

Programação p Sistemas Paralelos e Distribuídos  
Prof.: Fernando W. Cruz  
Laboratório sobre programação OMP, CUDA e OpenCL

### A) Objetivos do laboratório

O objetivo deste laboratório é permitir que o aluno avance seus conhecimentos sobre programação paralela com uso da API OpenMP e das linguagens CUDA e OpenCL em GPUs.

### B) Roteiro do laboratório

Considere o texto a seguir:

A geometria fractal é baseada no princípio de que um objeto geométrico pode ser dividido em partes menores, cada uma delas semelhante ao objeto original. São, portanto, objetos com muitos detalhes, com similaridade recursiva (os detalhes são similares ao objeto original). Um dos fractais interessantes de se observar é o fractal Julia. Um conjunto Julia (*Julia set*) é uma generalização do famoso conjunto Mandelbrot [[https://pt.wikipedia.org/wiki/Conjunto\\_de\\_Mandelbrot](https://pt.wikipedia.org/wiki/Conjunto_de_Mandelbrot)], que está ilustrado na Figura 1.

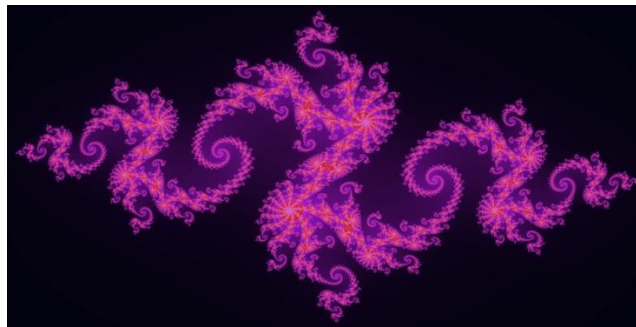


Figura 1 – Fractal Julia

Esse fractal é definido como segue. Dado  $z$  um ponto no plano complexo 2-D, calculamos a série definida como:  $z_0 = z$  e  $z_{n+1} = z_n^2 + c$ , onde  $c = -0,79 + i * 0,15$ , ou seja, um número complexo. Valores diferentes de  $c$  levam a imagens diferentes, e conhecemos muitos valores que produzem imagens “bonitas”. A cor de um pixel correspondente ao ponto  $z$  é determinada com base no número de iterações antes que o módulo de  $z_n$  seja superior a 2 (ou até que um número máximo de iterações seja atingido). O programa `fractal.c` é o código que produz a imagem da Figura 1, mas é possível alterá-lo para criar imagens diferentes. Este programa pode ser compilado com o parâmetro a seguir:

```
$ gcc fractal.c -o fractal -lm
```

Para executá-lo, basta digitar o comando

```
$ ./fractal <N>
```

onde  $N$  é a altura da figura do fractal (ou número de linhas). Esse parâmetro é utilizado para o cálculo da largura ( $2*N$ ) e o cálculo da área do fractal ( $\text{altura} * \text{largura} * 3$ ).

A saída desse programa é um arquivo em formato bmp (Bitmap) que pode ser aberto com qualquer editor de imagens do seu sistema operacional.

Sobre o fractal citado, os alunos devem:

1. Construir as seguintes versões do código `fractal.c`:

- Uma versão OpenMP com o nome `fractalomp.c`, de modo que  $N$  seja dividido pelo número de *threads* disparadas no momento da chamada. Para este código, procure paralelizar o cálculo do fractal e a escrita do arquivo de saída, em função do parâmetro `OMP_NUM_THREADS`. Por exemplo, supondo 4 *threads*, o programa irá calcular o fractal e gerar o arquivo de saída numa proporção de 25% para cada thread.
- Uma versão CUDA com o nome `fractalcuda.cu`, de modo que o cálculo do fractal seja feito com uso de uma das GPUs do cluster chococino
- Uma versão OpenCL com o nome `fractalocl.c`, de modo que o cálculo do fractal seja feito com uso de uma das GPUs do cluster chococino

2. Montar um experimento que permita comparar os códigos acima, de modo a se identificar qual deles apresenta melhor performance (menor tempo de execução), sob as mesmas condições (dimensões do fractal, porções do código paralelizadas, número de *threads*, núcleos, etc.).

**C) Questões de Ordem**

- Este laboratório pode ser feito por grupos de até 3 alunos
- Este laboratório deve ter os artefatos entregues no Moodle (arquivo zipado) e apresentado em data estabelecida pelo professor. A entrega deve ser composta por: (i) relatório do laboratório + slides, e (ii) códigos, instruções de uso e todas as informações necessárias para esclarecimento e uso dos programas entregues.
- Para a versão com GPUs, comparar o resultado encontrado com as versões em OMP e original, para verificar se o programa está executando os cálculos corretamente (número de bytes do arquivo de saída, qualidade, etc.). Uma outra dica é utilizar um editor de imagens pra ver a figura resultante produzida por cada código
- Os alunos podem realizar o experimento em qualquer plataforma, mas devem considerar que a correção do experimento será feita no cluster chococino (códigos que não rodarem neste cluster não serão validados).
- O relatório deve conter o seguinte:
  - i) Identificação da disciplina/turma, do grupo (matrícula e nome) e o nome do laboratório
  - ii) Listagem dos códigos com comentários e indicações sobre qual parte foi paralelizada, instruções de execução e comentários sobre dificuldades e soluções encontradas em cada uma das versões do `fractal.c`
  - iii) Descrição do experimento – Apresentar os cenários de teste que foram feitos (número de execuções realizadas, resultados encontrados, etc.), considerando parâmetros idênticos para cada um dos códigos. O resultado dos testes deve ser apresentado na forma de uma tabela comparativa, apresentando os de tempos de execução e melhoria de performance em relação à versão sequencial. Obs.: Sugere-se utilizar funções da linguagem que consigam calcular os tempos de execução para melhorar a qualidade da resposta.
  - iv) Conclusão – Gerar um texto conclusivo sobre o experimento, indicando, por exemplo, (i) qual o percentual de ganho de uma solução em relação a outra, (ii) qual das GPUs do cluster chococino é mais eficiente para executar essa aplicação e outros achados que julgar pertinentes.