

Programação Paralela – OPRP001

Arquiteturas Paralelas

Desenvolvido por Prof. Guilherme Koslovski e Prof. Maurício Pillon

Algumas questões...

- Complexidade computacional
- Custo e desempenho
- Tempo de desenvolvimento?
- Memória compartilhada
- Troca de mensagens
-

Algumas questões...

- Heterogeneidade
 - Recursos computacionais
 - Rede de computadores
 - Latência
 - Largura de banda
 - Protocolos
- Aglomeradores / clusters?
- Migração de processos/tarefas

Definições

- ▢ Sistema distribuído
 - ▢ Transparência
 - ▢ Flexibilidade
 - ▢ Confiabilidade
 - ▢ Desempenho
 - ▢ Escalabilidade
- ▢ E em sistemas paralelos?

Critérios para programação distribuída

- Minimização
 - Do tempo de execução total da aplicação
 - Do número de processadores e de recursos adicionais necessários
- Balanceamento de carga
 - Computacional da aplicação
- Maximização
 - Da utilização dos recursos do sistema
 - Do *throughput* do sistema

Agenda

- ▢ Motivação
- ▢ Definições
- ▢ **Classificação de arquiteturas paralelas**
- ▢ Exemplos de arquiteturas paralelas
- ▢ Estudo de caso

Classificação de Arquiteturas Paralelas

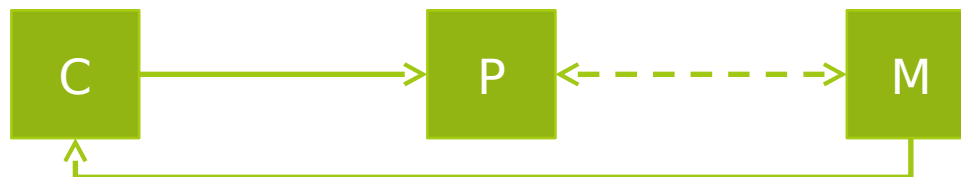
- ▣ Classificação de Flynn [Some Computer Organizations and their Effectiveness. IEEE Transaction on Computers, 1972]
 - ▣ Classificação “acadêmica”
 - ▣ Modelo baseado em fluxo de instruções atuando sobre fluxos de dados
- ▣ Classificação segundo o compartilhamento de memória

Classificação de Flynn

SD (Single Data)		MD (Multiple Data)
SI (Single Instruction)	SISD (Single Instruction Single Data) Máquinas von Neumann convencionais	SIMD (Single Instruction Multiple Data) Máquinas Array
	MISD (Multiple Instruction Single Data) Sem representantes / Controvérsias	MIMD (Multiple Instruction Multiple Data) Multiprocessadores e multicomputadores

Classificação de Flynn - SISD

- Um único fluxo de instruções atua sobre um único fluxo de dados.
- Máquinas de von Neumann tradicionais
- C = Unidade de Controle
- P = Unidade de Processamento
- M = Memória

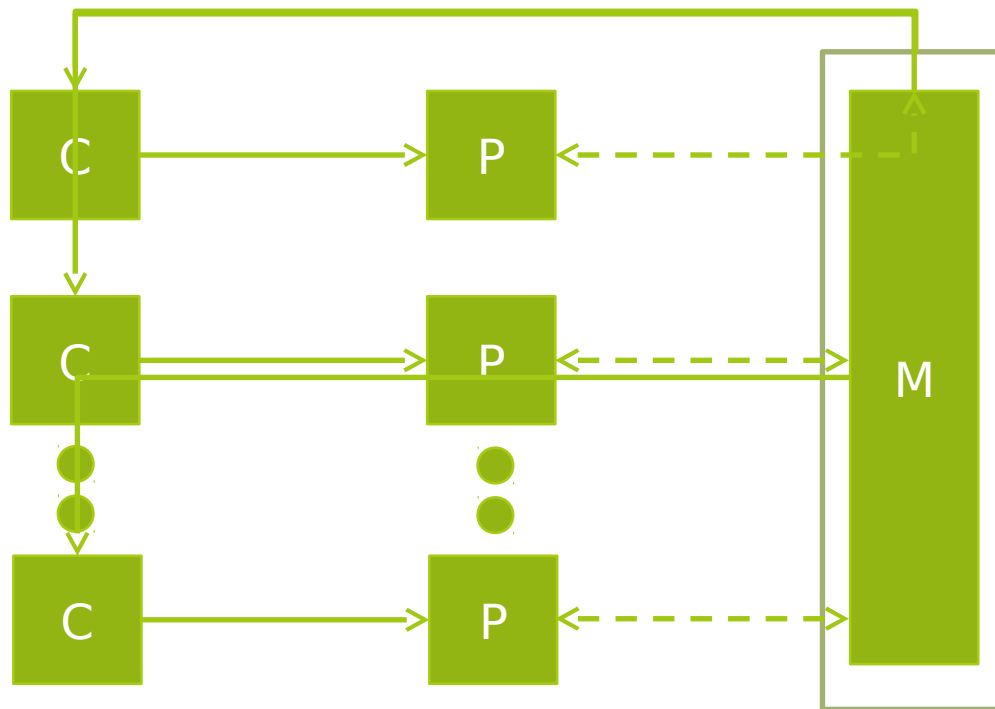


—> Instruções

<- -> Dados

Classificação de Flynn - MISD

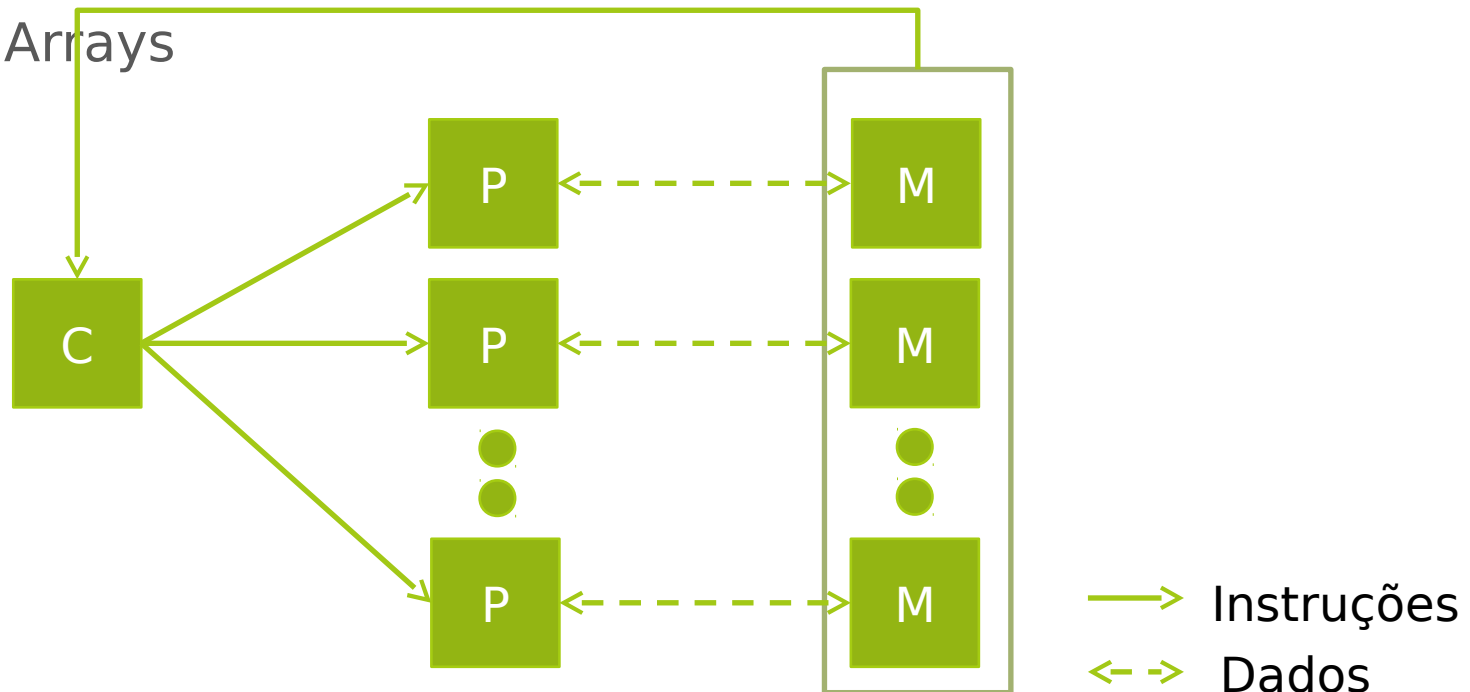
- ▢ Múltiplos fluxos de instruções atuam sobre um único fluxo de dados
- ▢ Tecnicamente viável?



—> Instruções
<- -> Dados

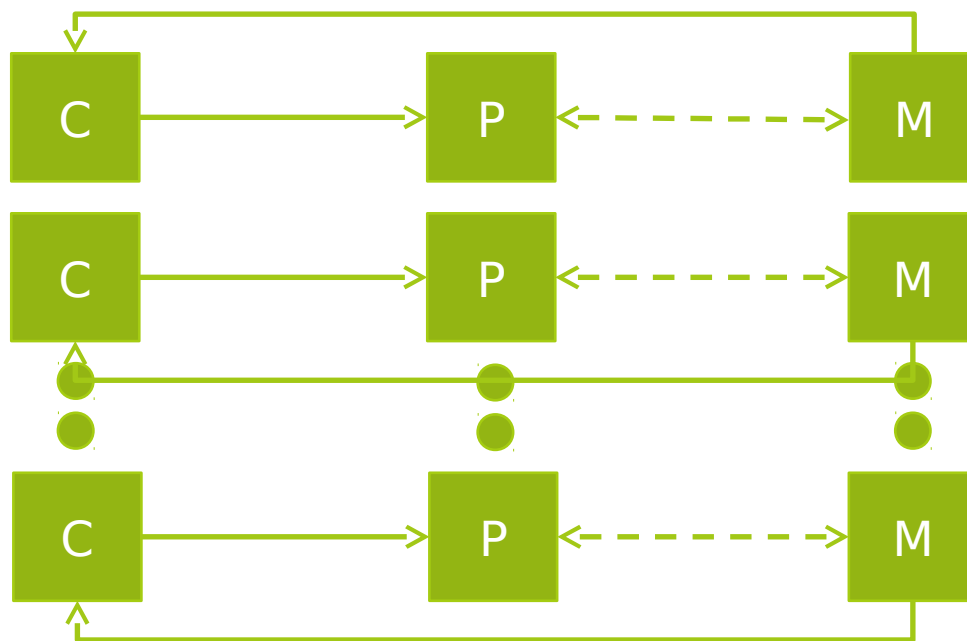
Classificação de Flynn - SIMD

- Uma única instrução é executada ao mesmo tempo sobre múltiplos dados
- Na prática, pode-se dizer que o mesmo programa está sendo executado sobre diferentes dados
- Máquinas Arrays



Classificação de Flynn - MIMD

- ▣ Cada unidade de controle recebe um fluxo de instruções
- ▣ Servidores com múltiplos processadores, redes de estações,



—> Instruções
← -> Dados

Classificação de Flynn

	SD (Single Data)	MD (Multiple Data)
SI (Single Instruction)	SISD (Single Instruction Single Data) Máquinas von Neumann convencionais	SIMD (Single Instruction Multiple Data) Máquinas Array SPMD??
MI (Multiple Instruction)	MISD (Multiple Instruction Single Data) Sem representantes / Controvérsias	MIMD (Multiple Instruction Multiple Data) Multiprocessadores e multicomputadores MPMD??

Classificação segundo o compartilhamento de memória

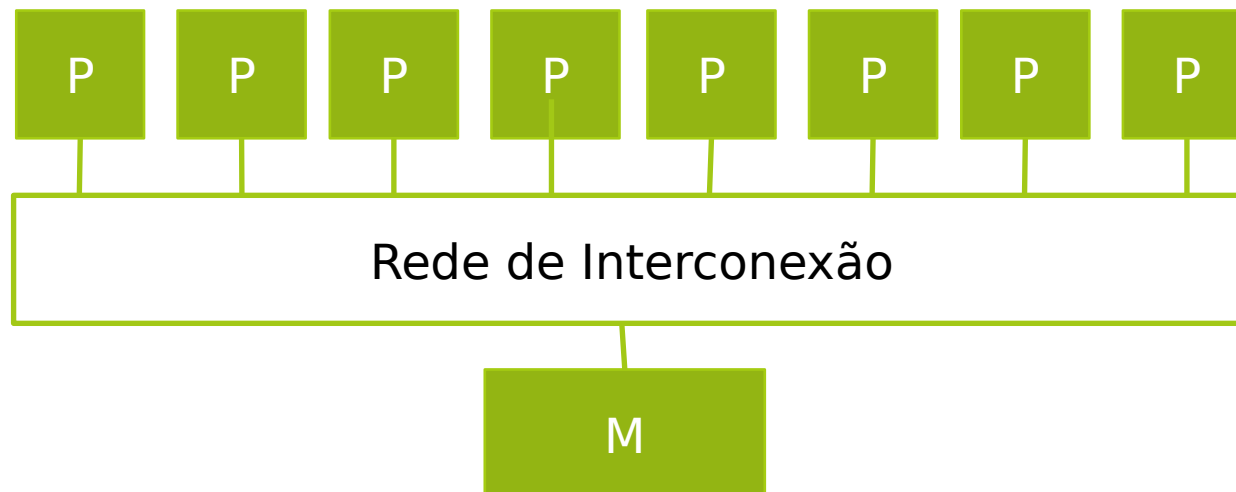
- Memória compartilhada
 - um único espaço de endereçamento é usado de forma implícita para comunicação entre processadores
 - Operações: load e store
- Memória não compartilhada
 - Múltiplos espaços de endereçamento privados, um para cada processador
 - Comunicação explícita através de troca de mensagens
 - Operações: send e receive
- Espaço único de endereçamento?

Classificação segundo o compartilhamento de memória

- ▣ Memória distribuída
 - ▣ Refere-se a localização física da memória
 - ▣ Memória implementada em vários módulos e cada módulo está próximo de um processador
- ▣ Memória centralizada
 - ▣ Encontra-se a mesma distância de todos os processadores
 - ▣ Independentemente de ter sido implementada em vários módulos
- ▣ Distância (latência) no acesso

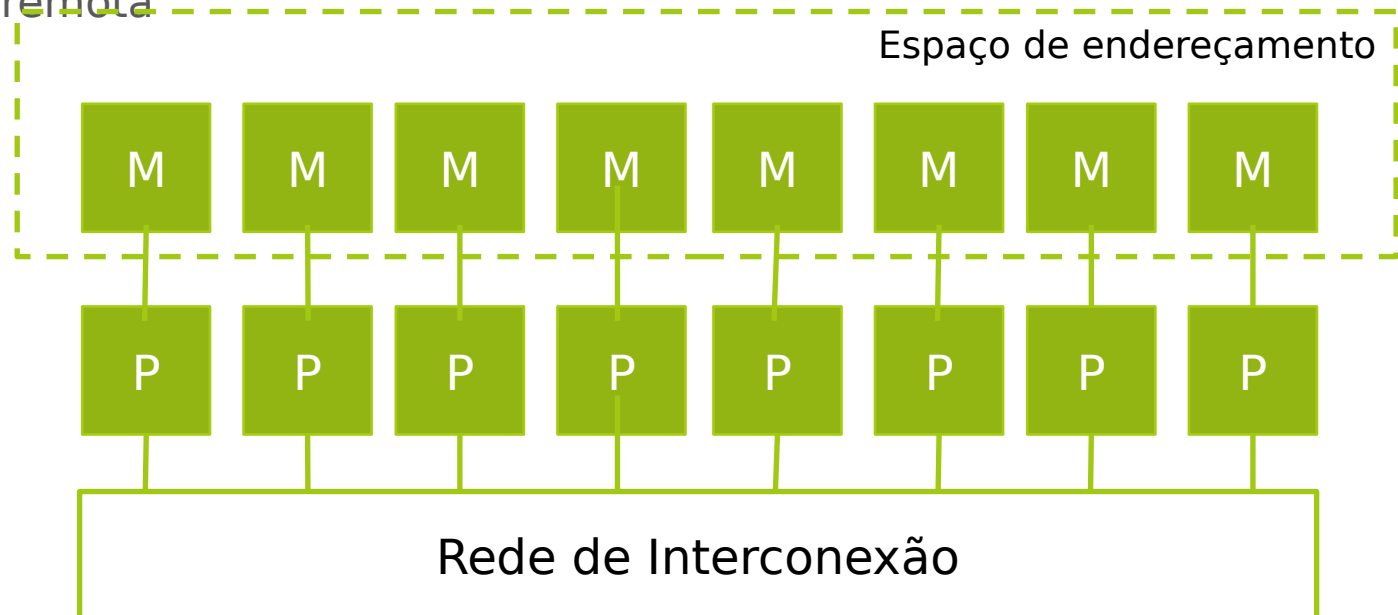
Multiprocessadores: UMA

- ▣ Acesso uniforme à memória (Uniform Memory Access - UMA)
- ▣ Memória que encontra-se a mesma distância de todos os processadores com endereçamento único
- ▣ **Classificação?**
- ▣ Latência de acesso à memória é igual para todos Ps
- ▣ Algumas máquinas implementam memória cache -> problema de sincronização e coerência



Multiprocessadores: NUMA

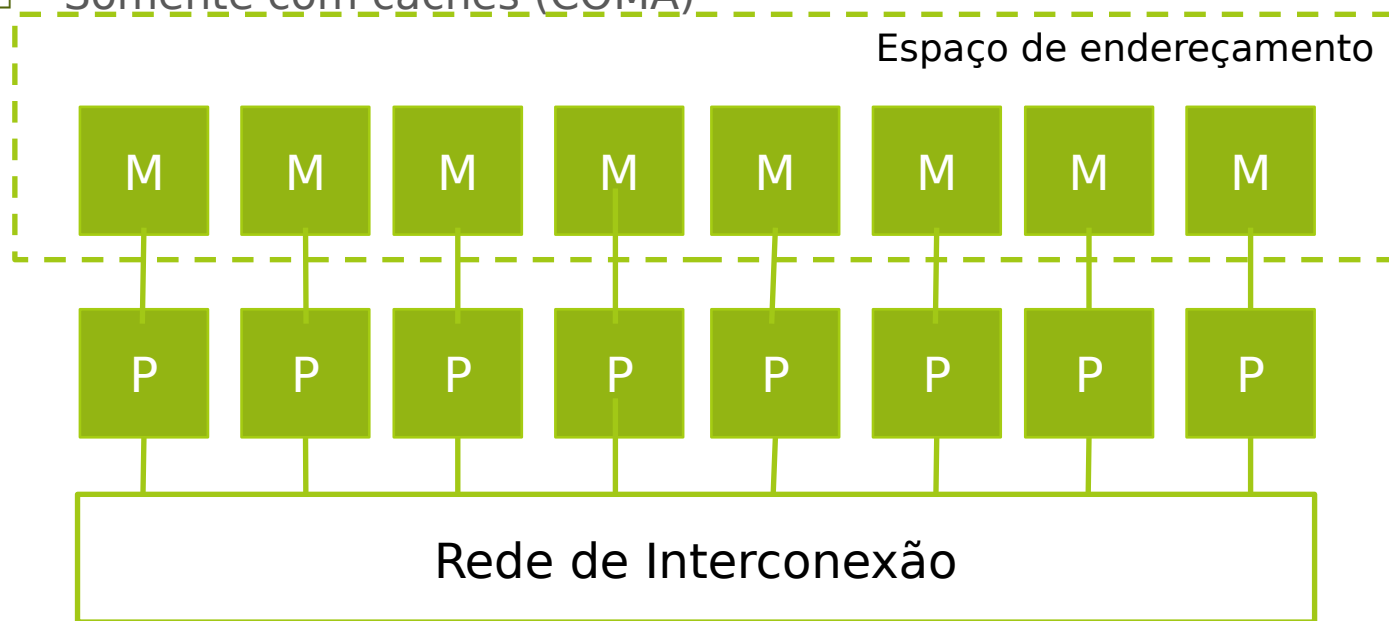
- ▣ Acesso não uniforme à memória (Non-Uniform Memory Access – NUMA)
- ▣ Cada módulo é associado a um processador mas o espaço de endereçamento é único
- ▣ **Classificação?**
- ▣ Não uniforme pois o tempo de acesso (distância) muda entre local e remota



Multiprocessadores: NUMA

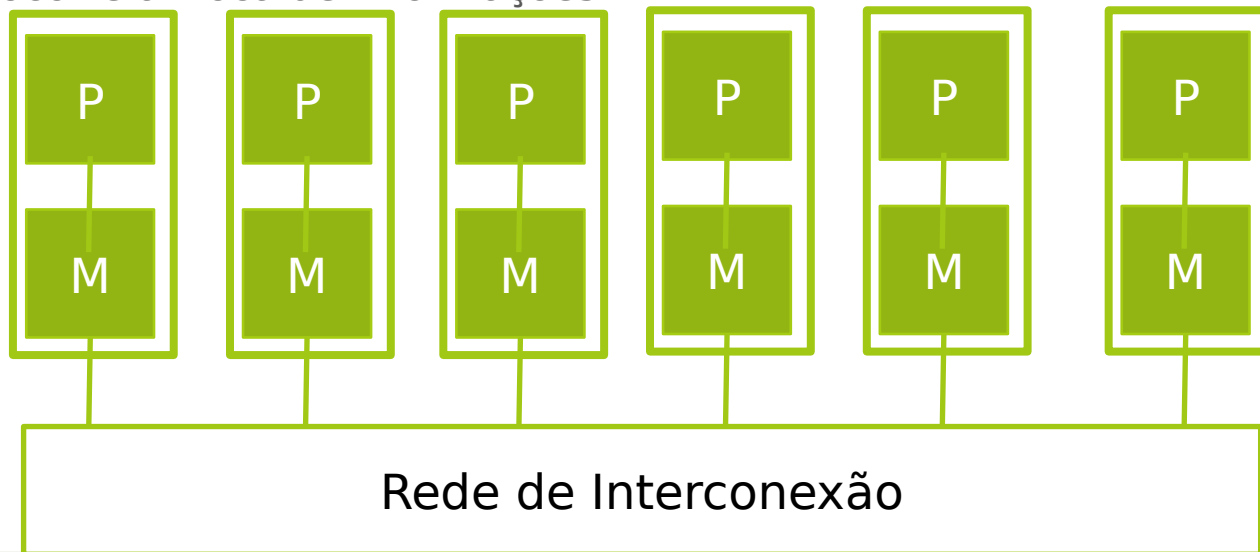
▣ Subdividida em:

- ▣ Sem coerência de cache (NCC-NUMA)
- ▣ Com coerência de cache em hardware (CC-NUMA)
- ▣ Com coerência de cache em software (SC-NUMA)
- ▣ Somente com caches (COMA)



Multicomputadores

- ▣ Cada P possui uma memória local M, à qual só ele tem acesso
- ▣ Outras memórias são consideradas memórias remotas com espaços de endereçamento distintos
- ▣ Em relação ao tipo de acesso à memória são classificados como NORMA (Non-remote memory access)
- ▣ É possível o uso de variáveis compartilhadas nesse ambiente?
- ▣ Como ocorre a troca de informações?



Classificação segundo o compartilhamento de memória

