SSC-0143
PROGRAMAÇÃO CONCORRENTE

Aula 08 – Avaliação de Desempenho de Programas Paralelos

Prof. Jó Ueyama

Créditos

Os slides integrantes deste material foram construídos a partir dos conteúdos relacionados às referências bibliográficas descritas neste documento

Visão Geral da Aula de Hoje

1. Origens da Sobrecarga
2. Métricas de Desempenho
3. Granularidade e Desempenho
4. Eficiência e Escalabilidade

INTRODUÇÃO

Introdução

- **Desempenho**: Capacidade de reduzir o tempo de resolução do problema à medida que os recursos computacionais aumentam
- **Escalabilidade**: Capacidade de aumentar o desempenho à medida que a complexidade do problema aumenta

Esses pontos são os objetivos principais ao se construir uma aplicação paralela

Introdução

- **Fatores que condicionam o desempenho e a escalabilidade**
  - **Limites Arquiteturais**
  - **Limites de Algoritmos**

Introdução

- **Limites ao Desempenho**
  - **Arquiteturais**
    - Latência e Largura de Banda
    - Coerência dos Dados
    - Capacidade de Memória
  - **Algoritmos**
    - Falta de paralelismo (código sequencial/concorrência)
    - Frequência de comunicação
    - Frequência de sincronização
    - Escalonamento Deficiente (granularidade das tarefas/ balanceamento de carga)

MÉTRICAS DE DESEMPENHO

Métricas de Desempenho

- Tempo de Execução → O centro das atenções
- Outras medidas também importantes:
  - Uso da rede
  - Vazão
  - Uso da memória

Métricas de Desempenho

- Objetivo
  - Explicar dados observados e prever comportamento em circunstâncias futuras
    - É preciso abstrair detalhes menos importantes
  - Previsão do tempo de execução
    - T = f (N, P, U, ...)
      - N → tamanho do problema
      - P → número de processadores
      - U → número de tarefas
      - ... → outras características

Métricas de Desempenho

- Tempo de Execução
  - Tempo decorrido do momento em que o primeiro processador começa a executar uma tarefa da aplicação até o momento em que o último processador para de executar
  - Pode ser escolhido o tempo de cada processador
    - $T = T^{j}_{comp} + T^{j}_{comm} + T^{j}_{idle}$
  - Ou o tempo total
    - $T = (T^{j}_{comp} + T^{j}_{comm} + T^{j}_{idle}) / P$

Métricas de Desempenho

- **Duas classes de métricas de desempenho**
  - **Para Processadores**
    - Métricas que permitem avaliar o desempenho de um processador com base na velocidade/número de operações que este consegue realizar em determinado espaço de tempo
  - **Para Aplicações Paralelas (Nosso maior interesse)**
    - Métricas que permitem avaliar o desempenho de uma aplicação paralela com base na comparação entre a execução com múltiplos processadores e a execução com somente um processador

Métricas de Desempenho

- **Métricas para Processador**
  - **MIPS**: Millions of Intructions Per Second
  - **FLOPS**: Floating Operations Per Second
  - **SPECint**: Conjunto de programas de tese da SPEC (Standard Performance Evaluation Corporation) que avaliam o desempenho do processador em aritmética de inteiros
  - **SPECfp**: Conjunto de programas de teste da SPEC que avaliam o desempenho do processador em operações de ponto flutuante
  - **Whestone**: Programa de teste sintético que avalia o desempenho do processador em operações de ponto flutuante
  - **Dhrystone**: Programa de teste sintético que avalia o desempenho do processador em aritmética de inteiros

Métricas de Desempenho

- **Métricas para Aplicações Paralelas**
  - Speedup
  - Eficiência
  - Redundância
  - Utilização
  - Qualidade

Métricas de Desempenho

- **Speedup**
  - Medida do grau de desempenho
    - Relação entre o tempo de execução sequencial e o tempo de execução em paralelo

$$S(p) = T(1) / T(p)$$

- T(1) → Tempo de execução com um processador
- T (p) → Tempo de execução com p processadores

Métricas de Desempenho

- **Eficiência**
  - Medida do grau de aproveitamento dos recursos computacionais
    - Mede a relação entre o grau de desempenho e os recursos computacionais disponíveis

$$E(p) = S(p) / p = T(1) / p \times T(p)$$

    - S(p) é o speedup para p processadores

Métricas de Desempenho

- **Redundância**
  - Medida do grau de aumento da computação
    - Mede a relação entre o número de operações realizada pela execução paralela e pela execução sequencial

$$R(p) = O(p) / O(1)$$

- O(1) → é o número total de operações realizadas com 1 processador
- O(p) → é o número total de operações realizadas com p processadores

Métricas de Desempenho

- **Utilização**
  - Medida do grau de aproveitamento da capacidade computacional.
    - Mede a relação entre a capacidade computacional utilizada durante a computação e a capacidade disponível

$$U(p) = R(p) \times E(p)$$

Métricas de Desempenho

- **Qualidade**
  - Medida do grau de importância de utilizar programação paralela

$$Q(p) = S(p) \times E(P) / R(p)$$

Granularidade e Desempenho

Granularidade

- Medida qualitativa da relação de computação para comunicação
- Períodos de cálculo são normalmente separados por períodos de comunicação por eventos de sincronização

Granularidade

- **Paralelismo de grão-fino**
  - Pequenas quantidades de trabalhos computacionais são realizados entre os eventos de comunicação
  - Baixa taxa de computação para a comunicação
  - Facilita o balanceamento de carga
  - Implica alta sobrecarga de comunicação e menos oportunidade para melhorias no desempenho
  - Se a granularidade é muito fina, é possível que a sobrecarga necessária para comunicação e sincronização entre tarefas demore mais do que a computação

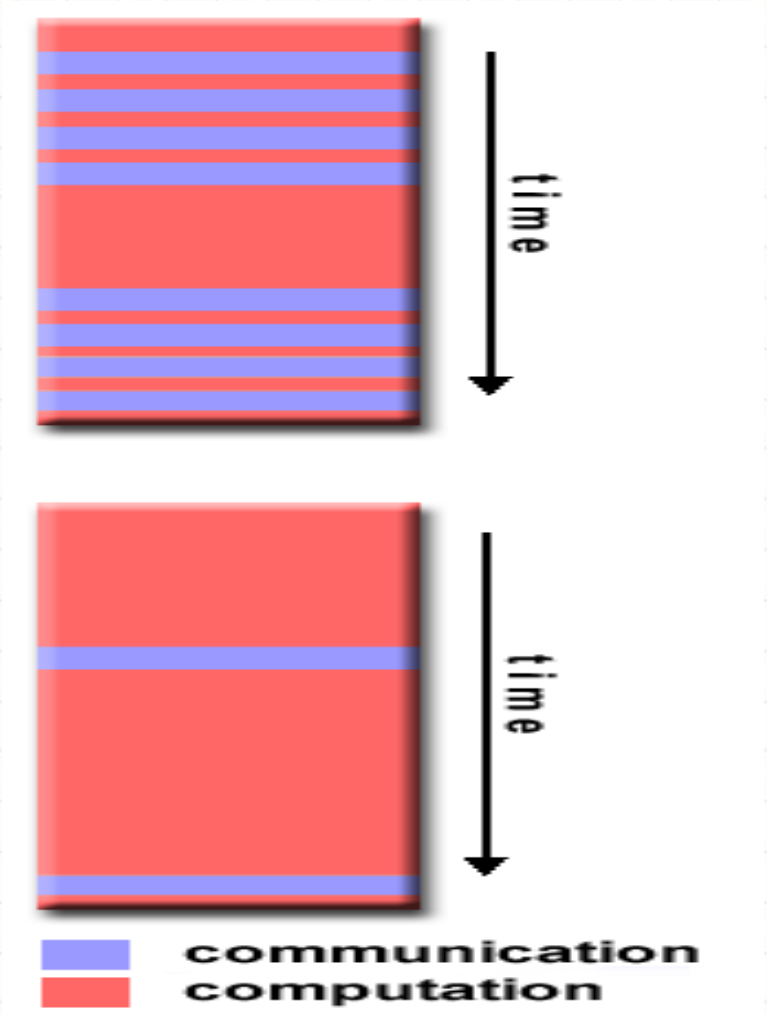
Granularidade

- Paralelismo de grão-grosso
  - Grandes quantidades de trabalhos computacionais são realizados entre os eventos de comunicação/sincronização
  - Alta taxa de computação para a comunicação
  - Maior oportunidade de aumentar o desempenho
  - Mais difícil de balancear a carga eficientemente

Granularidade

- **Grão-Fino x Grão-Grosso**
  - A granularidade mais eficiente depende do algoritmo e do ambiente de hardware no qual ele é executado
  - Na maioria dos casos a sobrecarga associada à comunicação e sincronização é elevada em relação à velocidade de execução. Nessa caso é vantajoso ter granularidade grossa.
  - Paralelismo de grão-fino pode ajudar a reduzir sobrecargas devido ao desequilíbrio das cargas

Granularidade



Eficiência e Escalabilidade

Escalabilidade e Eficiência

- A capacidade de desempenho de um programa escalável é resultado de uma série de fatores inter-relacionados
  - Adicionar máquinas não é a resposta
- O algoritmo pode ter limites inerentes à escalabilidade. Em algum ponto, acrescentar mais recursos pode diminuir o desempenho

Escalabilidade e Eficiência

- Fatores de Hardware desempenham um papel significativo em termos de escalabilidade
  - Barramento CPU-Memória
  - Largura de banda da rede de comunicação
  - Quantidade de memória disponível em qualquer máquinas ou conjunto de máquinas
  - Velocidade do processador
  - Bibliotecas de suporte paralelo e subsistemas de software também pode limitar a escalabilidade independente da aplicação

Eficiência e Escalabilidade

- Dos resultados anteriores podemos concluir que a eficiência de uma aplicação é:
  - Uma função decrescente do número de processadores
  - Tipicamente uma função crescente do tamanho do problema

Eficiência e Escalabilidade

- Um aplicação é considerada eficiente quando demonstra a capacidade de manter a mesma eficiência à medida que o número de processadores e a dimensão do problema aumentam proporcionalmente
- A escalabilidade de uma aplicação reflete a sua capacidade de utilizar mais recursos computacionais de forma efetiva

Dúvidas