

PROGRAMAÇÃO PARALELA T8

Natan Luiz Paetzhold Berwaldt

Parte 1

```
__global__  
void CalcFrame(unsigned char* pic, int width)  
{  
    int frame = threadIdx.x;  
    printf("%d\n",frame);  
    for (int row = 0; row < width; row++) {  
        for (int col = 0; col < width; col++) {  
            float fx = col - 1024/2;  
            float fy = row - 1024/2;  
            float d = sqrtf( fx * fx + fy * fy );  
            unsigned char color = (unsigned char) (160.0f + 127.0f *  
                                                    cos(d/10.0f - frame/7.0f) /  
                                                    (d/50.0f + 1.0f));  
  
            pic[frame * width * width + row * width + col] = (unsigned char) color;  
        }  
    }  
}
```

```
unsigned char* pic;  
cudaMallocManaged(&pic, frames * width * width * sizeof(char));  
  
CalcFrame<<<1,frames>>>>(pic,width);  
  
cudaDeviceSynchronize();
```

Parte 2

```
__global__  
void CalcPixel(unsigned char* pic, int width, int frame)  
{  
    int row = blockIdx.x;  
    int col = threadIdx.x;  
    float fx = col - 1024/2;  
    float fy = row - 1024/2;  
    float d = sqrtf( fx * fx + fy * fy );  
    unsigned char color = (unsigned char) (160.0f + 127.0f *  
                                            cos(d/10.0f - frame/7.0f) /  
                                            (d/50.0f + 1.0f));  
  
    pic[frame * width * width + row * width + col] = (unsigned char) color;  
}
```

```
// allocate picture array  
unsigned char* pic;  
cudaMallocManaged(&pic, frames * width * width * sizeof(char));  
  
for(int i = 0; i < frames; i++){  
    int b_size = num_b = width;  
    CalcPixel<<<num_b,b_size>>>(pic,width,i);  
}  
cudaDeviceSynchronize();
```

TESTES EFETUADOS

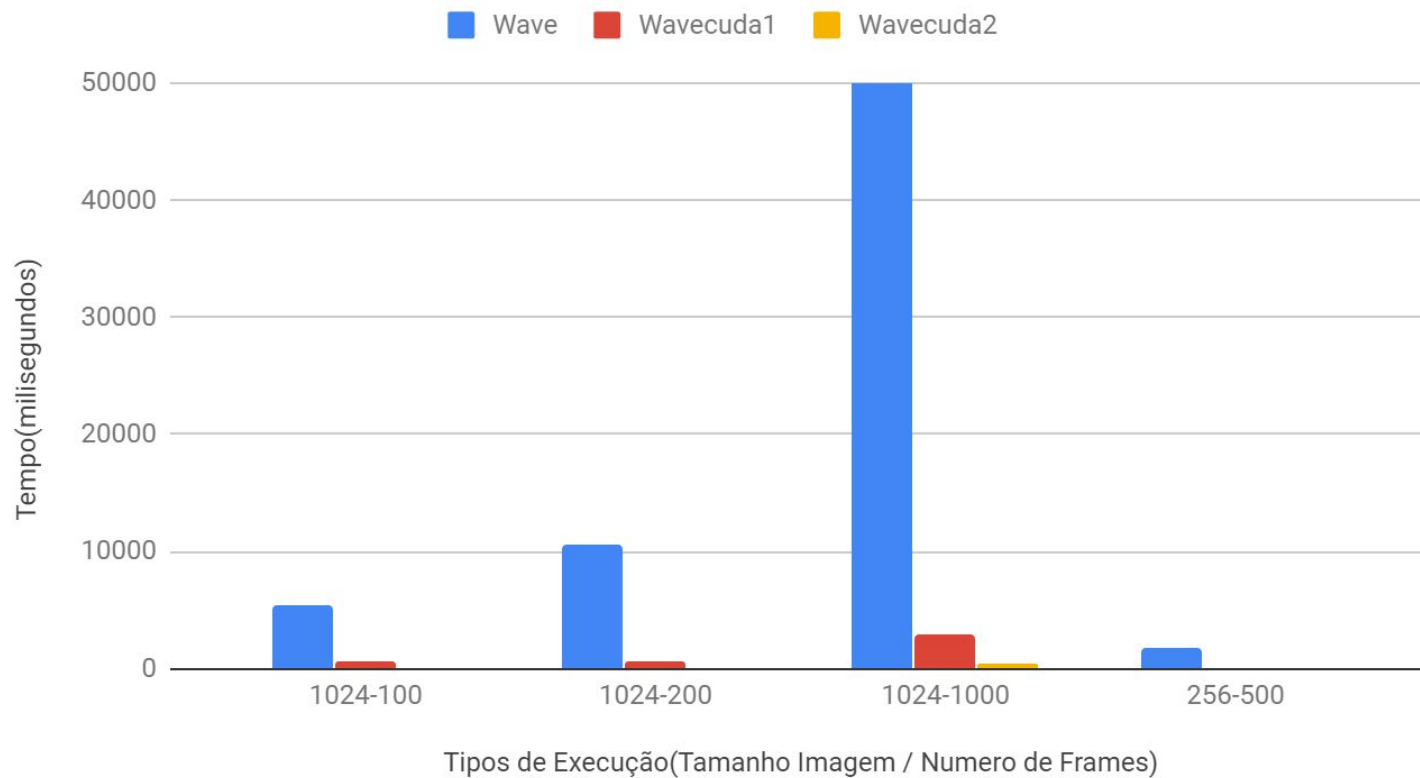
Para cada caso foram efetuados 4 testes na máquina do google Colab:

- Tamanho 1024 x 1024 e 100 Frames
- Tamanho 1024 x 1024 e 200 Frames
- Tamanho 1024 x 1024 e 1000 Frames
- Tamanho 256 x 256 e 500 Frames

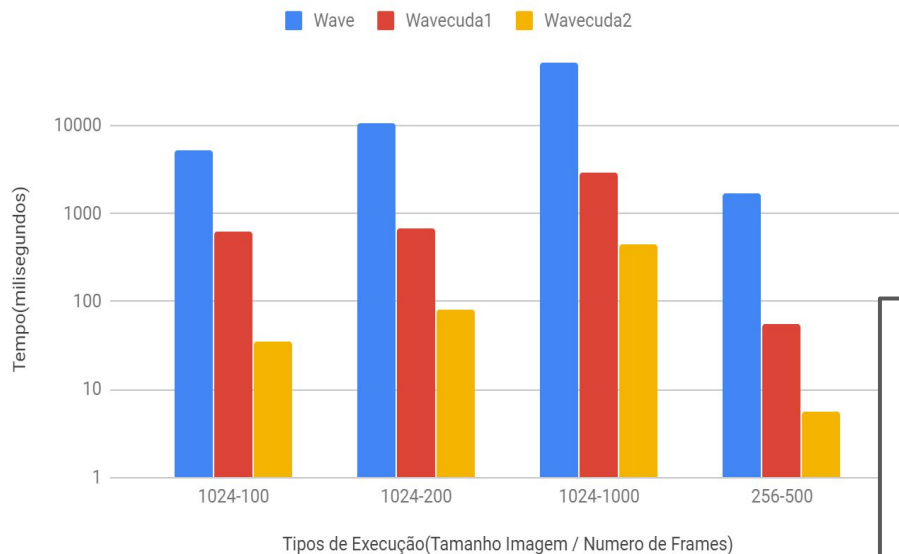
(Nota-se que testes maiores podem ou Passar do número máximo de threads/blocos disponíveis ou Estourar o tamanho máximo do buffer de memória).

1	EXEMPLO	-
2	Execução(Resolucao-Frames)	Tempo(ms)
3	1024-100	5321.0
4	1024-200	10629.8
5	1024-1000	53020.0
6	256-500	1680.5
7	-	-
8	-	-
9	PARTE 1	-
10	Execução(Resolucao-Frames)	Tempo(ms)
11	1024-100	635.0
12	1024-200	666.5
13	1024-1000	2906.9
14	256-500	55.5
15	-	-
16	-	-
17	PARTE 2	-
18	Execução(Resolucao-Frames)	Tempo(ms)
19	1024-100	35.5
20	1024-200	82.3
21	1024-1000	438.6
22	256-500	5.672

Tempos de Execução



Tempos de Execução (Escala Logarítmica)



Escalas Logarítmica e Reduzida para melhor visualização dos resultados

Tempos de Execução (Escala 0-3000)

