

Bacharelado em Ciência da Computação - DCC/IM-UFRJ
Programação Paralela e Distribuída
Prof. Gabriel P. Silva
3º Trabalho – 2/12/2008

- 1) O segundo trabalho consiste em paralelizar o algoritmo de ordenação chamado RankSort ou Enumeration Sort:
- Considere n números **todos** distintos.
 - Para cada número determinamos o seu rank, isto é, a quantidade de números que são menores que ele. Esse rank então determina a posição do número na sequência ordenada.
 - Um algoritmo seqüencial implementando esse método pode ser assim:
 - i. Suponha n números num vetor $a[0]; a[1]; \dots; a[n - 1]$.
 - ii. Primeiro $a[0]$ é comparado com todos para determinar a quantidade de números menores que ele.
 - iii. Digamos que esta quantidade seja x , então $a[0]$ é armazenado no vetor $b[x]$. Fazemos então o mesmo com $a[1]$ e depois sucessivamente com todos os outros.
 - iv. O vetor b assim obtido será a resposta.
 - v. O algoritmo sequencial é $O(n^2)$.
 - Elabore um programa paralelo em OpenMP usando esse método. Justifique o aumento de desempenho esperado com esta versão executando com múltiplos processadores.
 - Execute o código com 4 processadores e avalie o speed-up obtido.
 - Apresente um relatório com código fonte, resultados e comentários sobre todo esse processo.
- 2) 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 4 processadores e avalie o speed-up obtido. Elabore um relatório com o código fonte, resultados e comentários.
- 3) Neste trabalho você deve paralelizar o programa **mandelbrot**, que calcula um fractal de mandelbrot, utilizando rotinas do OpenMP. Para isso observe os seguintes passos:
- Você necessitará de um arquivo começar o trabalho: a versão sequencial do mandelbrot, que pode ser obtida no endereço:
<http://www.dcc.ufrj.br/~gabriel/progpar/mandelbrot.zip>
 - Paralelize o programa. Para conseguir isto, imagine que cada thread irá calcular uma faixa (horizontal ou vertical) da imagem.
 - Note que apenas uma thread deverá mostrar o resultado final para o usuário.
 - Execute o código com 4 processadores e avalie o speed-up obtido.
 - Apresente um relatório com código fonte, resultados e comentários sobre todo esse processo.