DOCUMENTAÇÃO PI DE MONTE CARLO

ALUNOS:

PEDRO MARCELO - 58717

ELSON GOIS - 60771

PAULO ROBERTO - 58994

KELVIN LIMA - 58992

2024

VERSÃO 1.0

Sumário

1. Introdução

2. Método de Monte Carlo

3. Abordagens Implementadas

• 3.1 Multiprocessing

• 3.2 Threading

• 3.3 Concurrent Futures

• 3.4 Threading com Semaphore

4. Requisitos

5. Utilização

6. Código Fonte

7. Conclusão

8. Referências

1. Introdução

A documentação apresenta uma implementação do método de Monte Carlo para estimar o valor de Pi. O objetivo principal é demonstrar diferentes abordagens de implementação para calcular Pi utilizando processos, threads e outras técnicas de concorrência.

1. Método de Monte Carlo

O método de Monte Carlo é uma técnica estatística que utiliza números aleatórios para resolver problemas. Neste contexto, é utilizado para estimar o valor de Pi. O algoritmo gera pontos aleatórios dentro de um quadrado e conta quantos desses pontos estão dentro de um círculo inscrito no quadrado. Com base na proporção de pontos dentro do círculo para o total de pontos gerados, é possível estimar o valor de Pi.

1. Abordagens Implementadas
   1. Multiprocessing

Esta abordagem utiliza o módulo multiprocessing do Python para distribuir o cálculo em vários processos. Cada processo executa uma parte do cálculo de forma independente, aproveitando os recursos de processamento paralelo disponíveis no sistema.

* 1. Threading

Nesta abordagem, o cálculo é distribuído em várias threads utilizando o módulo threading. Cada thread executa uma parte do cálculo, permitindo a execução concorrente em sistemas com suporte a multithreading.

* 1. Concurrent Futures

Esta abordagem utiliza o módulo concurrent.futures.ThreadPoolExecutor para realizar o cálculo em várias threads de forma assíncrona. Ele fornece uma interface de alto nível para trabalhar com threads e simplifica a escrita de código concorrente.

* 1. Threading com Semáforo

Similar à abordagem de Threading, esta implementação utiliza um Semáforo para sincronização entre as threads. O Semáforo é usado para controlar o acesso concorrente a recursos compartilhados, garantindo consistência nos resultados.

Formula:

A black text on a white background

Description automatically generated

1. Requisitos

Para executar o programa, é necessário ter o Python instalado juntamente com os seguintes módulos: random, math, multiprocessing, threading, concurrent.futures e tkinter.

1. Utilização

O programa oferece uma interface gráfica simples onde o usuário pode inserir o número de pontos desejado e selecionar a abordagem de cálculo (Multiprocessing, Threading, Concurrent Futures ou Threading com Semaphore). Após clicar no botão correspondente, o resultado do cálculo de Pi e o tempo de execução serão exibidos na interface.

1. Código Fonte

O código fonte completo do programa está disponível para referência e modificação. Ele inclui as implementações das diferentes abordagens de cálculo de Pi, bem como a interface gráfica utilizando o módulo tkinter.

Git hub com código fonte:

https://github.com/Paulerou/PI-Monte-Carlo

1. Conclusão

A documentação conclui ressaltando a importância do método de Monte Carlo como uma ferramenta poderosa para resolver uma variedade de problemas, incluindo a estimativa de Pi. Além disso, destaca a relevância das diferentes abordagens de concorrência para melhorar o desempenho e a eficiência computacional do cálculo.

1. Referências

Metropolis, N., & Ulam, S. (1949). O método de Monte Carlo. Revista da Associação Estatística Americana, 44(247), 335-341.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (2007). Receitas Numéricas 3ª Edição: A Arte da Computação Científica. Editora da Universidade de Cambridge.

Fundação Python Software. (2024). Referência da Linguagem Python, Versão 3.10. Disponível em: https://docs.python.org/3.10/

Fundação Python Software. (2024). Multiprocessing - Paralelismo baseado em processos. Disponível em: https://docs.python.org/3/library/multiprocessing.html

Fundação Python Software. (2024). Threading - Paralelismo baseado em threads. Disponível em: https://docs.python.org/3/library/threading.html

Fundação Python Software. (2024). Futuros Concorrentes - Lançando tarefas paralelas. Disponível em: https://docs.python.org/3/library/concurrent.futures.html

Fundação Python Software. (2024). Tkinter - Interface Python para Tcl/Tk. Disponível em: https://docs.python.org/3/library/tkinter.html