

Disciplina: Performance em Sistemas Ciberfísicos

Professor: Guilherme Schnirmann

Nome Estudante: Ary Felipe Farah e Silva

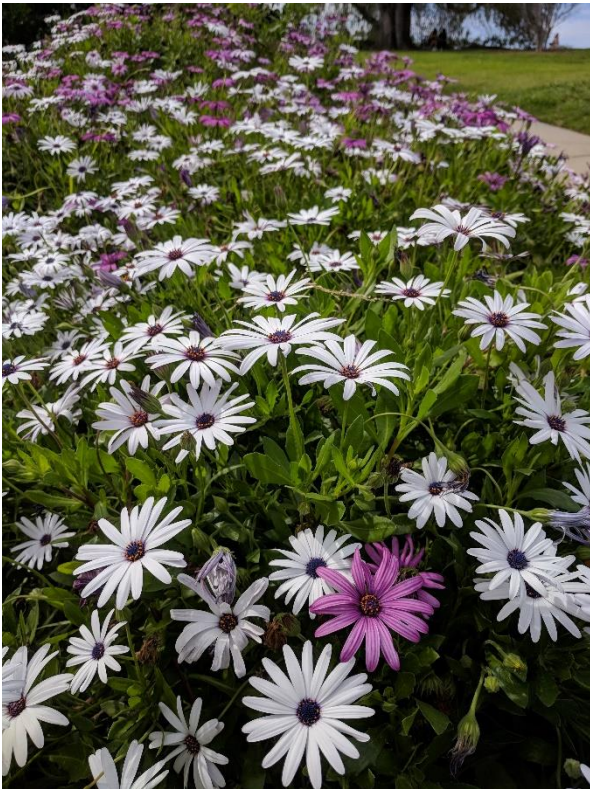
Atividade Prática / Relatório

Descrição da Atividade:

Esta atividade tem como objetivo buscar o entendimento da criação de múltiplas threads e seus benefícios em um problema de processamento de imagens.

Utilize o código fonte disponibilizado no AVA que resolve o problema single-thread.

Deseja-se pintar todas as flores brancas:



Exercício:

Crie agora um método para executar o processamento da imagem multi-threaded:



Faça um código que permita criar quantas threads forem desejadas.

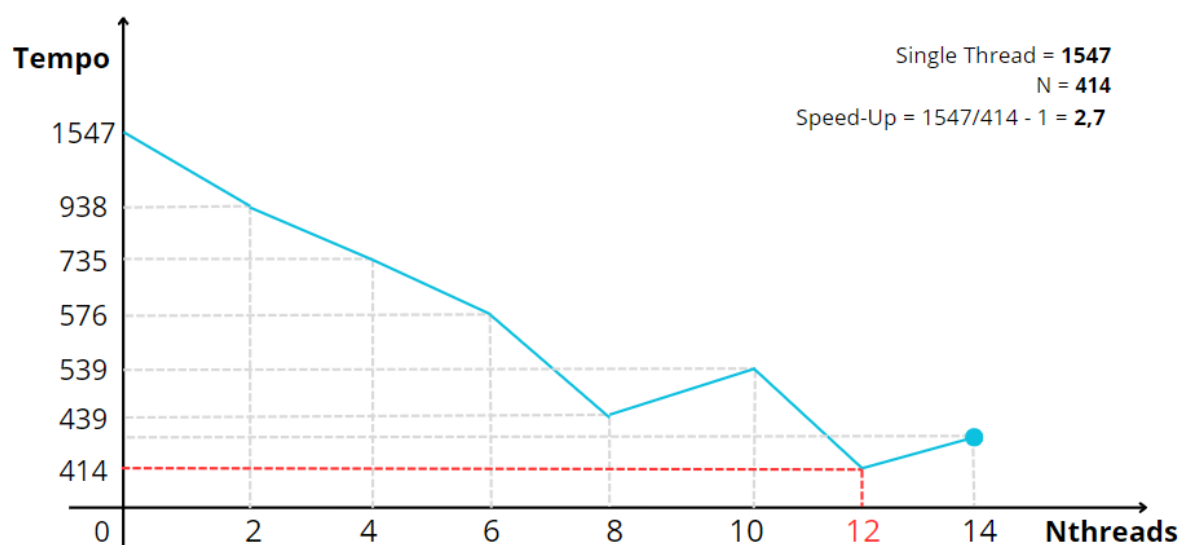
1. Meça o tempo (utilizando funções do java ou python) para executar o programa sem thread.
2. Agora meça o tempo utilizando somente uma thread. O tempo é maior ou menor? Explique.
[Mesmo tempo, pois o desempenho sem threads ou com uma thread é o mesmo.](#)
3. Agora meça o tempo utilizando duas threads e quatro threads. Explique os tempos, foram melhores?
[Sim, os tempos são melhores em relação aos anteriores](#)
4. Faça várias medidas e gere um gráfico relacionando o número de threads com o tempo. Apresente aqui o gráfico e explique a relação com o hardware do seu computador.
5. Agora fixe o número de threads no melhor número encontrado no gráfico anterior e gere um gráfico variando o tamanho da imagem (diminuindo). Para isso calcule o speed-up (em relação a execução single-thread). O gráfico será de speed-up X resolução.

[Respostas nos gráficos abaixo!](#)

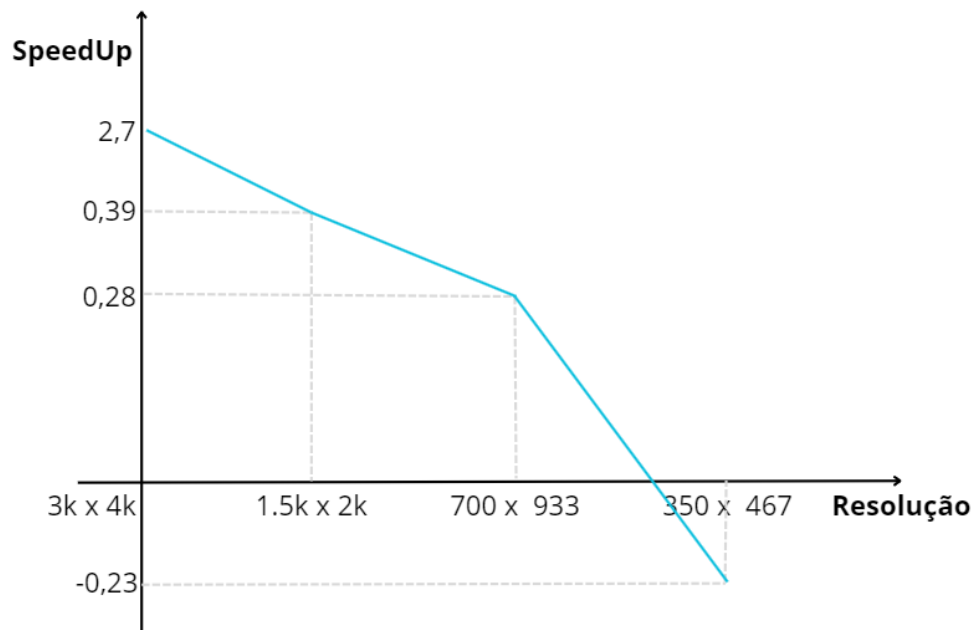
6. Na sua entrega deve ser apresentada a imagem das flores pintadas por uma cor escolhida pela equipe (diferente da do professor).



7. Deve-se apresentar o gráfico de tempo x número de threads com o apontamento do melhor N (número de threads)



8. Deve-se apresentar o gráfico de speed-up x resolução (chegue até uma resolução bem baixa – exemplo: 180x120).



Exercício teórico:

Pesquise o que são deadlocks, demonstre sua resposta com um exemplo. Explique também as condições para acontecerem deadlocks e possíveis soluções.

Deadlock: quando dois ou mais processos ficam presos esperando por recursos que estão sendo usados em outros processos, resultando em uma espera infinita.

Exemplo: processo A usa recurso 1 e quer recurso 2, enquanto o processo B usa o recurso 2 e deseja o recurso 1. Assim, os dois processos ficam esperando o recurso utilizado pelo outro.

Soluções:

- **Prevenir:** Usar algoritmos para evitar deadlocks, como Algoritmo do Banqueiro.
- **Resolver:** Deixar deadlocks acontecerem, mas detectá-los e resolvê-los depois