

Bacharelado em Ciência da Computação - DCC/IM-UFRJ
Programação Paralela e Distribuída
Prof. Gabriel P. Silva
2º Trabalho – 18/02/2021

- 1) Elabore um programa utilizando diretivas do OpenMP que calcule o total de números primos entre 2 e N usando o método do crivo de Eratóstenes, ou seja, marcando em um vetor todos os múltiplos de 2, todos os múltiplos de 3, todos os múltiplos de 5, etc. Execute o código com 1, 8 e 16 threads e um valor de N que resulte em um tempo de execução sequencial da ordem de 300 segundos para uma única thread. Avalie o tempo de execução, speed-up e eficiência obtidos. Elabore um relatório com o código fonte, resultados e comentários. **(2,5 pontos)**.
- 2) Neste trabalho você deve paralelizar o programa **mandelbrot**, que calcula um fractal de mandelbrot, utilizando rotinas do OpenMP. Ao paralelizar o código do **mandelbrot**, você aumentará sua compreensão dos modelos de programação, ganhará experiência no desenvolvimento de uma aplicação MPI. Para isso observe os seguintes passos: **(2,5 pontos)**.
 - Você necessitará de um arquivo começar o trabalho: a versão sequencial do mandelbrot, que pode ser obtida no AVA.
 - Compile, verifique o seu funcionamento e familiarize-se com o modo com que o programa sequencial funciona.
 - Experimente modificar o código para imprimir o **mandelbrot** colorido, aumente o tamanho da imagem para obter um tempo significativo de computação.
 - Paralelize o programa. Para conseguir isto, imagine que cada thread irá calcular faixas (horizontais ou verticais) da imagem.
 - Note que apenas uma tarefa deverá escrever o resultado final da imagem para exibição posterior para o usuário.
 - Compare o desempenho rodando em 1, 8 e 16 threads.
 - Apresente um relatório com código fonte, resultados e comentários sobre todo esse processo.
- 3) Neste trabalho você deve paralelizar o programa mandelbrot, que calcula um fractal de mandelbrot, utilizando rotinas do OpenACC. Ao paralelizar o código do mandelbrot, você aumentará sua compreensão dos modelos de programação, ganhará experiência no desenvolvimento de uma aplicação MPI. Para isso observe os seguintes passos: **(2,5 pontos)**.
 - Organize as matrizes de forma a otimizar o acesso à cache e à memória.
 - Apresente um relatório com código fonte, resultados e comentários sobre todo esse processo.
 - Compare o desempenho com a versão sequencial e com OpenMP. Avalie a movimentação de dados realizada na versão paralela e verifique se há otimizações possíveis. Apresente um relatório com código fonte, resultados e comentários sobre todo esse processo.
- 4) Implemente uma versão paralela em OpenMP da simulação 2-D de transferência de calor em uma placa de metal NxN, utilizando o método Gauss-Seidel SOR (*successive over-relaxation*). Este método funciona realizando M iterações, nas quais é calculada a temperatura de cada ponto de uma placa de metal como a média de cada um dos seus vizinhos. A versão sequencial pode ser obtida no AVA. **(2,5 pontos)**

Dados para o problema:

- Número de iterações: 4098
- Tamanho da placa de metal: 1022
- A temperatura inicial da placa é 20°C
- A temperatura da fonte de calor localizada no ponto 800x800 (meio da placa) da placa é 100°C
- As bordas estão sempre com a temperatura igual a 20°C