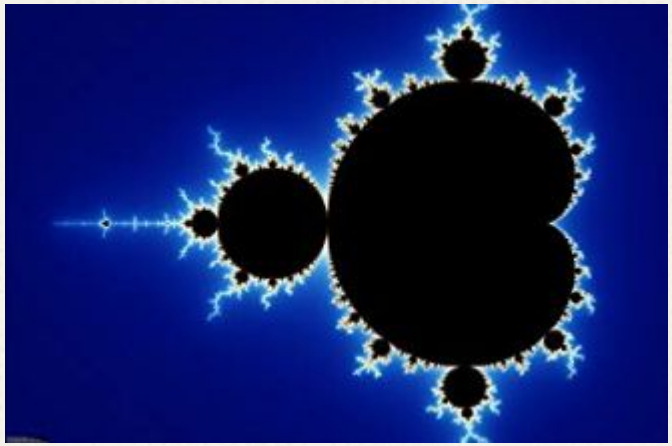
# Programação Paralela

# T4: Geração de Fractais de Mandelbrot em OpenMP

Rafael Vales Bettker

# Fractais de Mandelbrot

O Conjunto de Mandelbrot é um tipo de fractal obtido a partir da fórmula recursiva:



$$Z(0) = 0$$

$$Z(n+1) = Z(n)^2 + c$$

A partir dela pode ser obtida uma representação gráfica que consiste em um cardióide tangenciado por quase-círculos que variam de tamanho e com raio, tendendo a zero. Lógica que se repete para cada um desses quase-círculos infinitamente.

# Estratégias

## Particionamento

A divisão de tarefas entre as threads foram feitas de duas maneiras: entre frames e entre linhas de um frame.

## Escalonamento

Foi testado escalonamento estático e dinâmico para os dois casos de particionamento.

# fractalpar1.cpp

O particionamento é feito nos frames com escalonamento dinâmico. Cada thread fica responsável por um frame do fractal, e assim que termina a execução de um, pega o próximo.

```
#pragma omp parallel for schedule(dynamic) num_threads(threads)
for (int frame = 0; frame < frames; frame++) {
  double delta = Delta * pow(0.98, frame);
  const double xMin = xMid - delta;
  const double yMin = yMid - delta;
  const double dw = 2.0 * delta / width;
  for (int row = 0; row < width; row++) {
    // ...
  }
}
```

# fractalpar2.cpp

O particionamento é feito entre as linhas de cada frame. Os frames são executados em sequência, durante a execução de cada iteração é feita a divisão do trabalho, com cada thread ficando responsável por uma linha diferente do frame em questão.

```
double delta = Delta;
for (int frame = 0; frame < frames; frame++) {
  const double xMin = xMid - delta;
  const double yMin = yMid - delta;
  const double dw = 2.0 * delta / width;
  #pragma omp parallel for schedule(dynamic) num_threads(threads)
  for (int row = 0; row < width; row++) {
    // ...
  }
  delta *= 0.98;
}
```

# Experimentos

Variações de entrada

- Width: 128, 256, 512, 1024 e 2048;
- Frames: 32, 64, 128, 256 e 384;
- Threads: 1, 2, 4, 8, 16 e 32.

Foram realizadas 10 execuções de cada, e então feita a média.

# Resultados

Speedup/eficiência do primeiro caso é maior para tamanho de imagens menores.

| width | frames | threads | segs (1) | efficiency | segs (2) | efficiency |
|---|---|---|---|---|---|---|
| 128 | 32 | 1 | 0.6729 | | 0.6807 | |
| 128 | 32 | 2 | 0.3493 | 96.31% | 0.3681 | 92.45% |
| 128 | 32 | 4 | 0.1812 | 82.85% | 0.1935 | 87.93% |
| 128 | 32 | 8 | 0.1159 | 72.60% | 0.1068 | 79.67% |
| 128 | 32 | 16 | 0.0683 | 61.54% | 0.0604 | 70.49% |
| 128 | 32 | 32 | 0.0396 | 53.10% | 0.0936 | 22.73% |

# Resultados

| width | frames | threads | segs (1) | efficiency | segs (2) | efficiency |
|---|---|---|---|---|---|---|
| 128 | 128 | 1 | 2.7338 | | 2.7310 | |
| 128 | 128 | 2 | 1.3989 | 97.71% | 1.3974 | 97.72% |
| 128 | 128 | 4 | 0.7227 | 94.57% | 0.7518 | 90.82% |
| 128 | 128 | 8 | 0.4190 | 81.55% | 0.4061 | 84.06% |
| 128 | 128 | 16 | 0.2277 | 75.03% | 0.2272 | 75.13% |
| 128 | 128 | 32 | 0.1321 | 64.69% | 0.3587 | 23.79% |
| 128 | 384 | 1 | 9.5169 | | 9.2288 | |
| 128 | 384 | 2 | 4.8321 | 98.48% | 4.7601 | 96.94% |
| 128 | 384 | 4 | 2.5267 | 94.16% | 2.5793 | 89.45% |
| 128 | 384 | 8 | 1.3526 | 87.95% | 1.3456 | 85.73% |
| 128 | 384 | 16 | 0.7263 | 81.89% | 0.7432 | 77.61% |
| 128 | 384 | 32 | 0.3918 | 75.90% | 0.5747 | 50.19% |

# Resultados

Quanto maior o tempo de execução, maior o speedup para valores de threads altos.

| width | frames | threads | segs (1) | efficiency | segs (2) | efficiency |
|---|---|---|---|---|---|---|
| 128 | 32 | 1 | 0.6729 | | 0.6807 | |
| 128 | 32 | 2 | 0.3493 | 96.31% | 0.3681 | 92.45% |
| 128 | 32 | 4 | 0.1812 | 82.85% | 0.1935 | 87.93% |
| 128 | 32 | 8 | 0.1159 | 72.60% | 0.1068 | 79.67% |
| 128 | 32 | 16 | 0.0683 | 61.54% | 0.0604 | 70.49% |
| 128 | 32 | 32 | 0.0396 | 53.10% | 0.0936 | 22.73% |

# Resultados

| width | frames | threads | segs (1) | efficiency | segs (2) | efficiency |
|---|---|---|---|---|---|---|
| 512 | 128 | 1 | 43.7582 | | 42.8347 | |
| 512 | 128 | 2 | 21.9686 | 99.59% | 21.9297 | 97.66% |
| 512 | 128 | 4 | 11.6953 | 93.54% | 11.0744 | 96.70% |
| 512 | 128 | 8 | 6.2843 | 87.04% | 5.9167 | 90.50% |
| 512 | 128 | 16 | 3.3847 | 80.80% | 3.0553 | 87.62% |
| 512 | 128 | 32 | 1.8706 | 73.10% | 1.7612 | 76.00% |
| 2048 | 384 | 1 | 2366.0652 | | 2413.7174 | |
| 2048 | 384 | 2 | 1189.8515 | 99.43% | 1214.8666 | 99.34% |
| 2048 | 384 | 4 | 619.5239 | 95.48% | 624.3399 | 96.65% |
| 2048 | 384 | 8 | 329.6238 | 89.73% | 329.2729 | 91.63% |
| 2048 | 384 | 16 | 166.7988 | 88.66% | 167.3842 | 90.13% |
| 2048 | 384 | 32 | 87.6265 | 84.38% | 87.9374 | 85.78% |

# Resultados

Diferença significativa do tempo de execução apenas percebida com width grandes.

| width | frames | threads | segs (1) | efficiency | segs (2) | efficiency |
|---|---|---|---|---|---|---|
| 2048 | 32 | 1 | 153.4127 | | 163.1400 | |
| 2048 | 32 | 2 | 77.1797 | 99.39% | 80.8667 | 100.87% |
| 2048 | 32 | 4 | 40.6883 | 94.26% | 41.8862 | 97.37% |
| 2048 | 32 | 8 | 21.5039 | 89.18% | 22.4192 | 90.96% |
| 2048 | 32 | 16 | 11.4599 | 83.67% | 11.4918 | 88.73% |
| 2048 | 32 | 32 | 6.4693 | 74.11% | 5.8037 | 87.84% |

# Resultados

| width | frames | threads | segs (1) | efficiency | segs (2) | efficiency |
|---|---|---|---|---|---|---|
| 2048 | 128 | 1 | 644.2625 | | 679.7192 | |
| 2048 | 128 | 2 | 323.2579 | 99.65% | 344.7152 | 98.59% |
| 2048 | 128 | 4 | 171.9104 | 93.69% | 175.6997 | 96.72% |
| 2048 | 128 | 8 | 91.3834 | 88.13% | 94.3166 | 90.08% |
| 2048 | 128 | 16 | 48.6451 | 82.78% | 48.4765 | 87.64% |
| 2048 | 128 | 32 | 25.4944 | 78.97% | 25.3737 | 83.71% |
| 2048 | 384 | 1 | 2366.0652 | | 2413.7174 | |
| 2048 | 384 | 2 | 1189.8515 | 99.43% | 1214.8666 | 99.34% |
| 2048 | 384 | 4 | 619.5239 | 95.48% | 624.3399 | 96.65% |
| 2048 | 384 | 8 | 329.6238 | 89.73% | 329.2729 | 91.63% |
| 2048 | 384 | 16 | 166.7988 | 88.66% | 167.3842 | 90.13% |
| 2048 | 384 | 32 | 87.6265 | 84.38% | 87.9374 | 85.78% |

# Resultados

Escalonamento estático possui speedup/eficiência menor em relação ao dinâmico para grande maioria dos testes.

| width | frames | threads | segs (D) | efficiency | segs (S) | efficiency |
|---|---|---|---|---|---|---|
| 1024 | 384 | 1 | 605.8476 | | 597.3818 | |
| 1024 | 384 | 2 | 307.7439 | 98.43% | 323.4291 | 92.35% |
| 1024 | 384 | 4 | 163.1463 | 92.84% | 175.8729 | 84.92% |
| 1024 | 384 | 8 | 86.0075 | 88.05% | 94.2501 | 79.23% |
| 1024 | 384 | 16 | 44.9966 | 84.15% | 48.5512 | 76.90% |
| 1024 | 384 | 32 | 22.9888 | 82.36% | 25.7994 | 72.36% |

# Referências


- Fractal Foundation. Mandelbrot Magic. https://fractalfoundation.org/OFC/OFC-5-5.html.

- FCUP. Introdução ao OpenMP. https://www.dcc.fc.up.pt/~fds/aulas/PPD/0708/intro_openmp-1x2.pdf.

- CECI. Slurm Quick Start Tutorial. https://support.ceci-hpc.be/doc/_contents/QuickStart/SubmittingJobs/SlurmTutorial.html.