UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
LICENCIATURA EM CIÊNCIA DA COMPUTAÇÃO
06202 - Programação Paralela e Distribuída
PROF. JONES ALBUQUERQUE

Lista de Exercícios

Questão 1.

Discuta, em poucas linhas (+/- 10), vantagens e desvantagens em se utilizar paralelismo de dados e de controle.

Questão 2.

Dada uma tarefa que pode ser dividida em m subtarefas, cada uma requerendo 1 unidade de tempo, quanto tempo será necessário para que um pipeline de m-estágios processe n tarefas?

Questão 3.

Prove que se $\frac{1}{k}$'avos do tempo gasto executando um algoritmo é de operações seqüenciais, então o limite superior para o speedup alcançável executando o algoritmo em processadores paralelos é k. (Dica: use o Teorema de L'Hopital - cálculo)

Questão 4.

Por que parece não ser possível resolver um problema NP-Completo num PRAM em tempo polilogarítmico, utilizando um número polinomial de processadores?

Questão 5.

Considere o seguinte algoritmo seqüencial para multiplicação de matrizes

$$A_{l \times m} \ * \ B_{m \times n} \ = \ C_{l \times n},$$

```
MATRIX MULTIPLICATION
Global variables: a[0 ... (l − 1)][0 ... (m − 1)] {Matrices to be multiplied}
                  b[0 ... (m − 1)][0 ... (n − 1)]
                  c[0 ... (l − 1)][0 ... (n − 1)]  {Product matrix}
                  t                                {Acumulates dot product}
                  i, j, k
  begin
    for i ← 0 to l − 1 do
      for j ← 0 to n − 1 do
        t ← 0
        for k ← 0 to m − 1 do
          t ← t + a[i][k] * b[k][j]
        endfor
        c[i][j] ← t
      endfor
    endfor
  end
```

5.1

Qual a ordem de complexidade deste algoritmo (em número de operações) para multiplicar duas matrizes $n \times n$?

5.2

Assuma um algoritmo paralelo para multiplicação de matrizes com complexidade de tempo

$$\theta(\frac{n^3}{p} \ + \ p),$$

onde p é o número de processadores. Assim, para $p = n^3$ o algoritmo possui complexidade de tempo $\theta(n^3)$. Poderíamos afirmar que este algoritmo é ótimo? Por quê?

5.3

Conseguiríamos implementá-lo com um número menor de processadores? Discuta utilizando o Teorema de Brent.

Cola Oficial

Lei de Amdahl (Amdahl 1967). Seja f a fração de operações numa computação que deve ser executada seqüencialmente, onde $0 \leq f \leq 1$. O Speedup máximo (S) alcançado por um computador paralelo com p processadores executando a computação é

$$S \leq \frac{1}{f + \frac{(1-f)}{p}}.$$

Teorema de Brent. Dado A, um algoritmo paralelo com complexidade de tempo t, se o algoritmo paralelo A realiza m operações computacionais, então p processadores conseguem executar o algoritmo A num tempo

$$t + \frac{(m-t)}{p}.$$

Definição. A expressão $T(n)^{O(1)}$ representa as funções polinomiais de $T(n)$.

Definição. O conjunto $(log\ n)^{O(1)}$ é chamado conjunto de funções **polilogarítmicas**.

Teorema. (Tese da Computação Paralela, von zur Gathen 1986). A classe dos problemas resolvíveis em tempo $T(n)^{O(1)}$ por um PRAM é igual à classe dos problemas resolvíveis em espaço $T(n)^{O(1)}$ por um RAM, se $T(n) \geq log\ n$.

Teorema. Se o número de processadores num PRAM é restrito a alguma função polinomial no tamanho da entrada, então os problemas resolvíveis em tempo polinomial paralelo é a classe $\mathcal{P}$, isto é, o conjunto dos problemas resolvíveis em tempo polinomial numa máquina seqüencial.

Definição. Um algoritmo paralelo tem **complexidade de tempo polilogarítmica** se sua complexidade é uma função polilogarítmica do tamanho do problema.

Definição. $\mathcal{NC}$ é a classe dos problemas resolvíveis em um PRAM em tempo polilogarítmico utilizando um número de processadores que é uma função polinomial do tamanho da entrada.

Definição. (Gibson e Rytter, 1988). Um problema $\mathcal{L} \in \mathcal{P}$ é **$\mathcal{P}$-Completo** se qualquer dos outros problemas em **$\mathcal{P}$-Completo** pode ser transformado para $\mathcal{L}$ em tempo paralelo polilogarítmico utilizando um PRAM com um número polinomial de processadores.