

Implementação do Método de Monte Carlo Distribuído utilizando MPI

Murillo Freitas Bouzon

Resumo—O uso de HPC para a resolução de problemas científicos tem se tornado cada vez mais comum devido ao avanço dos *hardwares* das máquinas de hoje. Uma das abordagens de HPC é o uso de programação distribuída, onde um programa é dividido para diferentes máquinas para ser executado. Seguindo essa linha, surgiu o padrão MPI para facilitar a escrita de programas feitos para serem executados em *clusters*. Com isso em mente, neste trabalho é feita a implementação do método Monte Carlo de forma distribuída utilizando MPI, para o cálculo do volume de um objeto. Os resultados apontam que foi possível calcular o volume do objeto de forma distribuída. Além disso, o tempo de processamento diminuiu quando o número de nós utilizados aumentou.

Index Terms—Integração numérica, Toroide Parcial, Método de Monte Carlo

I. INTRODUÇÃO

Com o avanço da capacidade dos *hardwares* utilizados nos computadores, a complexidade dos problemas a serem resolvidos também aumentou, tanto em questão de tempo e espaço necessário para encontrar a solução.

Isso fez com que surgisse o termo HPC (*High Performance Computing*) para a resolução de problemas computacionais. HPC é um conjunto de abordagens que utiliza super-computadores e/ou *clusters*, um conjunto de computadores que trabalham como um só, conectados por uma rede de alto desempenho.

Desta forma, a programação feita em *clusters*, também chamada de programação distribuída, começou a ser cada vez mais comum para o estudo e resolução de problemas científicos. Em programação distribuída, os processadores não compartilham a mesma memória, tendo que trocar informações em forma de mensagens entre eles.

O surgimento do conceito de programação distribuída acabou trazendo diversos modelos de forma que tornassem mais fácil o modo de programar em *clusters*. Um importante modelo é o MPI (*Message Passing Interface*), proposto em [1]. O MPI é um padrão para troca de mensagens entre os processadores de um *cluster*, criado para facilitar o desenvolvimento de programas que utilizam programação distribuída.

O MPI pode ser encontrado em diversos trabalhos da literatura que utilizam programação distribuída, podendo o citar o trabalho de [2] que adaptou o algoritmo bio-inspirado *Firefly* para ser feito de forma paralelo, distribuindo os vaga-lumes em diferentes máquinas utilizando o MPI. Os resultados mostraram que a otimização não teve muita diferença entre a versão sequencial do *Firefly*, porém foi possível obter uma melhoria no tempo de processamento.

Outro trabalho que também utilizou programação distribuída com MPI foi [3], onde foi proposto a execução do *Google*

Tensor Flow em grandes *clusters* utilizando MPI. O trabalho foi avaliado utilizando um *cluster InfiniBand* e a avaliação mostrou a eficiência da implementação proposta, tendo um aumento na velocidade.

Um método clássico que tem como característica a sua facilidade de ser distribuído entre várias máquinas é o método de Monte Carlo, onde é necessário repetir diversas iterações sucessivamente para poder se aproximar de uma solução.

Sendo assim, este trabalho tem como objetivo implementar o método de Monte Carlo utilizando o padrão MPI para aplicar o método de forma distribuída com o intuito de melhorar o seu desempenho, aplicado no cálculo do volume de um toroide parcial feito no trabalho anterior.

II. METODOLOGIA

A metodologia deste trabalho se baseia em resolver o problema proposto em [4], que trata-se da integração de um toroide parcial para encontrar o seu volume. Todavia, o padrão MPI também é aplicado na metodologia para o cálculo ser feito de forma distribuída. A Figura 1 ilustra a parte do toroide no espaço Euclidiano.

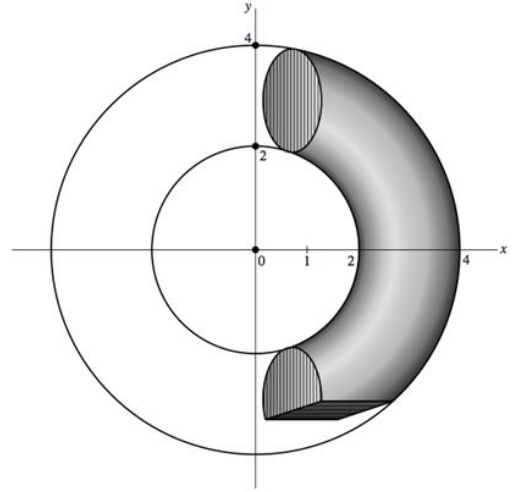


Figura 1: Ilustração de um toroide parcial.

A interseção do toroide parcial com um cubo é dada pelas restrições da Equação (1);

$$H(x, y, z) = \begin{cases} 1, & \text{if } (\sqrt{x^2 + y^2} - 3)^2 + z^2 \leq 1, \\ & x \geq 1 \text{ e } y \geq -3 \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

sendo o limite de integração $\Omega = [-1, 4] \times [-3, 4] \times [-1, 1]$ e o volume $V = 42$.

O método recebe como entrada um valor N , sendo o número de pontos que são calculados e um número T , sendo o número de nós utilizados para o processamento. Desta forma, os N pontos são divididos entre os T nós utilizados de forma distribuída. Dessa forma, cada nó calcula a função $H(x, y, z)$ para $\frac{N}{T}$ pontos aleatórios e retorna para o nó mestre o número de pontos dentro do espaço de integração. O nó mestre calcula a média dos pontos recebidos pelos outros nós para encontrar o volume do toroide.

III. EXPERIMENTOS E RESULTADOS

Para avaliar o método implementado neste trabalho, foi realizado um experimento para avaliar o tempo levado de acordo com o número de nós utilizados, variando o número de iterações N entre 10 a 10^6 e o número de nós T entre 1, 2, 4 e 8. Foi calculado o tempo em segundos para cada valor de N e T diferentes e o volume encontrado pelo método.

A Figura 2 apresenta um gráfico com os resultados obtidos, onde o eixo x representa o número de iterações N utilizado, o eixo y o tempo de processamento em segundos e as cores o número de nós utilizados.

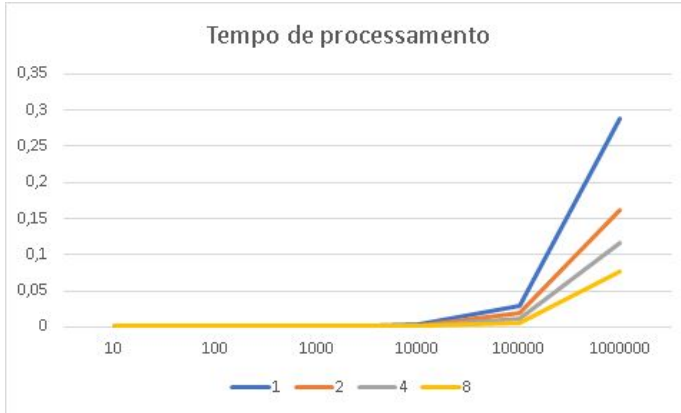


Figura 2: Gráfico comparando o tempo de processamento entre os diferentes números de nós utilizados.

Observando o gráfico é possível afirmar que quando o número de iterações é menor que 10000, o tempo de processamento entre os diferentes valor de T utilizados é bem próximo, porém para valores de $N \geq 10000$ o tempo cresce exponencialmente. Além disso, é possível afirmar que o tempo de processamento reduz de acordo com o aumento do número de nós escolhido para $N \geq 10000$.

A Tabela I apresenta os resultados do experimento realizado, onde cada célula representa o volume encontrado para cada valor de N e T selecionado para o teste. Sabendo que o volume real do objeto é aproximadamente 22.08, quando $N = 1000000$, o resultado calculado se aproxima bem do resultado real.

IV. CONCLUSÃO

Neste trabalho foi feita a implementação do método de Monte Carlo, utilizando MPI para realizar a integração numérica para calcular o volume de um toroide parcial,

$N \backslash T$	1	2	4	8
10	25.2	25.2	16.8	33.6
100	26.88	22.68	20.16	26.88
1000	22.34	22.09	22.84	24.86
10000	21,78	22.23	21.57	22.54
100000	22.01	22.08	22.15	21.9
1000000	22.08	22.08	22.09	22.09

Tabela I: Tabela de resultados.

utilizando mais de um nó e reduzindo o tempo necessário para realizar cálculo.

Os resultados mostraram que quando o $N < 10000$, o tempo de processamento com diferentes números de nós não possui uma diferença notável, porem para valores de $N \geq 10000$ o tempo varia bastante, sendo que para maiores quantidades de nós, o tempo acaba tendo uma redução maior. Além disso, foi possível calcular o volume do objeto em questão para todas as variações da quantidades de nós, onde o melhor valor encontrado foi utilizando $N = 10^6$.

REFERÊNCIAS

- [1] Message P Forum. Mpi: A message-passing interface standard. Technical report, Knoxville, TN, USA, 1994.
- [2] G. Linhares, G. Costa, L. Jacobson, L. Rodrigues, V. Ballabenute, G. Wachs-Lopes, and P. Rodrigues. Parallel approach to the firefly algorithm. In *2018 XLIV Latin American Computer Conference (CLEI)*, pages 744–748, Oct 2018.
- [3] Abhinav Vishnu, Charles Siegel, and Jeff Daily. Distributed tensorflow with mpi. *ArXiv*, abs/1603.02339, 2016.
- [4] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. Numerical recipes: the art of scientific computing, 3rd edition. In *SOEN*, 2007.