Insper

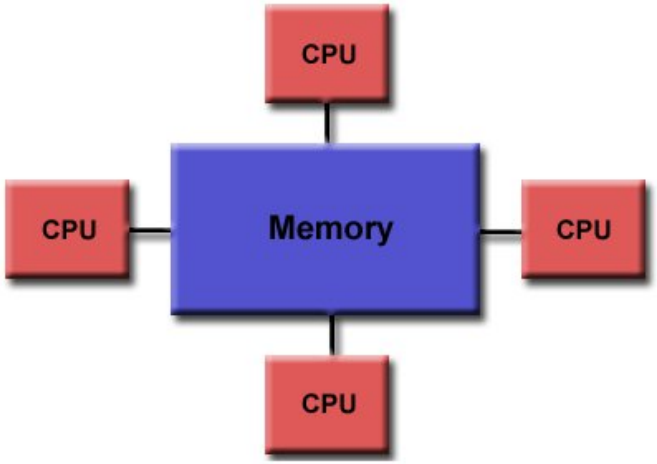
SuperComputação

Aula 15 – Efeitos colaterais e sincronização

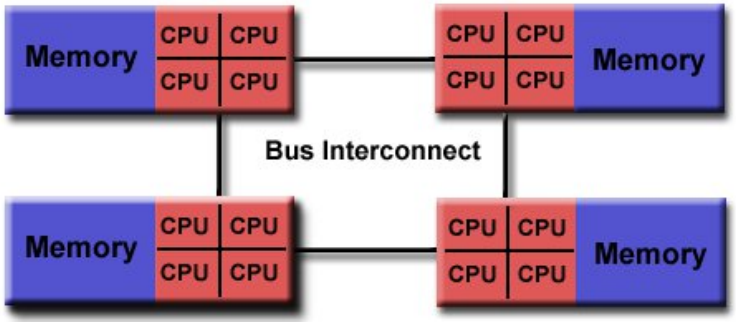
2020 – Engenharia

Luciano Soares <lpsoares@insper.edu.br>
Igor Montagner <igorsm1@insper.edu.br>

Sistemas Multi-core



Uniform Memory Access



Non-Uniform Memory Access

Insper

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**.

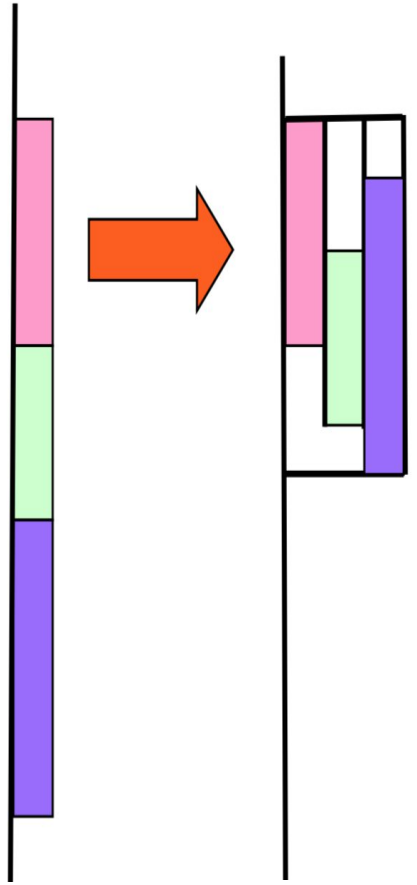
Conceito 2: Paralelismo

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.

Paralelismo de tarefas

- A tarefa é definida em um bloco estruturado de código
- Tarefas podem ser aninhadas: isto é, uma tarefa pode gerar novas tarefas
- Cada thread pode ser alocada para rodar uma tarefa
- Não existe ordenação no início das tarefas
- Tarefa são unidades de trabalho independentes



Serial **Paralela**

Paralelismo de dados

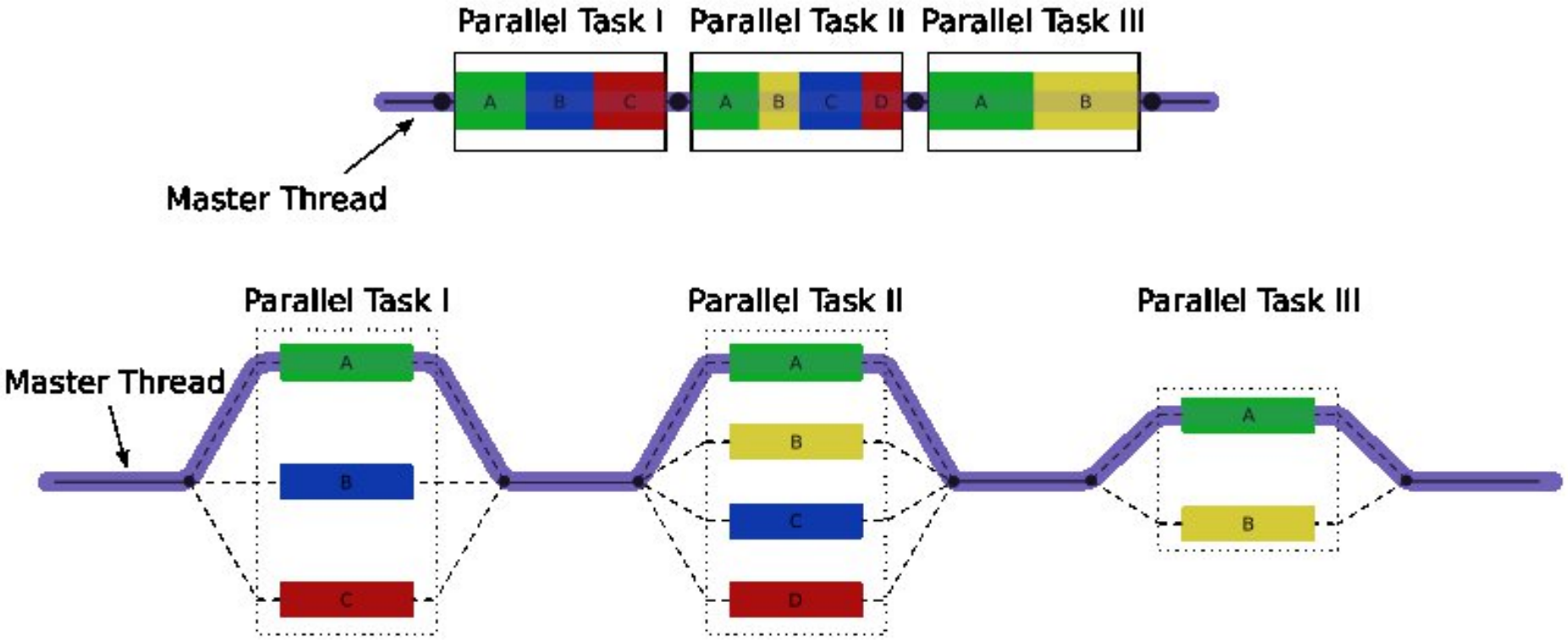


Figura: https://en.wikipedia.org/wiki/File:Fork_join.svg

Comentários

Exercício do pi_recursivo

1. Controlando efeitos colaterais em dados compartilhados

Atividade prática

Seção "Um primeiro teste"

1. Identificar em um código regiões que podem causar problemas se forem paralelizadas.

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.

Efeitos colaterais e dependências

Mundo ideal: nenhuma função modifica o estado global do programa, facilitando muito a paralelização

Mundo real: eliminar todos efeitos colaterais pode tornar o código menos claro, menos eficiente e muito menos legível

Efeitos colaterais e dependências

Linguagens funcionais

Mundo ideal: nenhuma função modifica o e
programa, facilitando muito a paralelização

Mundo real: eliminar todos efeitos colaterais pode tornar o código menos claro, menos eficiente e muito menos legível

Efeitos colaterais e dependências

Mundo ideal: nenhuma função modifica o estado global do programa, facilitando muito a paralelização

Mundo real: **controlar** efeitos colaterais na **parte paralela** do código pode

1. evitar problemas de compartilhamento de dados e de concorrência por recursos
2. facilitar a identificação de dependências possivelmente problemáticas

Conceito 4: Thread safety

"Uma função é threadsafe quando pode ser executada por várias threads sem que ocorram interações não intencionais."

1. Escrever código sem efeitos colaterais
2. Usar primitivas de sincronização

Efeitos colaterais - Estratégias de controle

- Redução (for paralelo)
- Criar cópias por thread de um item de dados compartilhado
  - Resultados podem não ser equivalentes ao sequencial
- **Sincronização**

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

Atividade prática

Parte "Seção crítica"

1. Utilizar recursos do OpenMP para controlar acessos concorrentes a um recurso

Efeitos colaterais e dependências

Mundo ideal: nenhuma função modifica o estado global do programa, facilitando muito a paralelização

Mundo real: **controlar** efeitos colaterais na **parte paralela** do código pode

1. evitar problemas de compartilhamento de dados e de concorrência por recursos
2. facilitar a identificação de dependências possivelmente problemáticas

Atividade prática

Parte "Manejo de conflitos usando pré-alocação de memória"

1. Usar memória de maneira intencional para evitar conflitos

www.insper.edu.br