

Bacharelado em Ciência da Computação - DCC/IM-UFRJ
Programação Paralela e Distribuída
Prof. Gabriel P. Silva
1º Trabalho – 12/08/2006

1) Neste trabalho você deve paralelizar o programa **calpi**, que calcula o valor de π (3.14159265358979323846264 ...) usando uma aproximação da integral. É uma boa aplicação para se paralelizar porque a divisão do trabalho entre as tarefas é muito simples, e nenhum padrão complicado de comunicação é usado. Ao paralelizar o código do **calpi**, você aumentará sua compreensão dos modelos de programação, ganhará experiência no desenvolvimento de uma aplicação PVM e vai fazer o uso de muitas chamadas básicas do PVM. Para isso observe os seguintes passos:

- Você necessitará de um arquivo começar o trabalho: a versão sequencial do karp, que pode ser obtida no endereço:
<http://equipe.nce.ufrj.br/gabriel/progpar/calpi.zip>
- Compile, verifique o seu funcionamento e familiarize-se com o modo com que o programa seqüencial funciona.
- Paralelize o programa. Para chegar a este ponto, entretanto, você necessitará fazer uma quantidade substancial de mudanças. Isto inclui começar tarefas paralelas e distribuir a informação da configuração.
- Note que apenas uma tarefa deverá pedir dados para o usuário. Você pode substituir pela leitura de um arquivo.
- Use rotinas de comunicação coletivas sempre que possível.
- Compare o desempenho da versão seqüencial com a versão paralela, com valores de N para 100, 1000 e 10.000. Utilize 4, 8 e 16 processadores.
- Apresente um relatório com código fonte, resultados e comentários sobre todo esse processo.

2) 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 seqüencial é $O(n^2)$.
- Projete um programa paralelo em PVM usando esse método.
- Utilize rotinas de comunicação global sempre que possível.
- Compare o desempenho do programa em paralelo com o programa seqüencial, para $n=100, 1000$ e 10.000 . Utilize 4, 8 e 16 processadores.
- Apresente um relatório com código fonte, resultados e comentários sobre todo esse processo.

Se você tiver alguma dificuldade, envie por favor um email para o endereço **gabriel pt silva at ufrj pt br** explicando as dificuldades que você está tendo. Há muitas soluções "corretas" para este trabalho, porque os programadores podem escolher uma posição diferente das chamadas, nomes diferentes para as variáveis, etc. Logo, soluções idênticas não serão toleradas.

- Boa sorte!