

Geração de Fratais com MPI

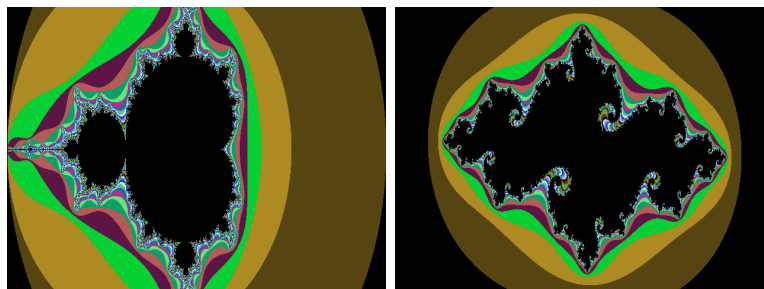


Figura D.1: Left: Julia fractal; Right: Mandelbrot fractal

D.1 Adaptação a MPI

Este trabalho consiste na adaptação do trabalho anterior tirando agora partido do suporte MPI. Por isso deverá começar por rever o código fornecido, bem como analisar como este poderia ser distribuído por uma estrutura de nós que não partilham zonas de memória entre si.

D.2 Trabalho a realizar

Parte I

Exercício D.1

Parta do código fornecido para geração de imagens de fractais, quer do fractal Julia, quer do fractal Mandelbrot. Compile-o e execute-o. Verifique como os tempos de execução se podem tornar extremamente longos quando se geram imagens de resolução considerável. Veja o exemplo seguinte.

```
$ make
$ ./fractal 10000 10000
```

Para os testes iniciais, poderá usar uma resolução de imagem baixa (HD, por exemplo), mas para os testes finais e a escrita do relatório do trabalho deverá usar uma resolução de 4K (3840x2160), tendo o cuidado de adaptar o valor das iterações necessárias para o detalhe atingir o pixel.

Exercício D.2

Repare que a geração do fractal é evolutiva, sendo o número de iterações importante para criar uma imagem cada vez mais detalhada (cujo detalhe, e portanto o número de iterações necessárias, são função da resolução da imagem).

Crie uma versão modificada do código para gravar uma imagem (formato pgm) por iteração, no formato `fractal_iter.pgm` (ex: `julia_35.pgm`), mostrando a evolução da construção do fractal "ao longo do tempo". Para tirar partido da capacidade de cálculo do conjunto do cluster, sugere-se aqui que distribua a criação das imagens desta sequência pelos nós disponíveis. Embora não seja o mais habitual neste tipo de ambientes, cada nó deverá gravar as imagens diretamente desde que mantenham os números de sequência. No entanto a distribuição do processamento por cada nó deverá ser efetuada da seguinte forma: O nó master envia a cada nó adicional o número da iterações da imagem que este deve gerar e apenas quando este completa e o confirma recebe novo pedido de geração de imagem ou então a indicação de que não há mais trabalho a realizar, terminando a execução de todo o cluster.

No final, converta o conjunto de imagens num vídeo de formato mp4, usando a aplicação **ffmpeg** da seguinte forma:

```
$ ffmpeg -framerate 25 -pattern_type glob -i 'julia_*.pgm'
-c:v libx264 -r 30 -pix_fmt yuv420p out_julia.mp4
```

Note que para a geração do vídeo deverá ter instalado os pacotes ImageMagic e ffmpeg.

Visualize o vídeo gerado e repare na evolução do fractal. Faça a medição exata do tempo de geração das várias imagens do fractal. Compare os tempos da geração desta sequência de imagens com os obtidos no trabalho 3.

Parte II

Exercício D.3

Partindo da matriz da imagem fractal gerada vamos simular a difusão das cores ao longo do tempo. Assim teremos de ir sucessivamente gerando novas imagens em que a imagem I_{k+1} é gerada a partir da I_k por difusão. Assim para um pixel de coordenadas i,j , o seu valor é dado por

$$\begin{aligned} I_{k+1}(i,j) = & (1 - \alpha)I_k(i,j) + \alpha \frac{1}{8} [I_k(i-1,j-1) + I_k(i-1,j) + \\ & + I_k(i-1,j+1) + I_k(i,j-1) + I_k(i,j+1) + \\ & + I_k(i+1,j-1) + I_k(i+1,j) + I_k(i+1,j+1)], \end{aligned}$$

onde α é o parâmetro que controla a difusão, pesando a contribuição que o píxel na mesma posição tem versus os 8 vizinhos. Os valores extremos levam a que não haja difusão ($\alpha = 0$) ou a difusão seja muito rápida pelos vizinhos ($\alpha = 1$).

No final de ser gerada cada nova imagem, esta deve ser guardada num ficheiro cujo nome será da forma "output%04d.jpg" (ver função `sprintf`), onde %04d será substituído pelo número de ordem do ficheiro.

Introduza o código para efetuar a difusão na zona indicada da função `diffusao(...)` tirando partido da distribuição do trabalho pelos nós disponíveis para acelerar o processo. Note que para ativar a chamada da função de difusão deverá passar dois parâmetros adicionais que são o número de épocas (iterações) de difusão e o fator alfa que deverá estar entre 0 e 1.

Muito importante: como neste processo de difusão cada imagem da sequência depende da gerada na iteração anterior, todo o processo deve ser comandado pelo nó master. Assim este nó master deve distribuir a informação pelos nós e recolher as sub-imagens geradas escrevendo-as em ficheiro depois de as compor numa única. Por exemplo se tivermos 4 nós e cada um processar 1/4 de imagem os resultados deverão ser reagrupados nos respetivos lugares da imagem final.

```
$ make
$ ./fractal 3840 2160 100 0.5
```

40%

Exercício D.4

Faça uma adaptação do problema da difusão tirando partido da possibilidade de explorar simultaneamente as capacidades de OpenMP e MPI.

Bonus: 20%

Exercício D.5

Elabore um breve relatório onde descreverá as abordagens e compare os tempos de execução com e sem MPI. **Este relatório deverá ser submetido juntamente com os ficheiros fonte num arquivo do tipo ZIP.**