MAC 0219/5742 – Introdução à Computação Concorrente, Paralela e Distribuída

Prof. Dr. Alfredo Goldman

EP3 - Mandelbrot distribuido `versão 1.0`

Monitores: Giuliano Belinassi e Matheus Tavares

1 Introdução

Você implementou no EP2 um Mandelbrot muito eficiente usando GPU. O objetivo agora é fazê-lo ainda mais eficiente através de computação distribuída.

2 Problema

Além de muito bem paralelizavel em memória compartilhada, o problema do cálculo do Conjunto de Mandelbrot também pode ser calculado de maneira distribuida facilmente, já que a quantidade de dados a ser transmitida pela rede é baixa.

Sendo assim, vocês deverão melhorar a implementação do EP2 utilizando a biblioteca `MPI` para distribuir o trabalho entre diversas máquinas[1][2].

3 Especificação do Programa

Seu programa deve ser capaz de atender os seguintes requisitos:

1. Gerar uma imagem em formato `PNG` de acordo com o nome do arquivo de saída especificado no `ARGV`. Recomendamos o uso da `libpng` para isso.

2. Utilizar a biblioteca `OpenMPI` para comunicação distribuida.

3. Aceitar um parâmetro no `ARGV` especificando se o cálculo do Conjunto de Mandelbrot será efetuado na CPU ou na GPU.

4. Conter um `Makefile` que gera o binário `dmbrot`. Este binário deve aceitar como argumento a seguinte linha de comando:

   ```
   dmbrot <C0_REAL> <C0_IMAG> <C1_REAL> <C1_IMAG> <W> <H>
                          <CPU/GPU> <SAIDA>
   ```

   onde:

   - `C0_REAL` é a parte real de c_0.

[1] Felizmente, vocês deverão apenas se preoucupar em executar o EP em uma única máquina.

[2] `https://en.wikipedia.org/wiki/Distributed_computing`

- `C0_IMAG` é a parte imaginária de c_0.
- `C1_REAL` é a parte real de c_1.
- `C1_IMAG` é a parte imaginária de c_1.
- `W` é a largura em pixels da imagem a ser gerada.
- `H` é a altura em pixels da imagem a ser gerada.
- `CPU/GPU` é `cpu` se especificado para executar na CPU, e `gpu` se especificado pra executar na GPU.
- `SAIDA` é o caminho para o arquivo de saída.

Novamente, você não precisa sanitizar os parâmetros de entrada.

4 Dicas

1. Vocês podem executar um programa MPI através do programa `mpiexec -np <NUM_PROCS>`.
2. Veja como as funções `MPI_Send`, `MPI_Recv` funcionam.
3. Utilize uma estratégia de divisão e conquista para distribuir o trabalho de modo que sua implementação do EP2 seja reaproveitada. Será que fazer blocagem na imagem ajuda?
4. Cuidado ao passar uma `struct` como mensagem. A diferença na ordem dos bytes pode corromper os dados se a devida cautela não for tomada. Leia sobre *Big-Endian* e *Little-Endian*.

5 Entrega

Deverá ser entregue um pacote no sistema PACA com uma pasta com o nome e o sobrenome do estudante que o submeteu no seguinte formato: `nome_sobrenome.zip`. Se o EP for feito em trio, o formato deve ser:

`nome1_sobrenome1.nome2_sobrenome2.nome3_sobrenome3.zip`

Somente um estudante do time submeterá a tarefa. Essa pasta deve ser comprimida em formato `ZIP` e deve conter dois itens:

- Os códigos fonte do programa, em conjunto com um `Makefile` que o compila, gerando o executável especificado.
- Um arquivo `.txt` ou `.pdf` com o nome dos integrantes, e uma breve explicação sobre a sua solução e desafios encontrados. Imagens também podem ser inseridas no pacote. Relatórios em `.doc`, `.docx` ou `odt` **não** serão aceitos.

Em caso de dúvidas, use o fórum de discussão do Paca. A data de entrega deste Exercício Programa é até às **16:00h do dia 20 de Junho**. Não se esqueça que o professor irá sortear um aluno para explicar sua implementação do EP neste mesmo dia. Não é necessário preparar slides, apenas falar brevemente sobre o código e as decisões de projeto tomadas.

Boa Sorte!