

# EP3: Cálculo Distribuído do Conjunto de Mandelbrot

Pedro Bruel e Alfredo Goldman

MAC 5742-0219 *Introdução à Programação Concorrente, Paralela e Distribuída*

## 1. Introdução

Você se familiarizou com o Conjunto de Mandelbrot<sup>1</sup> no EP1. Seu descobridor foi Benoit Mandelbrot, que trabalhava na IBM durante a década de 1960 e foi um dos primeiros a usar computação gráfica para mostrar como complexidade pode surgir a partir de regras simples. Benoit fez isso gerando e visualizando imagens de geometria fractal.

Um desses fractais foi nomeado *Conjunto de Mandelbrot* pelo matemático Adrien Douady. O Conjunto de Mandelbrot pode ser informalmente definido como o conjunto dos números complexos  $c$  para os quais a função  $f_c(z) = z^2 + c$  não diverge quando é iterada começando em  $z = 0$ . Isto é, a sequência  $f_c(0), f_c(f_c(0)), f_c(f_c(f_c(0))), \dots$  é sempre limitada. A Figura 1 mostra uma região do Conjunto de Mandelbrot conhecida como *Seahorse Valley*.

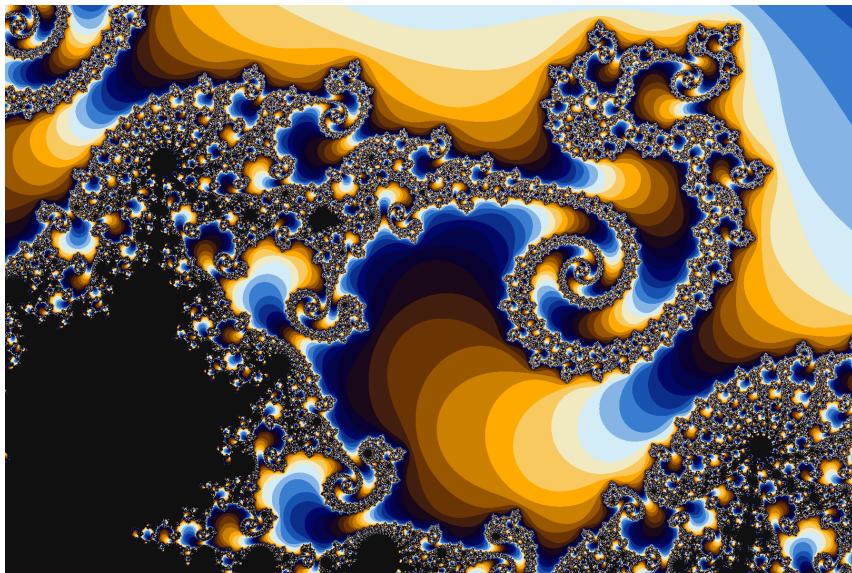


Figura 1: Seahorse Valley

As cores na Figura 1 indicam o número da iteração onde a função  $f_c(z)$  convergiu. A Figura 1 foi gerada usando o código fonte fornecido junto

---

<sup>1</sup>[https://en.wikipedia.org/wiki/Mandelbrot\\_set](https://en.wikipedia.org/wiki/Mandelbrot_set) [Acessado em 30/03/2017]

com este documento. O código na linguagem C e os arquivos em L<sup>A</sup>T<sub>E</sub>X necessários para gerar este documento estão disponíveis no *GitHub*<sup>2</sup>. O resto deste documento descreve as tarefas que você e seu grupo deverão realizar no EP3.

## 2. Tarefas

Você e seu grupo deverão paralelizar e distribuir o código para o cálculo do Conjunto de Mandelbrot usando a biblioteca OpenMPI e, se quiserem, as diretivas de compilador fornecidas pelo OpenMP. Depois, vocês deverão medir o tempo de execução das versões sequencial e distribuída com OpenMPI. Como vocês já estudaram o problema e mediram o tempo de execução para diferentes tamanhos de entrada e números de *threads* no EP1, basta apresentar as medições de desempenho para os melhores parâmetros e implementação que vocês conseguirem.

A implementação necessária para paralelizar o código fornecido usando OpenMPI é relativamente simples. A parte mais trabalhosa do EP3 é a elaboração de um relatório, seguindo o padrão deste documento, que apresente e discuta de forma científica os resultados que vocês obtiverem com as paralelizações e os motivos das otimizações que fizerem.

### 2.1. Programação Distribuída com OpenMPI

Vocês devem distribuir o cálculo do conjunto de Mandelbrot usando as primitivas da biblioteca OpenMPI. Vocês podem modificar à vontade o arquivo `mandelbrot_seq.c`, desde que ele continue produzindo a mesma saída. Opcionalmente vocês podem utilizar OpenMP para paralelizar as porções distribuídas em cada máquina.

### 2.2. Experimentos

Você deverão realizar medições do tempo de execução para apenas um tamanho de entrada  $N \geq 8192$ , onde  $N$  é tamanho em *pixels* do lado da imagem. Vocês podem escolher o tamanho e também apresentarem resultados para mais tamanhos caso achem necessário. Preferencialmente os experimentos deverão ser realizados na Google Compute Engine (GCE), usando os US\$300 “de brinde” que o serviço fornece por um ano. O tipo de instância que vocês devem usar é o `n1-standard`, e vocês devem apresentar medições da sua implementação para conjuntos de 1, 2, 4 e 8 instâncias com 8, 4, 2 e 1 *cores*, respectivamente. Se vocês usaram OpenMP, devem determinar e

---

<sup>2</sup><https://github.com/phrb/MAC5742-0219-EP3> [Acessado em 13/06/2017]

informar qual a melhor quantidade de *threads*, mas não precisam apresentar experimentos.

Vocês devem fazer um número de medições e analisar a variação dos valores obtidos. Sugerimos 10 medições para cada experimento, e também que vocês usem a média e o desvio padrão das 10 medições nos seus gráficos. Vocês podem fazer mais medições, desde que apresentem a média e o desvio padrão. **Não é recomendado** fazer menos de 10 medições.

Vocês não precisam analisar o impacto das porções não paralelizáveis do código sequencial, pois já fizeram isso no EP1. Cada experimento deverá ser repetido nas quatro regiões<sup>3</sup> especificadas na Figura 6. As coordenadas para cada região podem ser obtidas executando no diretório `src`:

```
$ make mandelbrot_seq
gcc -o mandelbrot_seq -std=c11 mandelbrot_seq.c
$ ./mandelbrot_seq
usage: ./mandelbrot_seq c_x_min c_x_max c_y_min c_y_max image_size
examples with image_size = 11500:
    Full Picture:      ./mandelbrot_seq -2.5 1.5 -2.0 2.0 11500
    Seahorse Valley:   ./mandelbrot_seq -0.8 -0.7 0.05 0.15 11500
    Elephant Valley:   ./mandelbrot_seq 0.175 0.375 -0.1 0.1 11500
    Triple Spiral Valley: ./mandelbrot_seq -0.188 -0.012 0.554 0.754 11500
```

O número de iterações para o critério de convergência foi escolhido<sup>4</sup> de forma a produzir uma imagem interessante em diferentes níveis de magnificação, mas ter um tempo de execução razoável para tamanhos grandes de entrada.

### 2.3. Apresentação dos Resultados

Depois de realizar os experimentos vocês deverão elaborar gráficos que evidenciem o comportamento da sua implementação distribuída em função do número de instâncias da GCE. Os gráficos deverão ser claros e legíveis, com eixos nomeados. Deverão apresentar a média e o desvio padrão das 10 execuções para cada cenário experimental.

Recomendamos que vocês usem ferramentas como a biblioteca `matplotlib` da linguagem `Python`. Se fizerem isso vocês conseguirão automatizar a realização dos experimentos e a geração dos gráficos. A automatização dos experimentos e da visualização dos dados gerados é fundamental para pesquisa em Ciência da Computação, pois permite gerar e analisar grandes conjuntos de dados sem muito esforço manual.

---

<sup>3</sup><http://www.nahee.com/Derbyshire/manguide.html> [Acessado em 30/03/2017]

<sup>4</sup><https://goo.gl/WpL9hs> [Acessado em 30/03/2017]

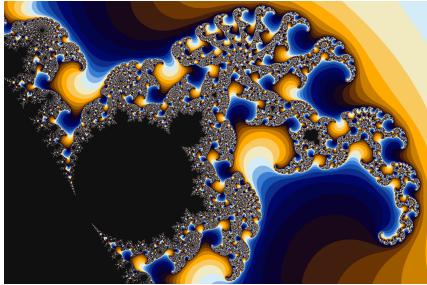


Figura 2: *Elephant Valley*

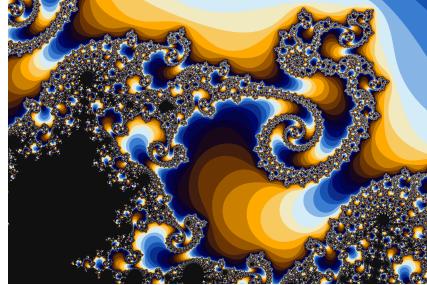


Figura 3: *Seahorse Valley*

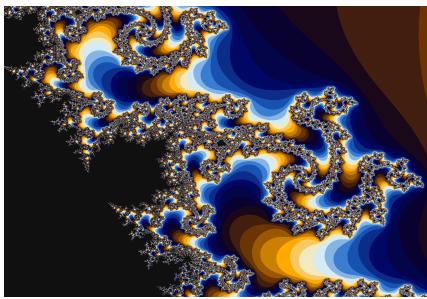


Figura 4: *Triple Spiral Valley*

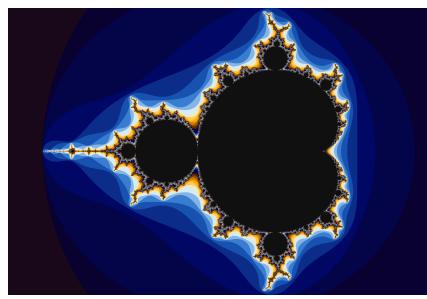


Figura 5: *Full Picture*

Figura 6: Regiões do Conjunto de Mandelbrot

#### 2.4. Entrega no PACA

Vocês deverão entregar no PACA **apenas um relatório e código fonte por grupo**. A entrega deve ser um único arquivo nos formatos `.tar`, `.zip`, ou qualquer formato que o `tar` consiga descompactar. Vocês também podem entregar apenas um `pdf`, desde que ele contenha um `link` para algum repositório aberto do seu código fonte. A entrega deve ser feita **até dia 04/07/17**.

### 3. Tecnologias

Esta seção descreve brevemente algumas tecnologias usadas no EP3. O monitor estará disponível na **Sala 120** para tirar dúvidas do EP3, envie um e-mail para `pedro.bruel@gmail.com` para marcar um horário.

#### 3.1. Google Compute Engine (GCE)

O grupo deve possuir uma conta no Google e se cadastrar na Google Compute Engine. Será necessário fornecer um cartão de crédito, mas não se preocupem, pois a GCE não cobrará nada automaticamente, nem mesmo ao fim do ano gratuito.

Com a conta vocês terão acesso a US\$300 que poderão usar por um ano. Isso é **muito mais do que suficiente** para este EP e pra vocês brincarem à vontade durante um ano.

Vocês devem criar instâncias do tipo **n1-standard** usando a imagem que o monitor vai compartilhar com o e-mail que vocês forneceram no registro do Grupo. Cadastre seu grupo no formulário <sup>5</sup>.

Vocês precisarão configurar IPs internos para as suas instâncias para poderem executar os experimentos. Recomendamos que vocês leiam os tutoriais disponíveis na documentação oficial da GCE <sup>6</sup> para aprender como configurar e utilizar grupos de instâncias.

### 3.2. *Shell scripting & GNU screen*

Para usar a máquina virtual vocês vão precisar usar um emulador do *shell* do Linux, como o **bash**. O arquivo **run\_measurements.sh** contém o necessário para gerar as medições de tempo de execução das três versões do programa, mas vocês vão precisar modificá-lo para medir o impacto das operações de I/O e alocações de memória.

Para deixar o *script* rodando na sua instância da GCE, faça o seguinte:

```
$ screen  
$ ./run_measurements.sh  
<Ctrl+A><D>
```

O comando **screen** lança uma seção da qual você pode se desconectar sem parar a execução de um comando. A sequência **<Ctrl+A>** seguida de **<D>** desconectará você da sessão. Para voltar, basta executar:

```
$ screen -r
```

### 3.3. *Linux perf*

Um dos comandos executados por **run\_measurements.sh**, expandindo as variáveis, é:

```
$ perf stat -r 10 ./mandelbrot_seq -2.5 1.5 -2.0 2.0 512
```

O comando acima executa dez repetições da versão sequencial do programa, calculando a imagem inteira do Conjunto de Mandelbrot com um tamanho de imagem de 512 *pixels*. Modificar os parâmetros desse comando será suficiente para realizar os experimentos do EP3.

---

<sup>5</sup><https://goo.gl/forms/HJomXSieWB0aJqWs1> [Acessado em 13/06/2017]

<sup>6</sup><https://cloud.google.com/compute/docs/configure-ip-addresses#assigninternalip> [Acessado em 13/06/2017]

### *3.4. L<sup>A</sup>T<sub>E</sub>X*

Instalem o L<sup>A</sup>T<sub>E</sub>X na máquina que vão usar para escrever o relatório e usem o arquivo `enunciado_ep3.tex` e o `Makefile` no repositório do EP3 como modelo.

## **4. Critério de Avaliação**

A nota do EP3 vai de **0.0** a **10.0**, e a avaliação será feita da maneira descrita a seguir, se os alunos concordarem.

### *4.1. Apresentação e Análise de Medidas para o Programa Sequencial*

Vale **2.0**, divididos da seguinte maneira:

- Relatório: **2.0**
  - Apresentação e Análise dos Experimentos: **1.8**
  - Clareza do texto e figuras: **0.2**

### *4.2. Apresentação e Análise de Medidas para o Programa Distribuído*

Vale **6.0**, divididos da seguinte maneira:

- Implementação: **4.0**
  - Código compila sem erros e *warnings*: **1.8**
  - Código executa sem erros e produz o resultado correto: **1.8**
  - Boas práticas de programação e clareza do código: **0.4**
- Relatório: **4.0**
  - Apresentação e Análise dos Experimentos: **3.6**
  - Clareza do texto e figuras: **0.4**

O monitor estará disponível na **Sala 120** para tirar dúvidas do EP3, envie um e-mail para `pedro.bruel@gmail.com` para marcar um horário. Bom EP!