

Questões Teóricas

1 - Qual a diferença entre divisão de tarefas e divisão de dados? Dos três modelos de programação que vimos, como cada um se comporta com os dois tipos de divisão.

Divisão de dados = Uma técnica de programação que divide uma grande quantidade de dados em partes menores que podem ser operadas em paralelo.

Resposta do plentz:

Divisão de dados:

Divide os dados entre os cores, onde geralmente estes aplicam o mesmo algoritmo em cima dos dados;

Divisão de tarefas:

Divide um problema maior em um menor que é resolvido pelos cores, não necessariamente essas tarefas são o mesmo problema.

O modelo de programação paralela pode ser considerado um modelo de divisão de dados;

O modelo de programação distribuída pode ser considerado um modelo de divisão de tarefas;

O modelo heterogêneo pode ser considerado um modelo de divisão de dados e tarefas, dependendo da implementação necessária para a solução do problema.

2 - O que é o método de redução? Para que serve?

Este método aplica repetidamente o mesmo operador de redução a uma sequência de operandos a fim de obter um único resultado. Todos os resultados intermediários da operação devem ser armazenados na mesma variável: a variável de redução.

exemplo: $op = '+'$, cada thread possui uma cópia local da variável, a thread efetuará a soma local em sua cópia e ao sair da seção paralela, as somas locais serão automaticamente adicionadas na variável.

3 - No livro "An Introduction to Parallel Programming" no capítulo 2 são discutidos vários exemplos de configurações de sistemas paralelos e suas vantagens e desvantagens. É discutida a principal dificuldade de hardware que leva a um limitador ao sistema de memória compartilhada. Diga qual é esse limitador e justifique

O controle de acesso ao barramento. À medida que o número de dispositivos conectados ao barramento aumenta, a disputa pelo uso do barramento aumenta e o desempenho diminui.

4 - Qual é a lei de Amdah'l? O que ela nos diz?

Lei de Amdahl diz respeito ao ganho de performance máximo que um programa pode ter, se uma parte sua for paralelizada.

Resposta do Plentz:

A lei de Amdahl diz respeito ao speedup teórico máximo que um programa parcialmente paralelo pode ter.

Exemplo: Um programa que possui 80% do seu código paralelizado e 20% sequencial pode no máximo ter um speedup de 5 vezes ($1/(1-0.8)$)

5 - Como a sequência de código a seguir pode ser alterada para expor mais paralelismo, mas ainda alcançar o mesmo resultado final (ou seja, no final: x, a, b e c têm o mesmo valor que o código sequencial)? Denote quantas threads e o que cada uma faria.

```
x++;  
a = x + 2;  
b = a + 3;  
c++;
```

```
++x ;           // feito antes de tudo  
a = x + 2;      // cria um thread que executa essa linha, thread #0  
b = a + 3;      // thread #0  
++c;           // cria um thread que executa essa linha, thread #1
```

O código pode ser executado em paralelo utilizando duas threads, mais do que isso precisa de sincronização. Embora para tarefas pequenas como essas a criação de threads pode trazer um overhead maior que o ganho pelo paralelismo.

6 - Diga e justifique um problema que é melhor resolvido usando memória compartilhada em detrimento de memória distribuída e programação heterogênea.

Memória compartilhada:

Memória distribuída:

Problemas que não cabem em apenas um computador devem ser computados de forma distribuída.

Exemplo: Análise de grandes volumes de dados utilizando o ecossistema Hadoop.

Programação Heterogênea:

Número massivo de operações matemáticas.

Exemplo: Treinamento de uma rede-neural em GPU.