Introdução à Arquitetura Paralela

Tópicos

- Motivação
- Definições
- Taxonomia
- Arquiteturas correntes

Motivações para Arquiteturas Paralelas

- Performance
- Custo
- Disponibilidade e confiabilidade

Arquitetura Paralela

- *Máquina Paralela*: conjunto de elementos de processamento (PEs) que cooperam para resolver problemas computacionalmente difíceis rapidamente
  - *Quantos PEs?*
  - *Poder computacional de cada PE?*
  - *Quanta memória em cada PE?*
  - *Como o dado é transmitido entre PEs?*
  - *Quais as primitivas para cooperação?*
  - *Qual a performance?*
  - *Como a máquina escala?*

Por que Estudar Arquiteturas Paralelas

- Paralelismo
  - Alternativa para clocks mais rápidos para melhorar performance
    - já vimos isso também em processadores superescalares
  - Aplica-se em todos os níveis do projeto de sistemas
  - Abre novas perspectivas para arquitetura de computadores
    - Por que não colocar múltiplos processadores em um um único chip ao invés de colocar um processador superescalar?

Arquiteturas Paralelas

- Demanda das aplicações: explorar ao máximo o tempo de CPU
  - Computação científica: Biologia, Química, Física
  - Computação de uso geral: Vídeo, Computação Gráfica, CAD, Banco de Dados
- Tendências da tecnologia e das arquiteturas
  - Tecnologia
    - # de transistores crescendo rapidamente
    - Frequência crescendo moderadamente
  - Arquitetura
    - Limites de ILP
- Paralelismo de granularidade grossa, mais viável
- Ganha tempo de avanço tecnológico (100 PEs => 10 anos, 1000 PEs => 15-20 anos)
- Tendência observada nos produtos da SUN, SGI, HP, ...

Tendências Arquiteturais ILP

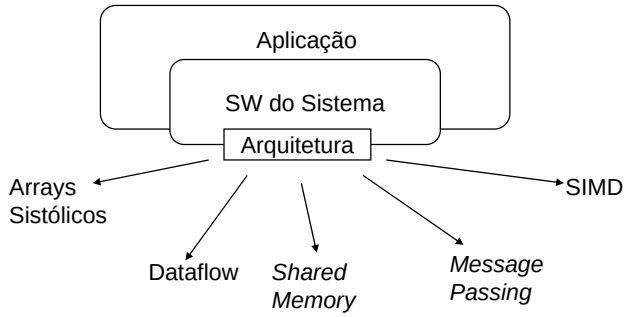
- Speedups reportados para processadores superescalares
  - Horst, Harris, Jardine [1990] 1,37
  - Wang, Wu [1988] 1,70
  - Smith, Johnson, Horowitz [1989] 2,30
  - Murakami, ... [1989] 2,55
  - Jouppi, Wall [1989] 3,20
  - Lee, Kwok, Briggs [1991] 3,50
  - Melvin, Patt [1991] 8,00
  - Butler, ... [1991] 17+
- Variância alta
  - Domínio de aplicações (numérica vs. não numérica)
  - Capacidades da máquina modelada

Arquiteturas Paralelas e Computação Científica

- Computação Científica
  - uPs conquistaram ganhos altos em performance de FP
    - clocks
    - FPUs com pipeline (mult-add todo ciclo)
    - Uso efetivo de caches
  - up são relativamente baratos
    - Custo de desenvolvimento de dezenas de milhões de dólares amortizado sobre os milhões de componentes vendidos
- Multiprocessadores de grande-escala vão substituir supercomputadores tradicionais
  - Já observado na Cray, Intel, IBM, ...

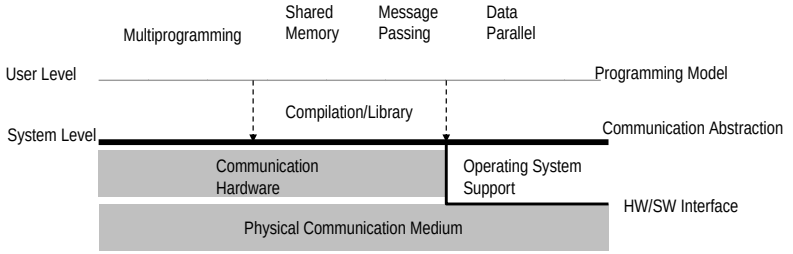
Histórico de Arquiteturas Paralelas

- *Perspectiva histórica:* Modelos e arquiteturas divergentes, sem padrão de crescimento



Arquiteturas Paralelas Hoje

- Extensão de *arquitetura de computadores* para suportar comunicação e cooperação
  - ANTIGAMENTE: *Instruction Set Architecture*
  - HOJE: Arquitetura de comunicação



Arquitetura de Comunicação

= Abstração de Comunicação + Implementação

- Abstração:
  - Primitivas de comunicação de HW/SW para o programador (ISA)
  - Modelo de memória compartilhada, baseado em mensagens, ...
  - Primitivas devem ser eficientemente implementadas
- Implementação
  - Onde interface de rede e controlador de comunicação integram no nó?
  - Rede de interconexão
- Objetivos:
  - Aplicações de uso geral (custo/aplicação)
  - Programabilidade
  - Escalabilidade

Considerações Sobre Escalabilidade

- Tanto máquinas *small-scale* quanto *large-scale* tem seu lugar no mercado
- Custo-performance-complexidade são diferentes para cada máquina

Questões de Projeto

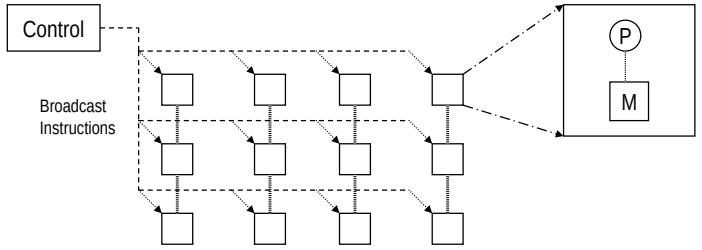
- *Nome*: Como dados compartilhados e/ou comunicação são nomeados?
- *Latência*: Qual é a latência da comunicação?
- *Bandwidth*: Quanto dado pode ser comunicado por segundo?
- *Sincronização*: Como a transferência de dados pode ser sincronizada?
- *Granularidade*: Silício = processador + memória
- *Aplicabilidade*: Propósito geral ou específico?

Taxonomia Histórica Classificação de Flynn

- Baseada nas noções de
  - *Instruction streams*
  - *Data streams*
- Organização da máquina é dada pela multiplicidade do hardware para manipular sequências independentes de dados e instruções
  - *SISD: Single Instruction and Single Data Stream (SparcStation)*
  - *SIMD: Single Instruction and Multiple Data Streams (CM-2)*
  - *MISD: Multiple Instructions and Single Data Stream (CMU Warp)*
  - *MIMD: Multiple Instructions and Multiple Data Streams (Challenge)*

Máquinas SIMD

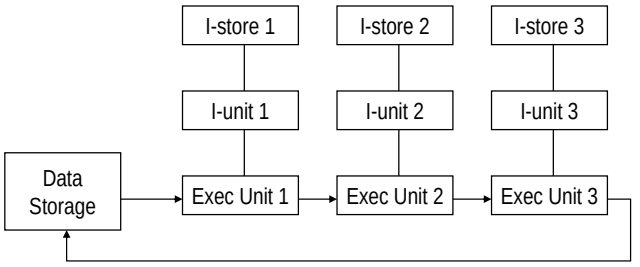
- Exemplos: Illiac-IV, CM-2



- Vantagens: simplicidade de controle, custo, fácil de depurar, baixa latência
- Desvantagens: modelo restrito, pode desperdiçar recursos se somente poucos PEs são utilizados por instrução

Máquinas MISD

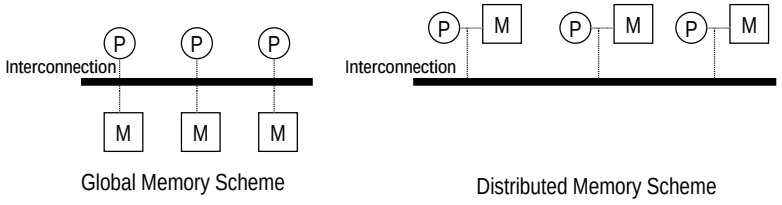
- Exemplo: arrays sistólicos (Warp de CMU)



- Vantagens: simples de projetar, custo-performance alto quando pode ser utilizado (processamento de sinais)
- Desvantagens: aplicabilidade limitada, difícil de programar

Máquinas MIMD

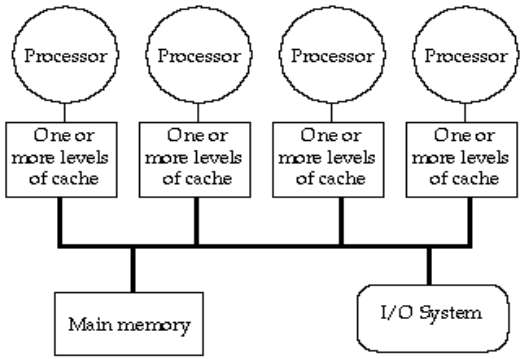
- Exemplo: SGI Challenge, SUN SparcCenter, Cray T3D



- Vantagens: aplicabilidade
- Desvantagens: difícil de projetar bem, overhead de sincronização pode ser alto, programação pode ser difícil

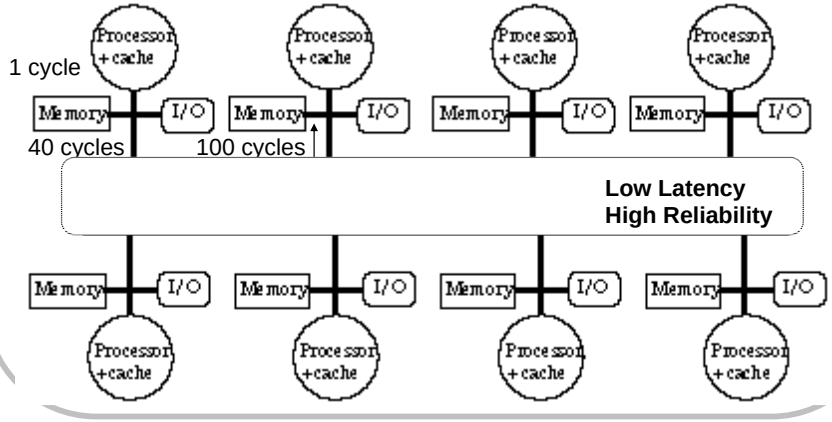
Small-Scale MIMD

- Memória: centralizada com *uniform access time* ("uma") e conexão por barramento
- Exemplos: SPARCCenter, Challenge



Large-Scale MIMD

- Memória: distribuída com *nonuniform access time* ("numa") interconexão escalável (*distributed memory*)
- Examplos: T3D, Paragon, CM-5



Modelo de Comunicação

- *Shared Memory* (centralizada ou distribuída)
  - Processadores comunicam com espaço de endereçamento compartilhado
  - Fácil em máquinas *small-scale*
  - Vantagens:
    - Escolhido para uniprocessadores, MPs small-scale
    - Fácil de programar
    - Baixa latência
    - Mais fácil para usar hardware de controle de cache
- *Message passing*
  - Processadores possuem memórias privadas e comunicam-se via mensagens
  - Vantagens:
    - Menos hardware, fácil de projetar
    - Escalabilidade
- HW pode suportar os dois modelos