Avaliação Substitutiva

Programação Concorrente e Distribuída

Nome: Davi Castilho Lopes

Matrícula: UC22100801

O que são Threads?

- Threads são fluxos de execução independentes dentro de um processo.

Cada programa, ou processo, possui normalmente um fluxo de controle. Assim o programa é executado sequencialmente passo a passo com seu único fluxo de controle. A partir desse ponto que threads entram em contexto, com os threads é possível ter mais de um fluxo em uma aplicação.

- Permitem que múltiplas tarefas ocorram simultaneamente no mesmo ambiente de aplicativo.

Nesse contexto o aplicativo que utiliza essa múltipla execução os threads atuam com um grande grau de independência uma da outra, portanto, se temos muitos threads executando em paralelo no sistema é análogo a múltiplos aplicativos executando em paralelo em um computador**.** Portanto, cada thread possui seu próprio fluxo de controle e compartilha o mesmo espaço de endereçamento do aplicativo.

Como threads funcionam?

Os threads funcionam recebendo e executando instruções de um processador. Portanto, uma CPU com um thread tem apenas uma linha de execução de trabalho realizando uma ação por vez. Logo, processadores classificados como multithread são

mais vantajosos, já que dão a possibilidade de operar em diversas frentes ao mesmo tempo.

Tempo de execução

O uso de threads pode afetar significativamente o tempo de execução de um algoritmo. Por exemplo o paralelismo, que permite a execução simultânea de operações, apresenta mais eficiência em programas que realizam tarefas que podem ser executadas em paralelo. É extremamente eficaz em sistemas com vários processadores e núcleos.

Para realizar uma avaliação de desempenho, pode ser considerado várias métricas, como uso de memória, tempo de execução e consumo de energia. A implementação do algoritmo determina sua eficiência, e mais etapas geralmente significam mais instruções executadas pelas CPU.

Programação Concorrente x Paralela

Ambas lidam com a coordenação, comunicação e acesso aos recursos computacionais. A programação concorrente está relacionada à interação entre tarefas que ocorrem simultaneamente, já a programação paralela envolve a execução simultânea de tarefas em vários processadores. É preciso entender esses dois conceitos para chegar a uma conclusão de qual vai ter mais eficiência e o quão impactante vai ser cada um dos modelos de programação, diante disso, algoritmos paralelos teoricamente são mais eficientes no modelo PRAM (Parallel Random Acess Machine), e podem não ter bom desempenho em máquinas reais. A arquitetura paralela utilizada pode afetar o desempenho dos algoritmos, pode chegar a uma conclusão de que considerar a natureza do problema (se é um problema paralelizável ou não) é fundamental para desenvolver algoritmos eficazes.

Projeto Experimental

Relatório geral

A relação de tempo de execução entre as quatro versões do algoritmo solicitado:

Primeira versão – Versão com apenas a THREAD principal.

- Soma do tempo de execução total de todas as 10 execuções: 71844 ms.
- Média de tempo de execução: 7184,4 ms.

Segunda versão – Versão com 3 threads.

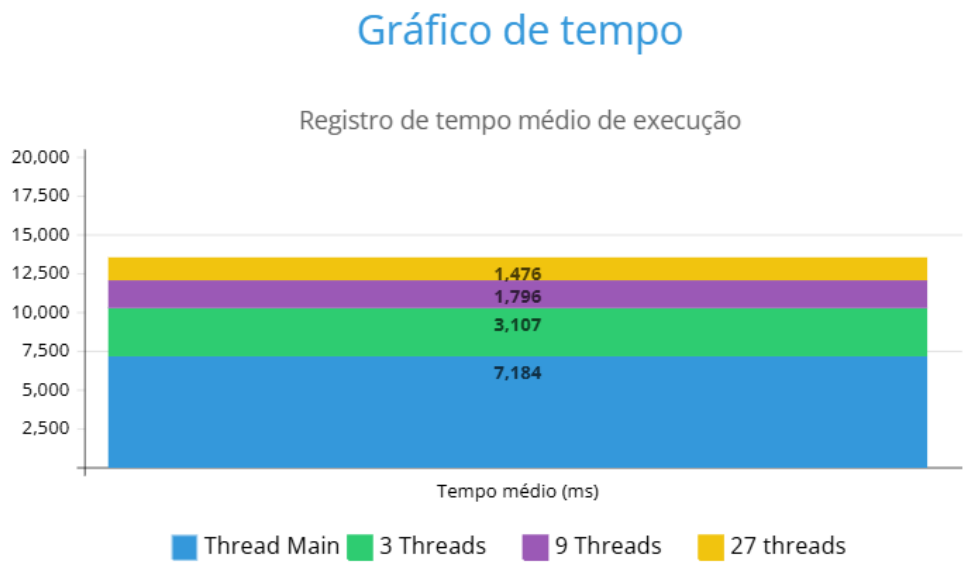
- Soma do tempo de execução total de todas as 10 execuções: 31073 ms.
- Média de tempo de execução: 3107,3 ms.

Terceira versão – Versão com 9 threads.

- Soma do tempo de execução total de todas as 10 execuções: 17963 ms.
- Média de tempo de execução:1796,3 ms.

Quarta versão – Versão com 27 threads.

- Soma do tempo de execução total de todas as 10 execuções: 14769 ms.
- Média de tempo de execução:1476,9 ms.



Referências bibliográficas:

- https://www.devmedia.com.br/trabalhando-com-threads-em-java/28780
- https://www.devmedia.com.br/programacao-com-threads/6152
- https://pt.stackoverflow.com/questions/98705/quais-maneiras-de-medir-o-desempenho-de-um-algoritmo
- https://learn.microsoft.com/pt-br/dotnet/standard/threading/threads-and-threading
- https://pt.wikipedia.org/wiki/Programa%C3%A7%C3%A3o_concorrente
- https://ulbra-to.br/encoinfo/wp-content/uploads/2020/03/Sistema_Concorrente.pdf