. Copie aqui a saída do seu programa modificado:

[SAÍDA DO PROGRAMA MODIFICADO]

Parte 4 — Análise com utilitários Linux

4.1. Use o comando `time` para medir o tempo de execução do programa com 2, 4 e 8 processos.

Processos	Tempo (real)
2	
4	
8	

4.2. O uso de CPU foi distribuído de forma equilibrada entre os processos?

Resposta:

Referências

- Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). *Equation of State Calculations by Fast Computing Machines*. Journal of Chemical Physics, 21(6), 1087. doi:10.1063/1.1699114. Disponível em: <https://bayes.wustl.edu/Manual/EquationOfState.pdf>