

Otimização do Programa da Aula 8

Desenvolvida por: Raul A. Gonzalez Augusto, RA.: 211023698

Objetivo: Observar o impacto da memória cache no tempo de execução de diferentes tamanhos de matrizes.

Computador Usado:

Avell High Performance A52 LIV

- Processador: Intel® Core™ i5-10300H CPU @ 2.50GHz
 - Cores: 4
 - Threads: 8
 - Frequência Máxima: 4.50 GHz
 - Cache: 8 MB Intel® Smart Cache
 - Bus Speed: 8 GT/s
- Placa de Vídeo: NVIDIA GeForce GTX 1650 Ti e Intel® UHD Graphics
- RAM: 16 GB DDR4-2666 Dual Channel
- Armazenamento: 500 GB nvme m.2 SSD 2GB/s E 1 Tera HDD 5400 rpm
- Sistema Operacional: Windows 10 pro 64bits

Programas

Compilado no gcc em uma máquina virtual Linux.

Executando o programa sem otimização

K = 1

Tempos:

0.213991

0.214469

0.212395

0.215939

0.219624

Media: 0.2152836

K = 16

Tempos:

0.208637

0.208462

0.207822

0.213020

0.210062

Media: 0.2096006

K = 128

Tempos:

0.212066

0.211373

0.210955

0.214550

0.209270

Media: 0.2116428

Porque tempos 1, 2, 3 são praticamente iguais?

Pois o numero de pixels processados se mantem, no for do main, esta multiplicando com $k * k$ e no for do processa esta dividindo por k a altura e a altura.

Executando programa com i, j invertidos

K = 1

Tempos:

2.055044

1.846362

1.893870

1.904168

1.583692

Media: 1.8566272

Performance: -762.40%

K = 16

Tempos:

0.365693

0.353722

0.357552

0.375835

0.362548

Media: 0.36307

Performance: -73.21%

K = 128

Tempos:

0.194495

0.197723

0.195964

0.191013

0.194073

Media: 0.1946536

Performance: 1.69%

K = 8000

Fiquei com curiosidade de executar com um k em um valor alto e coloquei aleatoriamente 8000 e o resultado do tempo de processamento foi 0.000001.

Analisando os resultados

O k não altera o numero de pixels processados pois, mantem-se constante a quantidade de pixels da imagem, a diferença é que com um k maior, processa menos pixels por vez, mas o processa é chamado mais vezes.

Uma imagem preta ficou assim pois calculamos o tom de cinza de cada RGB, deixando tudo preto.

$$T4/t1 = 8.62$$

$$T5/t2 = 1.73$$

$$T6/t3 = 0.91$$

A relação foi cada vez diminuindo pois cada vez que processa menos pixel por vez, mais rápido o programa fica, pois assim faz maior proveito da memoria cache.