

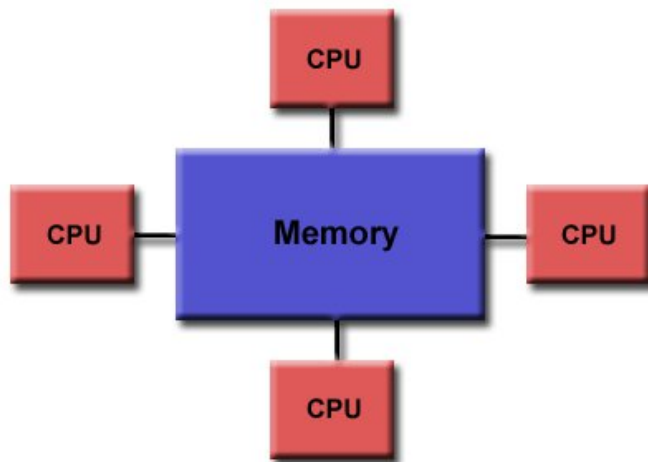
SuperComputação

Aula 16 – Efeitos Colaterais II

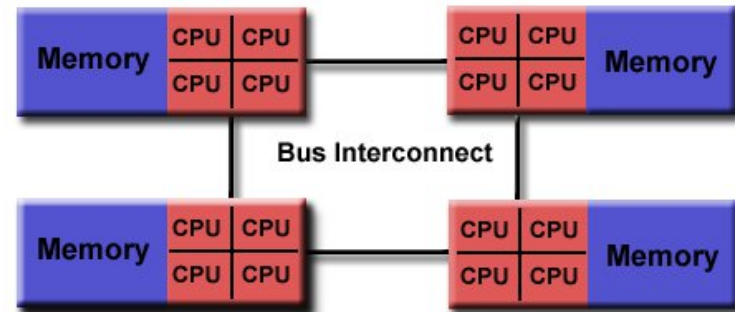
2021 – Engenharia

Igor Montagner <igorsm1@insper.edu.br>
Antônio Selvatici <antoniohps1@insper.edu.br>

Sistemas Multi-core



Uniform Memory Access



Non-Uniform Memory Access

Conceito 1: Dependência

Um loop tem uma **dependência** de dados sua execução correta depende da ordem de sua execução.

Isto ocorre quando **uma iteração depende de resultados calculados em iterações** anteriores.

Quando não existe nenhuma dependência em um loop ele é dito **ingenuamente paralelizável**.

Conceito 2: Paralelismo

Paralelismo de dados: faço em paralelo a mesma operação (lenta) para todos os elementos em um conjunto de dados (grande).

Paralelismo de tarefas: faço em paralelo duas (ou mais) tarefas independentes. Se houver dependências quebro em partes independentes e rodo em ordem.

Conceito 3: Efeito colateral

"Um bloco de código tem efeitos colaterais quando modifica o estado global do programa."

1. Escrever em uma variável compartilhada
2. Mexer no conteúdo de um ponteiro ou referência
3. Ler/escrever em um arquivo
4. Chamar uma função que tem efeitos colaterais

Conceito 3: Efeito colateral

"Um bloco de código tem efeitos colaterais quando modifica o estado global do programa."

Isto pode levar a situações em que

a ordem de execução das operações muda o resultado de um programa.

Conceito 5: Sincronização

"Definir quais ordens de execução entre threads são válidas"

- Threads esperarem umas pelas outras para evitar que façam operações inválidas
 - atualizar uma variável compartilhada
 - usar tipos de dados complexos (`std::vector`)
 - executar operações que precisam ser feitas sem interrupção

Conceito 6: Região crítica

"Bloco de código que só pode ser executado uma thread por vez"

- Força serialização de uma região
- Caro
- Implementada no OpenMP

Problema de hoje: “Monte Carlo”

- Método de simulação/otimização baseado em Sorteios Aleatórios
- “Caro”, mas útil quando outras técnicas não são possíveis
- Altamente paralelizável

Pi com Monte Carlo



Atividade prática

Implementação Sequencial (20 minutos)

1. Exercício de programação em C++



Atividade prática

Parte “É possível paralelizar o problema?” (25 minutos)

1. Analisar problema e identificar oportunidades de paralelismo
2. Utilização de técnicas de sincronização

Atividade prática

Parte “Paralelizando processos sequenciais”

1. Examinar a utilização de geradores de números aleatórios em programas paralelos
2. Implementar duas formas diferentes de paralelismo e comparar resultados



www.insper.edu.br