

Otimização do Programa da Aula 2

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

Objetivo: Otimizar o programa Unroll.c

Computador Usado:

Avell High Performance A52 LIV

- Processador: Intel® Core™ i5-10300H CPU @ 2.50GHz
 - Cores: 4
 - Threads: 8
 - Frequencia Maxima: 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 mingw64, porte do gcc para Windows. Apresentarei 5 tempos com a media deles e depois farei uma porcentagem dos ganhos de desempenho da otimização em relação a aplicação original.

Executando o programa original

Tempos:

1. 4.928
2. 4.968
3. 4.716
4. 4.698
5. 4.874

Média: 4.8368

Primeira Otimização

Remoção de ifs e desenrolando for para 2 pixels.

Tempos:

1. 5.358

2. 5.217

3. 5.311

4. 5.252

5. 5.167

Média: 5.243

Porcentagem: -8.39811446%

Segunda Otimização

Desenrolando for para 4 pixels.

Tempos:

1. 4.959

2. 4.937

3. 4.990

4. 4.946

5. 4.948

Média: 4.956

Porcentagem: -2.4644393%

Terceira Otimização

Desenrolando for para 4 pixels e Juntando os armazenamentos feitos na matriz da imagem.

Tempos:

1. 3.560

2. 3.542

3. 3.563

4. 3.561

5. 3.549

Média: 3.555

Porcentagem: 26.5009924%

Quarta Otimização

Desenrolando for para 8 pixels.

Tempos:

1. 3.415
2. 3.422
3. 3.457
4. 3.447
5. 3.425

Média: 3.4332

Porcentagem: 29.0191862%

Quinta Otimização

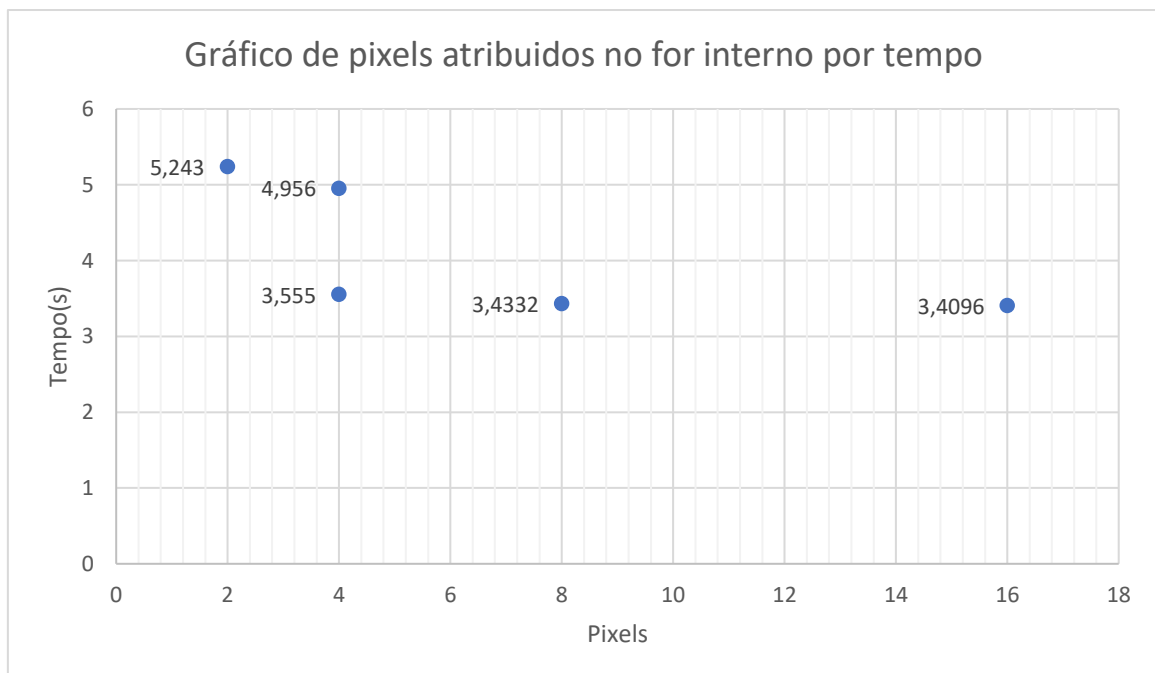
Desenrolando for para 16 pixels.

Tempos:

1. 3.410
2. 3.415
3. 3.438
4. 3.384
5. 3.401

Média: 3.4096

Porcentagem: 29.5071121%



Observando o gráfico notamos que colocando as gravações juntas diminui bastante o tempo, e o ganho de tempo desenrolando o for para mais de 4 atribuições foi ínfimo.