The background is a dark blue gradient. In the top-left corner, there are two overlapping parallelogram shapes, one blue and one light green. In the bottom-left, there is a circular inset showing a close-up of a circuit board. In the top-right, there is a 3D perspective view of a circuit board layout.

# Geração de fractais de Mandelbrot usando MPI

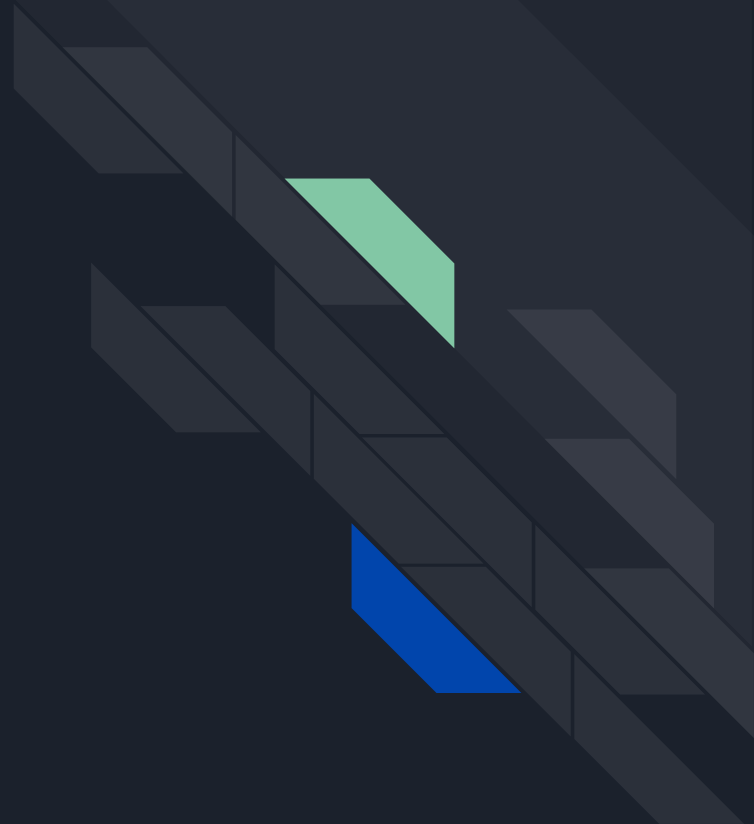
# Índice

Ambiente dos experimentos

Estratégia de Paralelização

Experimentos realizados

Resultados e conclusão





# Ambiente dos experimentos

Máquina virtual: Oracle VM VirtualBox

SO: Ubuntu 19.04

Cores: 2 Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz

Threads: 4



# Estratégias de Paralelização

A estratégia utilizada foi semelhante ao meu trabalho final de Sistemas Operacionais, nesse trabalho havia um buffer compartilhado, o server, e vários buffer particulares, clients, o objetivo do programa era permitir que qualquer um dos clients pudesse comunicar-se com o server sem atrapalhar o outro.

Já neste trabalho, foi criado um processo master, processo zero por default, que fica esperando uma ou mais mensagens contendo os dados do fractal de mandelbrot. O cálculo do fractal é feito pelos outros processos, em que cada um deles constrói um número 'x' de frames, e ao terminar de calcular os frames, esse processo envia os dados para o processo master, que por fim junta todos os dados em um único vetor.



# Experimentos Realizados

Foram testados três combinações do tamanho do fractal de mandelbrot

- A. Width = 1024, Frames = 32
- B. Width = 1024, Frames = 64
- C. Width = 256, Frames = 256

Em relação ao número de threads: 2, 4, 8





# Resultados e Conclusão

Sequencial

W   F	1024   32	1024   64	256   256
Seg	42.9514	90.1465	24.2417

2 Threads

W   F	1024   32	1024   64	256   256
Seg	25.6385	48.7358	12.6137

4 Threads

W   F	1024   32	1024   64	256   256
Seg	16.3945	32.2100	7.7122

8 Threads

W   F	1024   32	1024   64	256   256
Seg	16.8248	32.6241	7.4310



# Resultados e Conclusão

Tabela de SpeedUp em segundos

W   F	1024   32	1024   64	256   256
Seq.	---	---	---
2 Threads	1.6753	1.8497	1.9219
4 Threads	2.6199	2.7897	3.1433
8 Threads	2.5529	2.7632	3.2622



# Resultados e Conclusão

Há três pontos importantes para observar:

1. Há melhora no tempo com o aumento no número das threads.
2. Com 4 threads para mais o tempo se estabiliza e até pode piorar com o aumento do número de threads.
3. O speedup em uma thread em relação ao sequencial é maior quando o número de frames é maior.