ELC139 - Programação Paralela
Trabalho 4: Geração de Fractais de Mandelbrot em OpenMP

Lucas R. de Araujo[1]

[1]Curso de Ciência da Computação
Universidade Federal de Santa Maria

23 de Abril de 2019

Estudo do código

- Oportunidades de paralelização: 3 laços `for` na função principal:
  - Frames: `for (int frame = 0; frame < frames; frame++);`
  - Linhas: `for (int row = 0; row < width; row++);`
  - Colunas: `for (int col = 0; col < width; col++);`

Estudo do código

Oportunidades de paralelização: 3 laços `for` na função principal:

- **Frames:** `for (int frame = 0; frame < frames; frame++);`
  - Existe **dependência**: variável `delta`.

```
// compute frames
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;
  ...
  delta *= 0.98;
}
```

Figura 1: `delta` apresenta um valor diferente a cada iteração do laço.

Estudo do código

- Oportunidades de paralelização: 3 laços `for` na função principal:
  - Frames: `for (int frame = 0; frame < frames; frame++);`
  - Linhas: `for (int row = 0; row < width; row++);`
    - Não há dependências!
  - Colunas: `for (int col = 0; col < width; col++);`
    - Não há dependências!

Códigos paralelos gerados

- Caso 1: paralelização do laço das linhas:

```
#pragma omp parallel for schedule(dynamic)
for (int row = 0; row < width; row++) {
    const double cy = yMin + row * dw;
    for (int col = 0; col < width; col++) {
        const double cx = xMin + col * dw;
```

- Caso 2: paralelização do laço das colunas:

```
for (int row = 0; row < width; row++) {
    const double cy = yMin + row * dw;
    #pragma omp parallel for schedule(dynamic)
    for (int col = 0; col < width; col++) {
        const double cx = xMin + col * dw;
```

- Caso 3: paralelização dos dois laços aninhados:

```
#pragma omp parallel for schedule(dynamic) collapse(2)
for (int row = 0; row < width; row++) {
    for (int col = 0; col < width; col++) {
        const double cy = yMin + row * dw;
        const double cx = xMin + col * dw;
```

Metodologia

- Condições de execução:
  - Código: 3 versões.
  - Largura: 512 e 1024.
  - *Frames*: 32 e 64.
  - Número de *threads*: 1, 2, 4 e 8.
  - 5 execuções para cada conjunto.
- Ambiente de execução:
  - CDER01 *node*:
    - 2x Intel®Xeon®CPU E5-2620 v3 @ 2.40GHz.
    - RAM: 64 GB.
    - 8 núcleos alocados (12 disponíveis).

Resultados

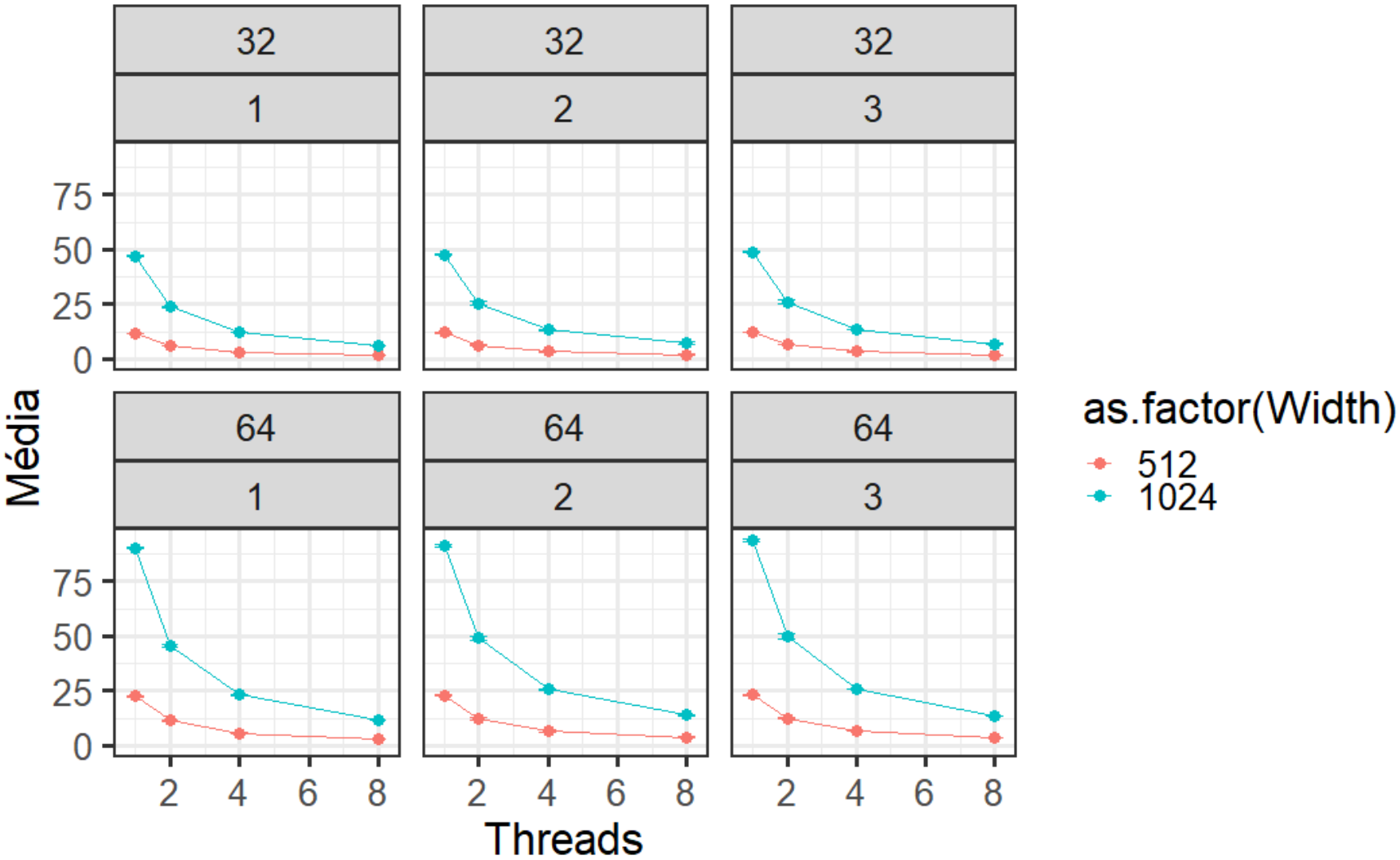

Figura 2: Média dos tempos de execução (segundos).

Resultados (Largura = 512)

Threads	Tempo (s)	*Speedup*	Eficiência
1	11.74(1) 12.21(3)	-	-
2	5.92(1) 6.68(3)	1.98(1) 1.82(3)	99%(1) 91%(3)
4	3.07(1) 3.36(2)	3.82(1) 3.57(2)	96%(1) 89%(2)
8	1.61(1) 2.03(2)	7.27(1) 5.9(2)	91%(1) 74%(2)

Tabela 1: 32 *Frames*.

Threads	Tempo (s)	*Speedup*	Eficiência
1	22.54(1) 23.41(3)	-	-
2	11.44(1) 12.49(3)	1.96(1) 1.86(2)	98%(1) 93%(2)
4	5.81(1) 6.69(2)	3.87(1) 3.42(2)	97%(1) 86%(2)
8	3.03(1) 3.88(2)	7.42(1) 5.91(2)	93%(1) 74%(2)

Tabela 2: 64 *Frames*.

Resultados (Largura = 1024)

Threads	Tempo (s)	*Speedup*	Eficiência
1	47.03(1) 48.81(3)	-	-
2	23.85(1) 25.96(3)	1.97(1) 1.87(2)	99%(1) 94%(2)
4	12.16(1) 13.46(3)	3.86(1) 3.59(2)	97%(1) 90%(2)
8	6.21(1) 7.38(2)	7.56(1) 6.46(2)	95%(1) 81%(2)

Tabela 3: 32 *Frames*.

Threads	Tempo (s)	*Speedup*	Eficiência
1	90.09(1) 93.66(3)	-	-
2	45.57(1) 49.83(3)	1.97(1) 1.86(2)	99%(1) 93%(2)
4	23.16(1) 25.82(3)	3.88(1) 3.55(2)	97%(1) 89%(2)
8	11.80(1) 13.88(2)	7.63(1) 6.57(2)	95%(1) 82%(2)

Tabela 4: 64 *Frames*.

ELC139 - Programação Paralela
Trabalho 4: Geração de Fractais de Mandelbrot em OpenMP

Lucas R. de Araujo[1]

[1]Curso de Ciência da Computação
Universidade Federal de Santa Maria

23 de Abril de 2019