Método de Monte Carlo para o cálculo de integral utilizando MPI

Mariana Bastos dos Santos
Centro Universitário FEI
São Bernardo do Campo, São Paulo
unifmsantos@fei.edu.br

***Resumo*—O artigo apresenta a resolução de uma integral método de Monte Carlo utilizando *Message Passing Interface* (MPI).**

***Index Terms*—Integração Numérica, Monte Carlo, MPI**

I. Introdução

Uma integração de uma equação pelo método de Monte Carlo consiste em analisar uma amostra de uma distribuição de pontos que probabilisticamente está contido na área abaixo da curva da equação de interesse. Esse tipo de integração permite resolver integrais muito complexas por meio de aproximações que são menos custosas computacionalmente, do que a integração de forma analítica. Para isso, é utilizado o MPI que trata-se de um padrão para comunicação de dados em computação distribuída que por meio de uma biblioteca de funções permite seu uso. Sendo assim, o objetivo do presente trabalho é calcular uma integral por meio do método de integração de Monte Carlo utilizando MPI para computação distribuída.

II. Conceitos Fundamentais

A. Integração pelo Método de Monte Carlo

O Método de Monte Carlo estima a integral da função f sobre o volume multidimensional Ω aproximando I por:

$$I = Q_n = V.\langle f \rangle \tag{1}$$

onde $\langle f \rangle$ é a soma das amostras da função:

$$\langle f \rangle = \frac{1}{N} \sum_{i=1}^{n} f(xi) \tag{2}$$

ou seja:

$$\int_{\Omega} f(x)dx = V.\langle f \rangle \tag{3}$$

III. Message Passing Interface (MPI)

MPI é uma biblioteca padrão para comunicação em computação distribuída desenvolvida em 1993. Possui diversas funções que facilitam e padronizam o desenvolvimento de aplicações paralelas. Com MPI é possível programar para vários processadores, nós de um cluster, supercomputadores ou internet utilizando a mesma infraestrutura. As funções *Broadcast* e *Reduce* enviam e recebm mensagens para todos os processos, inclusive para eles mesmos. As funções *Gather* e *Scatter* espalham e reunem os dados.

IV. Implementação

Cálculo de Integral por Monte Carlo

$$\int_0^1 \frac{4}{1+x^2}\,dx$$

```
In [35]: n = 100

In [4]: def funcao(aleatorio,n):
            a = 0
            b = 4
            integral = 0.0
            for x in aleatorio:
                integral += 4/(1+x**2)

            return (b-a)/float(n)*integral

In [41]: %%time
         comm = MPI.COMM_WORLD
         nprocs = comm.Get_size()
         myrank = comm.Get_rank()

         if myrank == 0:
             N = n // nprocs
             samples = np.random.uniform(0,4,size=(nprocs,N,2))
         else:
             samples = None
         samples = comm.scatter(samples, root=0)

         mypi = funcao(samples,N) / nprocs

         result = comm.reduce(mypi, root=0)

         result = result.mean()

         if myrank == 0:
             print("Result = {0}".format(result))

         Result = 5.133108317106942
         Wall time: 1.99 ms
```

Figura 1. Implementação da integração por Monte Carlo para cálculo de integral utilizando MPI.

Neste trabalho foi implementado o Método de Monte Carlo para o cálculo de integral utlizando MPI. Para isso, foi utilizada a linguagem Python.

Para o cálculo, primeiramente, é feita a definição da função a ser calculada, como pode ser observado na Figura 1. Na sequência, cria-se um looping que irá iterar n vezes. Para cada iteração determina-se o valor de entrada da função a ser calcula por meio da geração aleatório dos números no intervalo entre a e b da integral, e em seguida, calcula-se a Equação 3.

É utilizado no desenvolvimento do método a biblioteca MPI. Como pode ser visto na Figura 1. A função *Scatter* é utilizada para distribuir as amostras entre os processos e a função *Reduce* é utilizada para obter os resultados parciais de cada um dos processos e os consolidar em um resultado global.

Sendo assim, é possível alcançar, após o reduce, o resultado da integral.

V. Experimentos e Resultados

Foi calculada a integral $\int_0^1 \frac{4}{1+x^2} dx$. O resultado do cálculo da integral utilizando MPI foi 5.13, e o tempo de execução foi de 1.99 milissegundos enquanto o método sem MPI obteve o resultado no tempo de execução de 4.98 milissegundos. Ou seja, há um ganho utilizando MPI, mas por ser uma integral simples o tempo de execução não é tão relevante quanto a um cálculo computacionalmente mais custoso. Por esse motivo, o MPI se torna uma alternativa para a melhora do desempenho de cálculos custosos.

VI. Conclusão

Esse tipo de integração permite resolver integrais muito complexas por meio de aproximações que são menos custosas computacionalmente, do que a integração de forma analítica. Isso pode ser observado nos resultados dos experimentos realizados. E ainda, o uso do MPI pode melhorar o desempenho do cálculo, visto que a biblioteca possibilita a computação distribuída.

Referências

[1] Press, William H., et al. "Numerical recipes in C++."The art of scientific computing 2 (1992): 1002.