Universidade Tecnológica Federal do Paraná
Campus Pato Branco
Departamento Acadêmico de Informática
Disciplina de Sistemas Operacionais - SO25CP
Curso de Engenharia de Computação

Relatório Trabalho 3 - Threads
Cálculo do valor de Pi utilizando aproximações por série

Bruno Duarte

07 de Agosto de 2020

1 Introdução

Este relatório tem como objetivo apresentar os resultados obtidos com a implementação de cálculos para aproximação do número Pi (π). O cálculo computacional foi realizado utilizando *threads*. Para aproximação utilizaram-se as séries de Leibniz-Gregory-Mendel e de Nilakantha.

O uso de *threads* nesse trabalho tem como objetivo a diminuição no tempo de execução do cálculo, pois divide uniformemente as etapas de cálculo entre diferentes processadores, possibilitando o cálculo de N termos de uma série de maneira simultânea.

A seção 2 apresenta as expressões matemáticas das séries utilizadas, enquanto a seção 3 detalha as ferramentas utilizadas e alguns detalhes de implementação. A seção 4 apresenta os resultados e por fim, a seção 5 apresenta algumas conclusões.

2 Formulação matemática das séries utilizadas

Nessa seção apresentam-se as expressões matemáticas para as séries de Leibniz-Gregory-Mendel e de Nilakantha implementadas. Ambas as séries foram retiradas de `http://numbers.computation.free.fr/Constants/Pi/piSeries.html`. Deve-se ressaltar que as séries foram reescritas visando isolar o valor de π.

Série de Leibniz-Gregory-Madhava:

$$\pi = 4 \cdot \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots\right) \tag{1}$$

Série de Nilakantha:

$$\pi = 4 \cdot \left(3 + \frac{1}{2 \cdot 3 \cdot 4} - \frac{1}{4 \cdot 5 \cdot 6} + \frac{1}{6 \cdot 7 \cdot 8} \dots\right) \tag{2}$$

3 Informações referentes a implementação

Essa seção apresenta alguns aspectos da implementação.

Os algoritmos utilizados para o cálculo, bem como os experimentos realizados, foram executados em uma máquina com processador Intel® Core™ i5-7200U 2.5 GHz - 3.1 GHz, 8 GB de RAM, sistema operacional Windows 10 Home com subsistema linux Ubuntu 18.04.1. O algoritmo foi implementado na linguagem C (gcc 7.4.0). Utilizaram-se as bibliotecas *stdio.h, stdlib.h, math.h pthread.h* e *unistd.h*.

Durante a execução dos experimentos, variou-se o número de *threads* visando comparar a performance do cálulo mediante tal variação. Os números de *threads* utilizados foram 1, 2, 4, 8 e 16 e ambas as séries foram submetidas a todos eles.

O tempo de execução de cada teste foi obtido utilizando o comando *time* em linha de compilação. Utilizou-se esse método pois ele retorna o tempo real de execução, além do tempo utilizado pelo sistema. Para fins de comparação, considerou-se o tempo real.

Devido a isso, fez-se necessário realizar uma execução do código para cada combinação série/número de *threads*, pois em cada teste é preciso alterar manualmente (em código) a série utilizada, bem como o número de *threads*.

O número de termos calculados em cada série foi de 1 bilhão.

4 Resultados e discussões

Os resultados obtidos para as execuções de cada série permitiram construir um gráfico de comparação entre os tempos de execução, que pode ser visto na Figura 1.

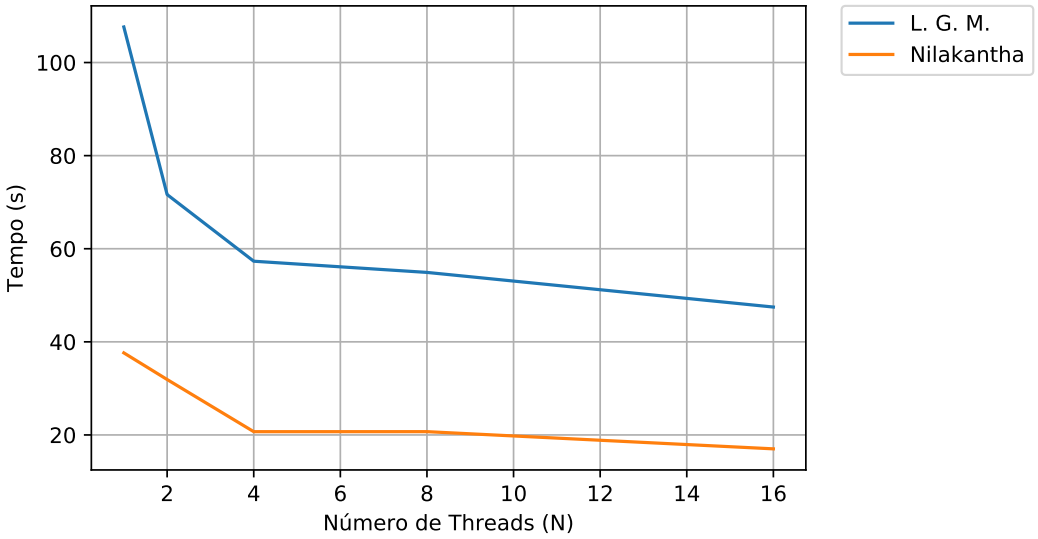

Figura 1: Relação entre o número de threads e o tempo necessário para calcular pi através de séries iterativas.

Através da Figura 1 é possível observar que a série de Nilakantha apresenta melhor desempenho que a de Leibniz-Gregory-Madhava em quase todos os números de *threads*, e em alguns casos é cerca de 60 % mais rápida.

Outro fator a se considerar é o grau de exatidão de cada série. O valor médio de π calculado com a série de Leibniz-Gregory-Madhava foi de $3,1415926525892095$, apresentando uma exatidão de 8 casas decimais. Para a série de Nilakantha, o valor médio de π obtido foi de $3,141592653589791$ com exatidão de 14 casas decimais.

5 Conclusões

A utilização de *threads*, que visa acelerar a execução de um processo, mostrou-se uma excelente ferramenta para a aplicação em questão, pois, como mostra a Figura 1, o tempo de execução de cada série diminuiu com o aumento do número de *threads*. Além disso, é possível concluir que a série de Nilakantha mostrou-se mais eficaz que a de Leibniz-Gregory-Madhava, pois tanto seu tempo de execução como seu grau de exatidão demonstraram-se de melhor qualidade.

Como sugestão de trabalhos futuros, fica a possibilidade de implementar outras séries para comparação.