CAP-334 / 1997
Quinta Lista de Exercícios
Data de Entrega: 27/11/97, 10:00hs

Objetivo: O objetivo desta lista de exercícios é permitir a aplicação prática, através de exercícios de programação, dos tópicos vistos em aula sobre sistemas paralelos com memória compartilhada.

Observações:

- Todos os programas desta lista devem ser executados, de forma remota, na estação `turing.nucleo.inpe.br`, uma Sparcstation de quatro processadores do laboratório da CAP. Os códigos fonte necessários estão disponíveis na rede do laboratório, e podem ser acessados no diretório ~celso/. Os programas devem ser *compilados* na máquina `visap.nucleo.inpe.br`. Todas as máquinas relacionadas têm acesso ao mesmo sistema de arquivos onde estão os arquivos de usuários.
- Conforme visto em sala de aula, em alguns dos exercícios é necessário apenas criar ou alterar o código do arquivo auxiliar, sem que se tenha que fazer mudanças no código fonte do programa principal (`geral.c`).
- Quando for necessário medir o tempo de uma execução, é conveniente executar o respectivo programa várias vezes, e tomar a média dos diversos tempos observados.
- Para os exercícios (3) e (4), além de entregar uma cópia impressa com o código fonte dos programas solicitados, cada aluno deve também submeter uma cópia do código fonte via e-mail. Enviar cada programa separadamente, com o campo "Subject:" contendo a sigla CAP-334, para o endereço `celso@dpi.inpe.br`.

1) [15 pontos] Considere o programa de multiplicação de dois vetores elemento-a-elemento, implementado pela função `prod_vet.c`.:

   (a) Compile este programa (junto com `geral.c`), e execute-o com 1, 2, 3 e 4 processadores, respectivamente. Apresente os tempos de execução observados em cada caso.

   (b) Calcule o speedup para cada caso em que P>1; plote o speedup para $1 \leq P \leq 4$.

2) [20 pontos] Dentro do programa `increm.c`, cada processador incrementa um certo número de vezes a variável compartilhada `s`, de tal forma que o número total de incrementos a `s` é constante e independente do número de processadores utilizados.

   (a) Compile este programa e execute-o com 1, 2, 3 e 4 processadores, respectivamente. Apresente o valor do speedup em cada caso em que P>1, bem como o valor final de `s` computado durante cada execução. Justifique os valores obtidos para `s`.

(b) Considere agora uma versão modificada deste programa, representada no arquivo `increm_2.c`. Compile este novo programa e execute-o com 1, 2, 3 e 4 processadores. Qual o valor obtido para `s` em cada caso? Justifique tais valores.

3) [40 pontos] Utilizando construções similares àquelas dos dois primeiros exercícios, você deve criar um novo programa que implemente o cálculo do produto escalar entre os dois vetores **B** e **C** do primeiro exercício, imprimindo ao fim do processamento o valor obtido para o produto escalar.

(a) Apresente o código fonte do novo programa, bem como o valor obtido para `s` em execuções utilizando 1, 2, 3 e 4 processadores.
(b) Apresente o speedup obtido em cada uma das execuções do ítem anterior.

4) [25 pontos] Considere o arquivo `barreira.c`, o qual contém uma função em que cada thread executa duas tarefas (A e B). Após terminar uma tarefa, cada thread imprime uma mensagem informando do término.

(a) Compile este programa, execute-o com 2, 3 e 4 processadores e apresente os dados impressos durante a execução, em cada caso.
(b) Altere o código fonte deste programa, de modo a implementar uma barreira entre a execução das duas tarefas A e B. Sua alteração deve ser tal que, durante a execução, todos os threads terminem a tarefa A antes de qualquer um deles iniciar a tarefa B, de acordo com o princípio de funcionamento de uma barreira. Apresente os dados impressos durante a execução deste novo programa para 2, 3 e 4 processadores.