



# Arquitetura de Computadores Paralelos

**Introdução**

**Conceitos Básicos**

**Ambientes de Programação**

**Modelos de Programação Paralela**

# Por que estudar Computação Paralela e Distribuída?

- Os computadores sequenciais estão cada vez mais velozes:
  - Velocidade do processador;
  - Aumento de memória;
  - Barramento;
- Demanda computacional esta aumentando cada vez mais;
- Limites em processamento sequencial

# Por que estudar Computação Paralela e Distribuída?

- Porque não utilizar mais de um processador?
- Barreiras iniciais:
  - Como paralelizar?
  - O que paralelizar?

# Desafios para a computação de Alto Desempenho (Levin, 1989)

- Química quântica, mecânica estatística, física relativista;
- Cosmologia e astrofísica;
- Dinâmica e turbulência computacional dos fluídos;
- Projeto de materiais e supercondutividade;
- Sequenciamento de Genomas;
- Medicina;
- Clima global;
- Jogos....

# Conceitos

- **Processamento Paralelo:**
  - Processamento de informação concorrente que pertencem a um ou mais processos que resolvem um único problema; Uma aplicação é executada por um conjunto de processadores em um ambiente único (dedicado);
- **Processamento Distribuído:**
  - Processamento de informações em um sistema cujos recursos estão sendo compartilhados por vários programas; Aplicações sendo executadas em plataformas distribuídas;
- **Computador Paralelo:**
  - Computador de múltiplos processadores capaz de realizar processamento paralelo;
- **Supercomputado:**
  - Computador de propósito geral capaz de resolver problemas em alta velocidade, comparado com outras máquinas da mesma época;
- **Concorrência:**
  - Termo mais geral, um programa pode ser constituído por mais de um thread/processo concorrendo por recursos;

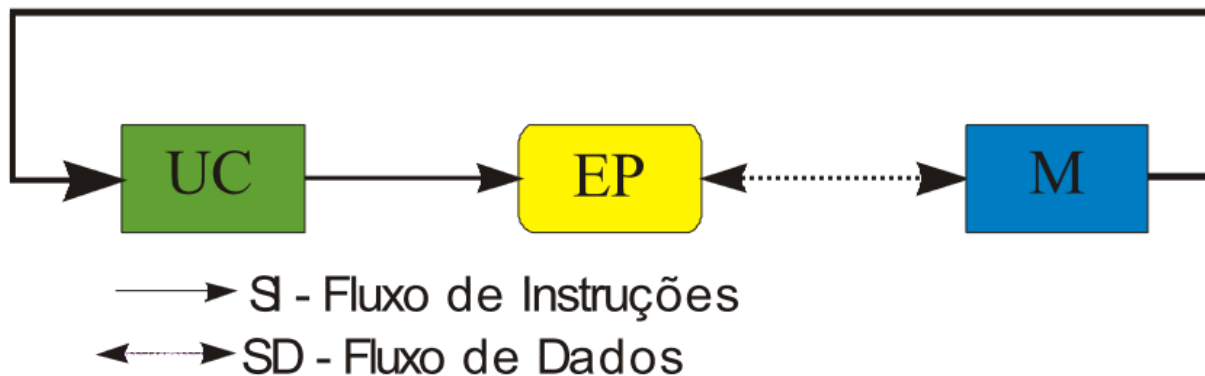
# O que paralelizar?

- Hardware
- Sistema Operacional
- Aplicação
- Pontos essenciais a se focar:
  - Desempenho;
  - Corretude;
  - Possibilidade de se explorar o paralelismo;
- O que se pretende ganhar?
  - Redução no tempo total (memória) de execução.

# Modelos de Programação Paralela

Modelo de computador proposto por  
Von-Neumann

UC – Unidade de controle  
EP – Elemento de processamento  
M - Memória



# Classificação dos Computadores

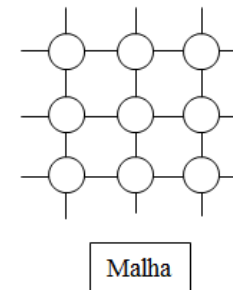
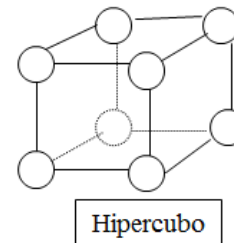
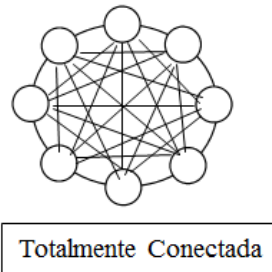
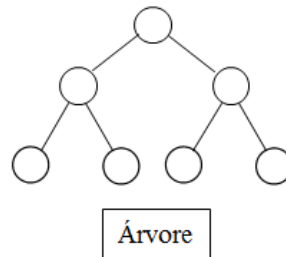
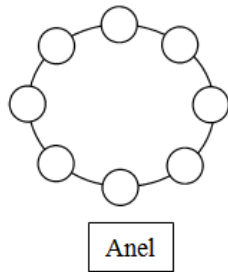
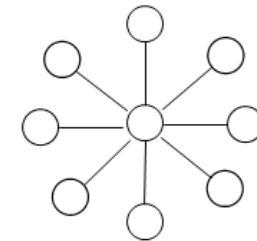
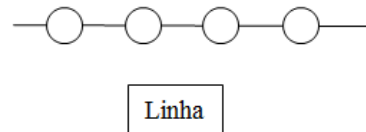
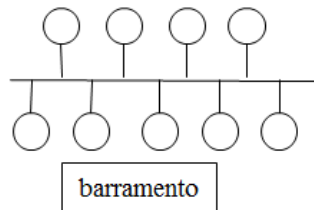
- **Arquitetura**
  - **Computadores Convencionais;**
  - **Memória Centralizada;**
  - **Memória Distribuída;**

# Plataforma de Execução Paralela

- Conectividade
  - Rede de Interconexão;
- Heterogeneidade
  - Hardware e software distintos
- Compartilhamento
  - Utilização de recursos
- Imagem do Sistema
  - Como o usuário o percebe
- Escalabilidade
  - + máquinas = maior desempenho eficiência;

# Topologias

## Topologias



# Classificação de Sistemas Paralelos

- Proposta por Flynn
  - Leva em conta a quantidade de instruções e a quantidade de dados em um determinado momento;
- SISD (*single instruction single data*)
  - Um contador de programa;
  - Computadores Sequenciais;
- SIMD (*single instruction multiple data*)
  - Um contador de programa, uma instrução executada por diversos processadores sobre diferentes dados;
  - Computadores paralelos com PRAM (veremos mais adiante).

# Classificação de Sistemas Paralelos

- **MISD** (*multiple instruction single data*)
  - Systolic array;
  - Não houve êxito comercial;
  - Elementos conectados por um *grid* regular.
  - Controlado por um relógio global;
- **MIMD** (*multiple instructions multiple data*)
  - Vários contadores de programa;
  - Diferentes dados;
  - Os vários computadores paralelos e distribuídos atuais;

# Classificação Flynn

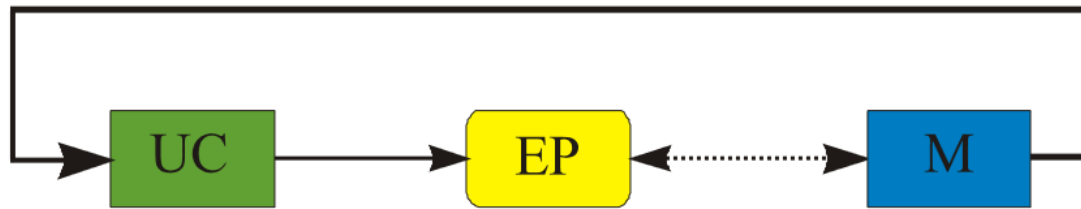
## Classificação de Flynn segundo fluxos de instruções e dados

	<b>SD</b> (Single Data)	<b>MD</b> (Multiple Data)
<b>SI</b> (Single Instrution)	<b>SISD</b> Máquinas von Neumann Convencionais, PC's e estações de trabalho	<b>SIMD</b> Máquinas Vetoriais (ICL Distributed Array Processor – DAP, Thinking Machine Corporations – CM-200)
<b>MI</b> (Multiple Instrution)	<b>MISD</b> Sem representante	<b>MIMD</b> Multiprocessadores e Multicomputadores (nCUBE, Intel Paragon, Cray T3D)

# SISD

## Diagrama da classe SISD

UC – Unidade de controle  
EP – Elemento de processamento  
M - Memória

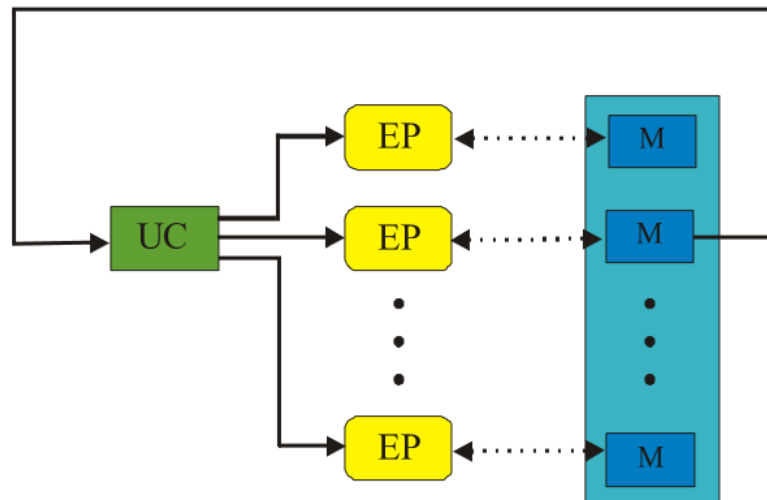


—> S - Fluxo de Instruções

↔ SD - Fluxo de Dados

# SIMD

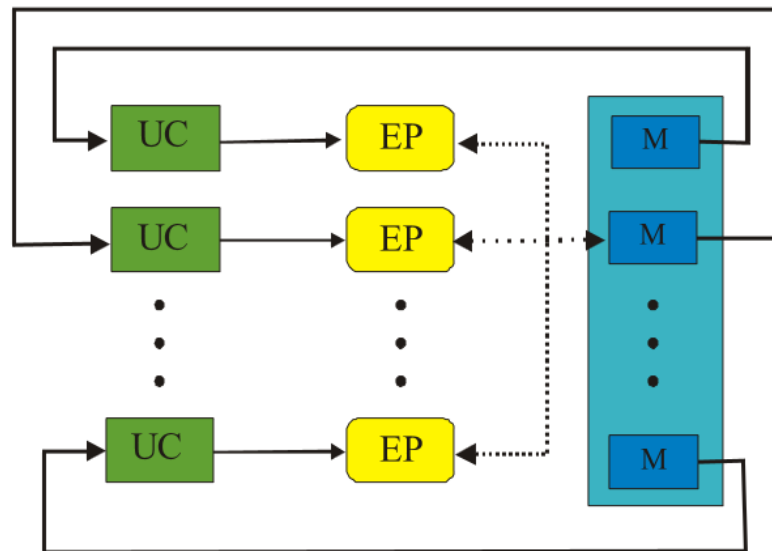
## Diagrama da classe SIMD



→ SI - Fluxo de Instruções  
↔ MD - Fluxo de Dados

# MISD

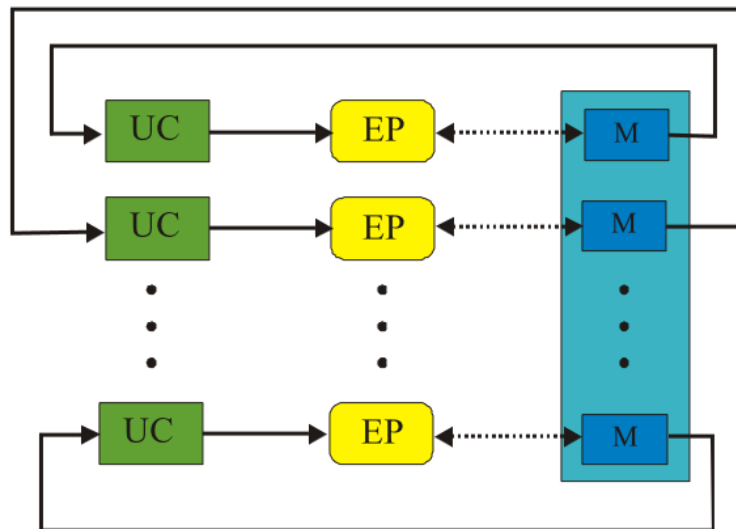
## Diagrama da classe MISD



—→ MI - Fluxo de Instruções  
←···· SD - Fluxo de Dados

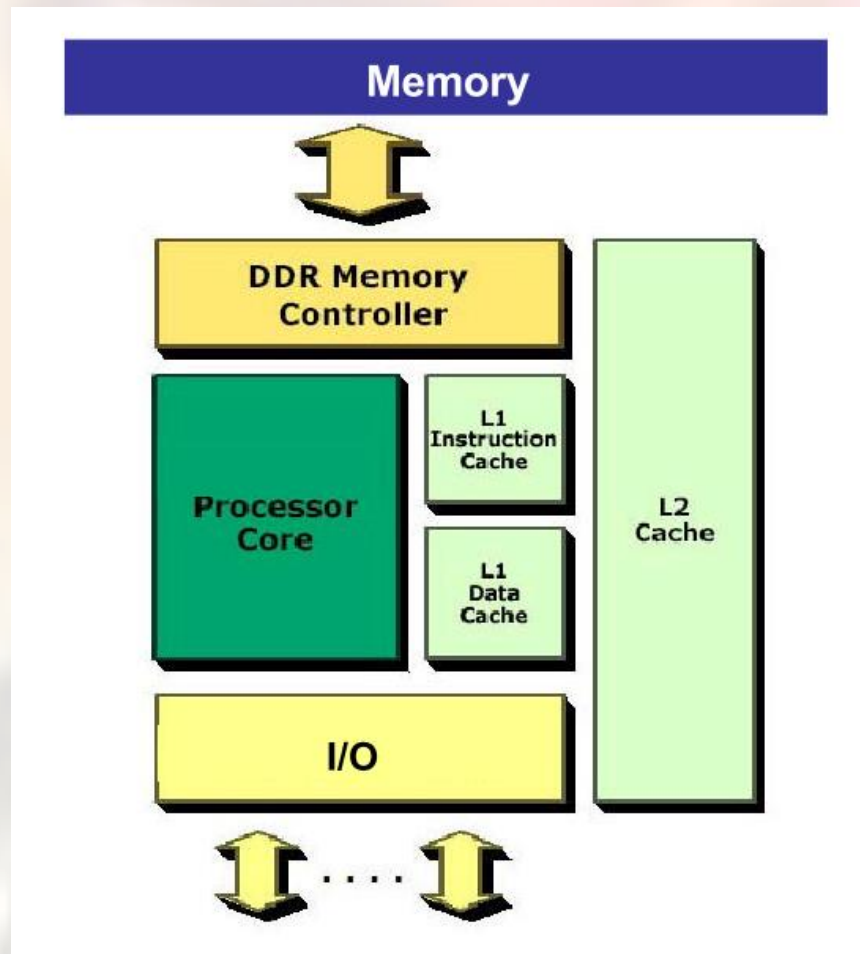
# MIMD

## Diagrama da classe MIMD



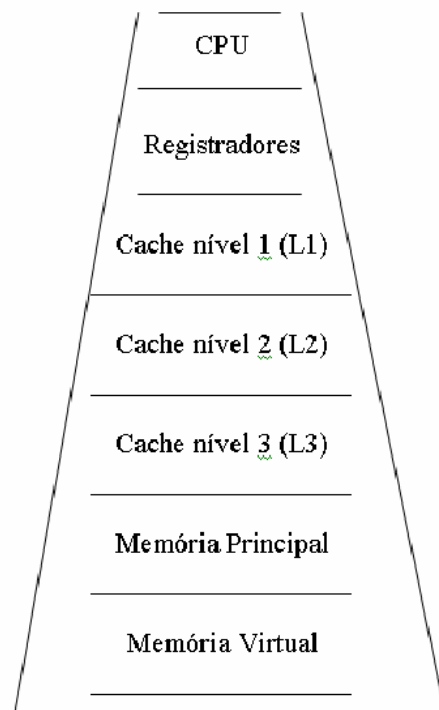
→ MI - Fluxo de Instruções  
← MD - Fluxo de Dados

# Hierarquia de Memória



# Hierarquia Memória

## Hierarquia de Memória

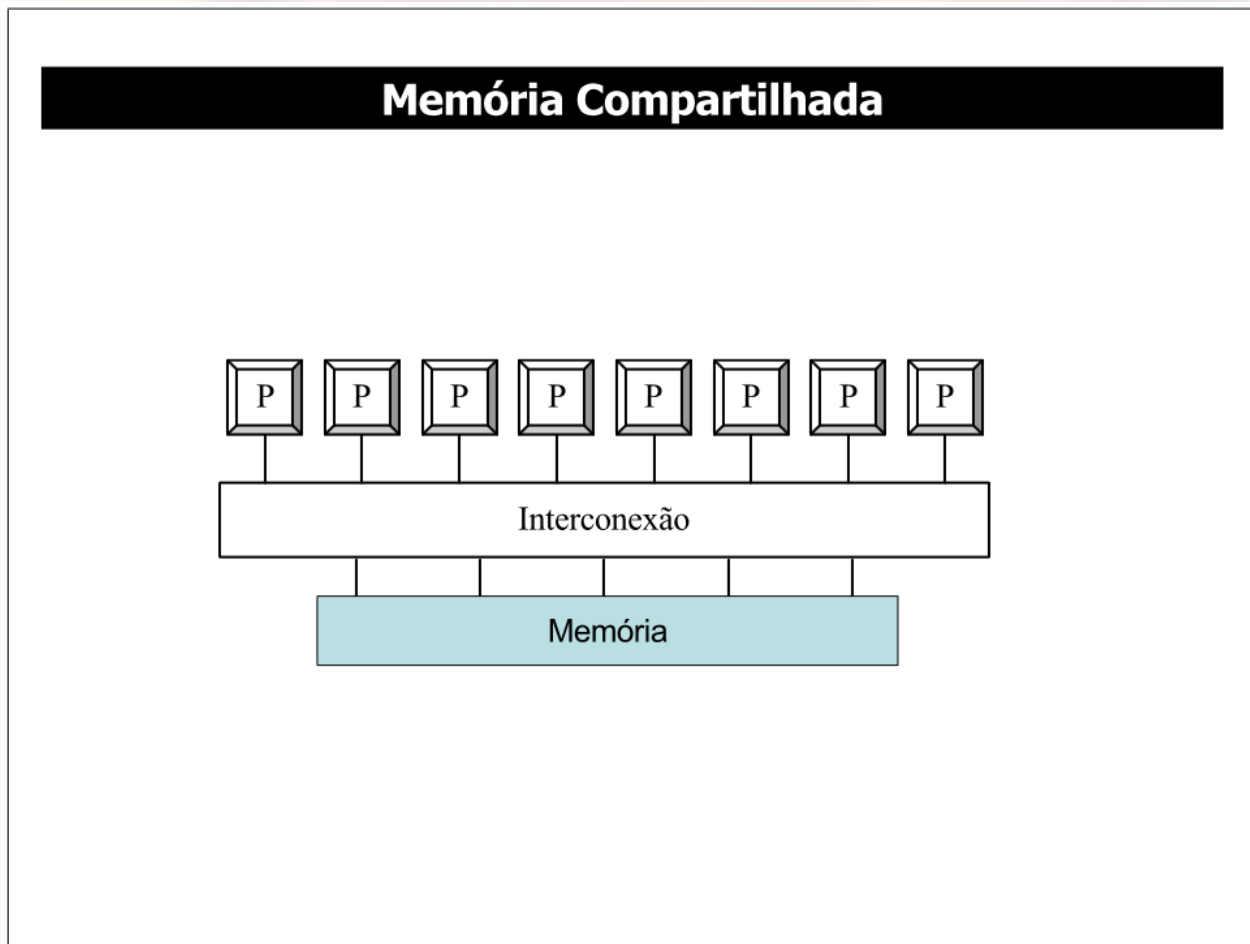


	Capacidade	Velocidade de transferência de dados	Latência
←	256 Bytes	24000 MB/s	2 ns
←	8 Kbytes	16000 MB/s	2 ns
←	96 Kbytes	8000 MB/s	6 ns
←	2 Mbytes	888 MB/s	24 ns
←	1.5 Gbytes	1000 MB/s	112 ns

# Classificação segundo compartilhamento de memória

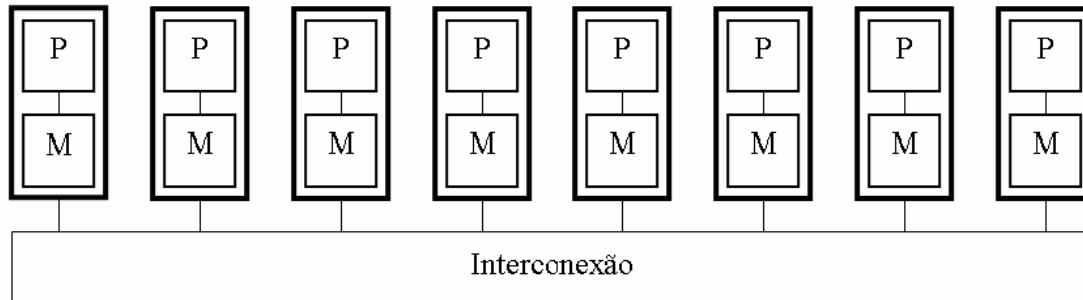
- Memória Compartilhada
  - Existe apenas um espaço de endereçamento, que é usado de forma implícita para a comunicação entre os processadores, com operações de *load* e *store*.
- Memória Distribuída
  - Múltiplos espaços de endereçamento privados, um para cada processador. A comunicação tem que ser explícita, por meio do uso de troca de mensagens com operações *send* e *receive*.

# Memória Compartilhada



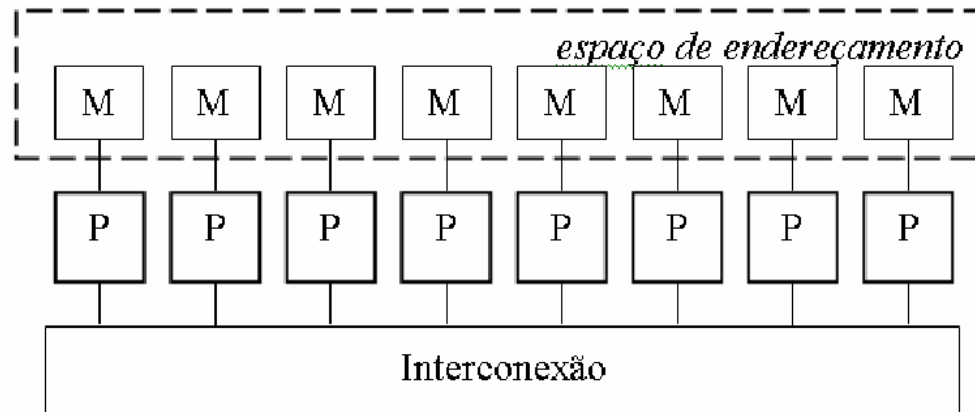
# Memória Distribuída

## Memória Distribuída



# Memória Compartilhada Distribuída

## Memória Compartilhada Distribuída (DSM)



- Endereçamento Único (memória compartilhada)
- Fisicamente distribuída
- Tempo de acesso não uniforme (NUMA  $\approx$  Non-Uniform Memory Access)
- Implementação através de Hardware ou Software

# Comparação entre Máquinas Paralelas

## COMPARAÇÃO ENTRE OS MODELOS FÍSICOS PARA CONSTRUÇÃO DE MÁQUINAS PARALELAS

Tipo	PVP	SMP	MPP	DSM	NOW
Estrutura	SIMD	MIMD	MIMD	MIMD	MIMD
Comunicação	memória compartilhada	memória compartilhada	troca de mensagem	memória compartilhada	troca de mensagem
Interconexão	Crossbar	barramento ou crossbar	rede específica	rede específica	rede comum
Número de nós	~10	~50	Acima de 100	10 a 2000	4 a 10000
Endereçamento	único	único	múltiplo	único	múltiplo
Processador	Específico	comum	comum	específico ou comum	comum
Acesso à memória	UMA	UMA	NORMA	NUMA	NORMA

NORMA – Non-remote memory access

UMA – Uniform memory access

NUMA – Non-uniform memory Access

# Plataforma de Execução Paralela

- Diferentes plataformas do modelo **MIMD** de acordo com os seguintes critérios:
  1. Espaço de endereçamento;
  2. Mecanismo de comunicação.

# Plataforma de Execução Paralela

- Podem ser agrupadas em quatro grupos:
  1. SMPs (*Symmetric Multiprocessors*)
  2. MPPs (*Massively Parallel Processors*)
  3. Cluster ou NOWs (*Network of workstations*)
  4. Grades computacionais ou GRIDs

# Processadores Vetoriais Paralelos - PVP

- Sistemas compostos de poucos processadores poderosos;
- A interconexão é feita, em geral, por uma matriz de chaveamento (*crossbar*) de alta vazão;
- A memória é compartilhada e os sistemas podem ser classificados como processadores UMA.

# Processadores Vetoriais Paralelos - PVP

- Normalmente não utilizam memória cache, usando para esta função um grande número de registradores vetoriais e um buffer de instrução;
- Exemplos: Cray C-90 (máximo de 16 processadores), Cray T-90 (máximo de 32 processadores), Fujitsu VPP 700 (máximo de 256 processadores).
- O NEC Sx-6 também é um PVP, e o earth Simulator, que é um NEC Sx-6 possui 5120 processadores;

# SMPs ou Multiprocessadores

- Único espaço de endereçamento lógico:
  - Mecanismo de hardware (memória compartilhada)
- Comunicação => espaço de endereçamento compartilhado;
  - Operações de *loads* e *stores*.
    - O acesso à memória é realizado por meio de leitura (*load*) e escrita (*store*), caracterizando desta forma, a comunicação entre processadores.

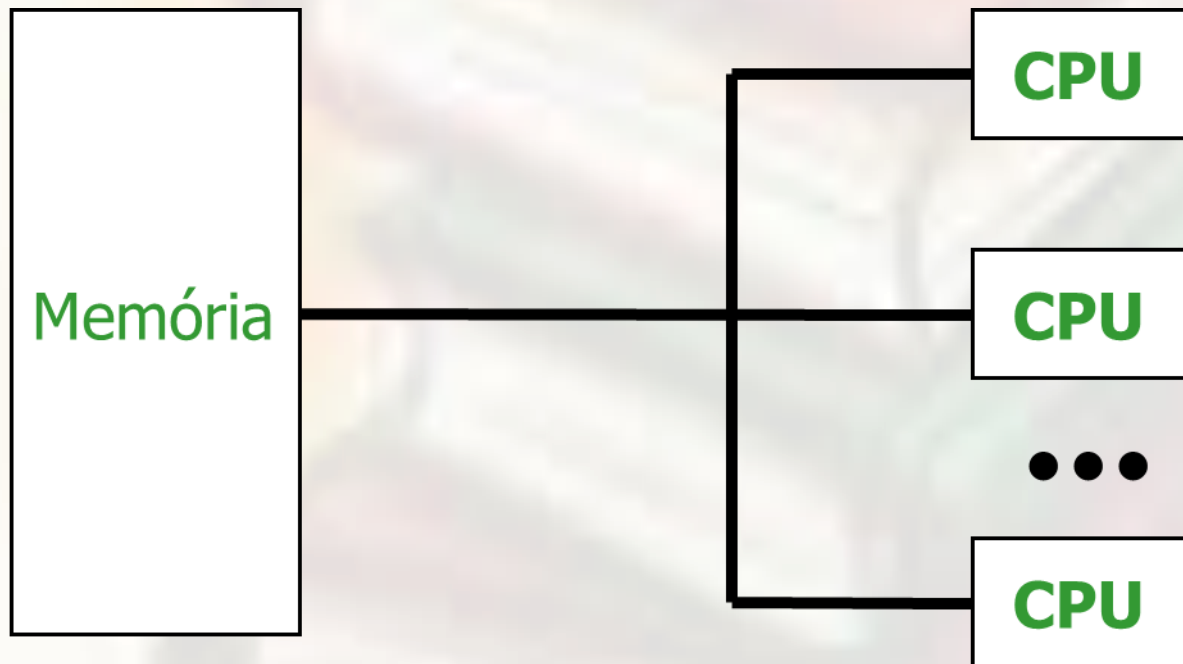
# SMPs

- Sistema homogêneo;
- Compartilhamento total da memória;
- Uma única cópia do sistema operacional;
- Imagem única do sistema;
- Excelente conectividade (fortemente acoplados);
- Não escalável;
- Exemplos: Sun HPC 10000 (star fire), SGI altix, SGI origin, IBM pSeries, Compac AlphaServer.

## **Multiprocessadores simétricos (SMP)**

Os Symmetric Multiprocessors são sistemas constituídos de processadores comerciais conectados a uma memória compartilhada, podendo também ser classificados como multiprocessadores UMA. Utilizam-se amplamente de memória cache e todos os processadores têm igual acesso ao barramento e à memória compartilhada. São mais fáceis de programar que máquinas que se comunicam por troca de mensagens, já que a forma de programação se aproxima daquela feita em sistemas convencionais, mas tem como desvantagem o uso de um barramento de interconexão (permitindo apenas uma transação por vez). Esta limitação pode reduzir a escalabilidade desta classe de sistemas, fazendo com que sistemas comerciais estejam, geralmente, limitados a 64 processadores. Exemplos: IBM R50 (máximo de 8 processadores), SGI Power Challenge (máximo de 36 processadores), SUN Ultra Enterprise 10000 (máximo de 64 processadores) e HP/Convex Exemplar X-Class (máximo de 32 nós de 16 processadores cada).

# SMP - Multiprocessadores



# MPPs - Multicomputadores

- Diferem quanto à implementação física;
- Módulos ou elementos de processamento contendo:
  - Múltiplos processadores com memória privada;
  - Computadores completos;
- Espaço de endereçamento:
  - Não compartilhado – memória distribuída;
- Comunicação – troca de mensagens;
- Rede de interconexão – diferentes topologias;
- Fracamente acoplados;
- Escaláveis;

# MPPs

- Sistema homogêneo (ou heterogêneo);
- Interconexão: Redes dedicadas e rápidas;
- Cada nó executa sua própria cópia do sistema operacional;
- Imagem única do sistema:
  - Visibilidade dos mesmos sistemas de arquivo;
- Um escalonador de tarefas:
  - Partições diferentes para aplicações diferentes.

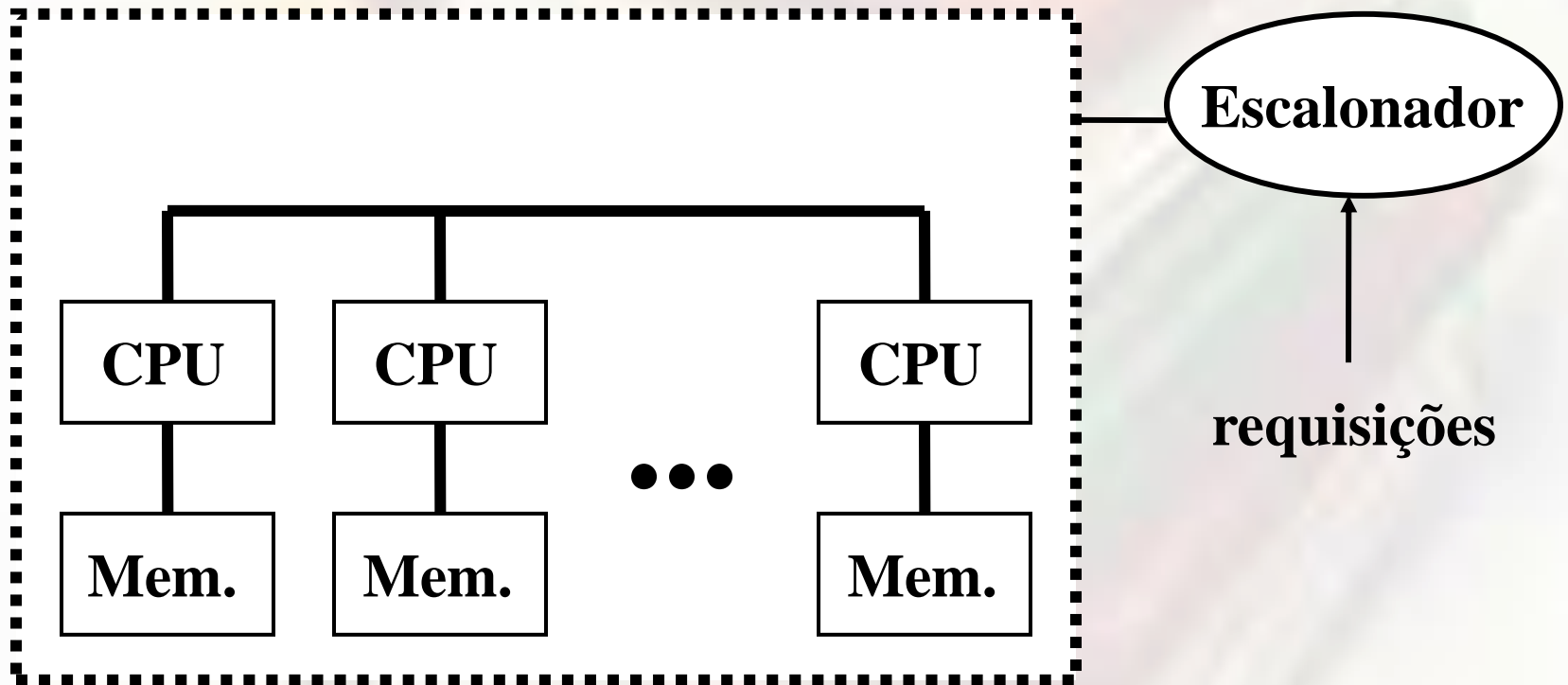
# MPPs

- Partições dedicadas a cada aplicação;
- Aplicações não compartilham recursos;
  - Pode ocorrer de uma aplicação permanecer em estado de espera;
- Exemplos: Cray T3E, IBM SP2s, clusters montados pelo próprio usuário, com propósito de ser um MPP.

## **Máquinas maciçamente paralelas (MPP)**

Os MPPs (Massively Parallel Processors) são multicomputadores NORMA construídos com milhares de processadores comerciais conectados por uma rede de alta velocidade. O alto desempenho é obtido com o grande número de processadores. O fato de haver troca de mensagens torna a programação mais difícil que nos casos em que a memória é compartilhada. Exemplos: Intel Paragon (máximo de 4000 processadores), Connection Machine CM-5 (máximo de 2048 processadores), IBM SP2 (máximo de 512 processadores) e Cray T3D (máximo de 2048 processadores).

# MPPs - Multicomputadores



# *Cluster* de Computadores ou NOWs

- Conjunto de estações de trabalho ou PCs;
- Interconexão: redes locais;
- Nós: elementos de processamento = processador + memória;
- Diferenças em relação a MPPs:
  - Não existe um escalonador centralizado;
  - Redes de interconexão tendem a ser mais lentas;

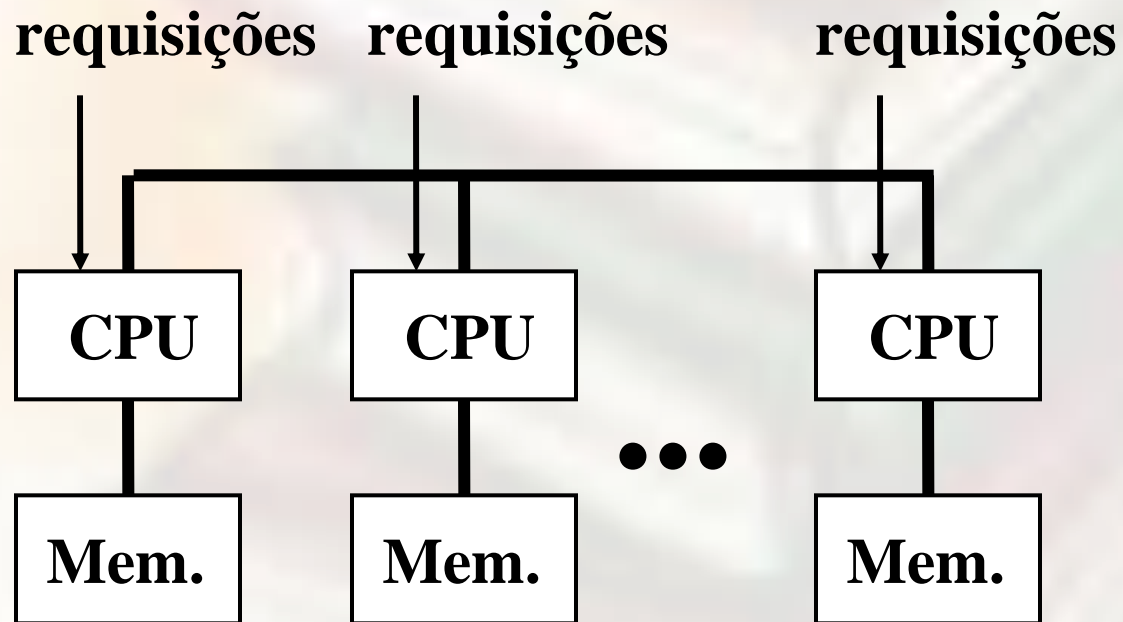
# *Cluster* de Computadores ou NOWs

- Como resultado destas diferenças:
  - Cada nó tem seu próprio escalonador local;
  - Compartilhamento de recursos => Sem partição dedicada a uma aplicação;
  - Aplicação:
    - Deve considerar impacto no desempenho;
    - Não tem o sistema dedicado;
  - Possibilidade de compor um sistema de alto desempenho e um baixo custo (principalmente quando comparados com MPPs);

## Redes de estações de trabalho (NOW)

As redes de estações de trabalho (NOW – *Network of Workstations*) são constituídas de várias estações de trabalho interligadas por alguma tecnologia tradicional de rede, como Ethernet e ATM. Na prática são redes locais utilizadas na execução de aplicações paralelas. Podem ser vistas como máquinas NORMA de baixo custo, ou sem custo algum caso a rede já exista, ou seja, esta é uma solução significativamente mais barata em relação aos MPPs. A desvantagem clara que se vê em uma rede de estações de trabalho é o fato de que as redes tradicionais costumam ser usadas apenas em tarefas menores (para compartilhar arquivos e acessar impressoras remotas, por exemplo), e geralmente não são otimizadas para operações de comunicação de uma aplicação paralela. O resultado é uma alta latência nessas operações, o que compromete o desempenho da máquina como um todo. São usadas principalmente em instituições de ensino para o estudo de processamento paralelo e distribuído. Exemplo: Estações de trabalho interligadas por tecnologia Ethernet.

# *Cluster* de Computadores ou NOWs



# Grades Computacionais (*Computational Grids*)

- Utilização de computadores
  - Independentes;
  - Geograficamente distantes;
- Diferenças entre Clusters e Grades:
  - Heterogeneidade de recursos;
  - Alta dispersão geográfica (escala mundial);
  - Compartilhamento;
  - Múltiplos domínios administrativos;
  - Controle totalmente distribuído;

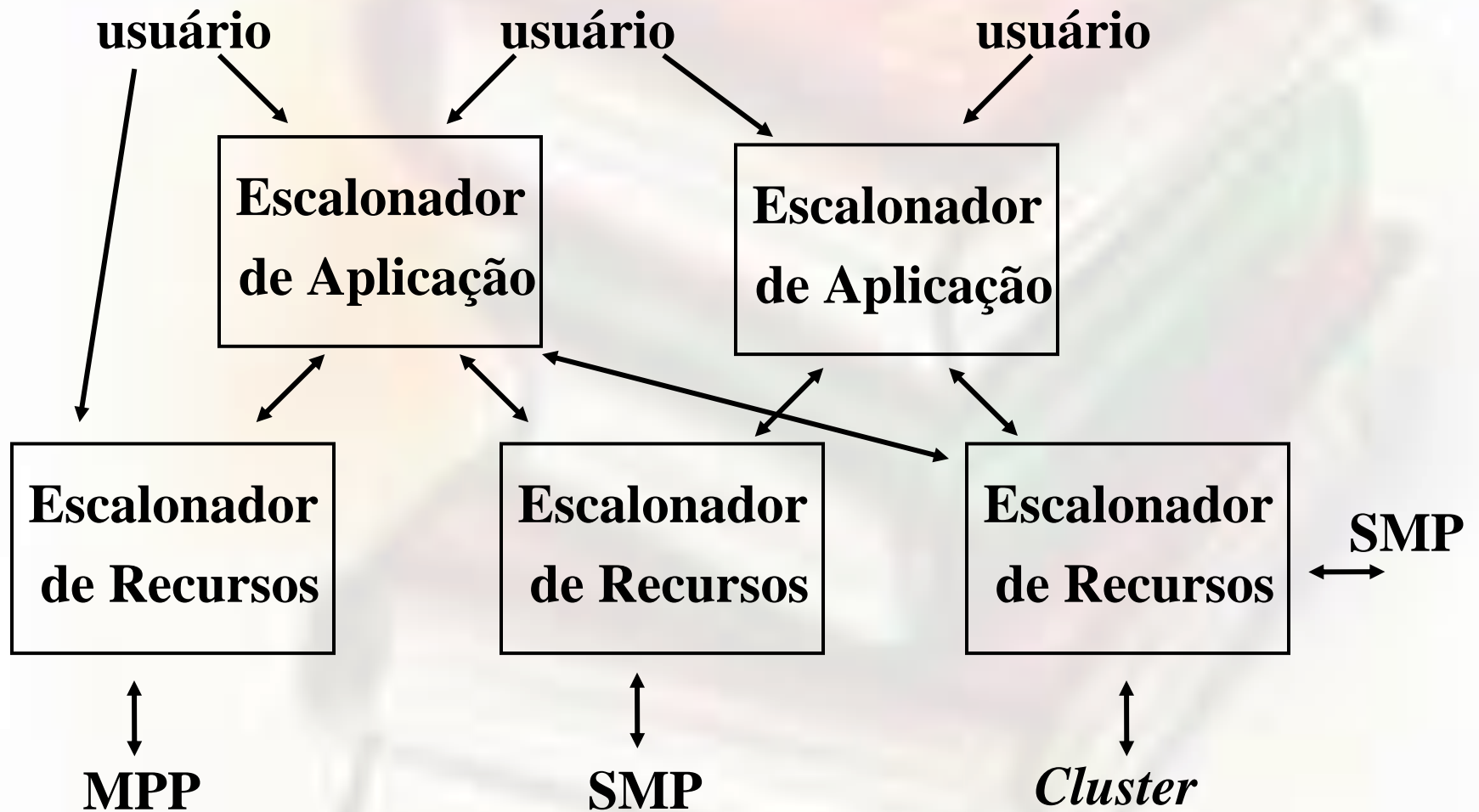
# Grades Computacionais

- Componentes:
  - PCs, SMPs, MPPs, clusters;
  - Controlados por diferentes entidades => diversos domínios administrativos;
- Não tem uma imagem única do sistema a princípio.
  - Vários projetos têm proposto o desenvolvimento de *middlewares* de gerenciamento => camada entre a infraestrutura e as aplicações a serem executadas na grade computacional.
- Aplicação deve estar preparada para:
  - Dinamismo;
  - Variedade de Plataformas;
  - Tolerância a falhas;

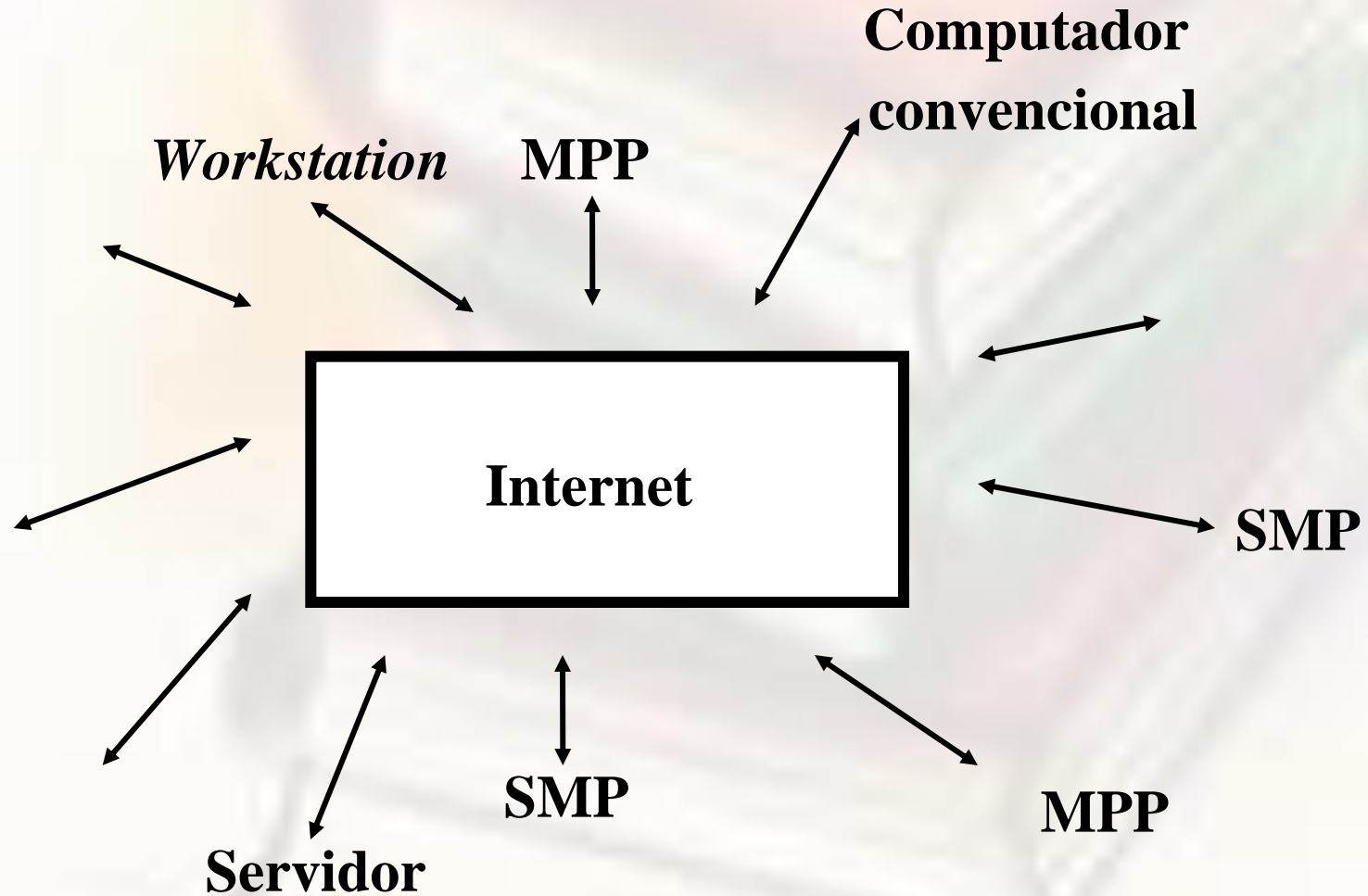
# Grades Computacionais

- Sistema não dedicado e diferentes plataformas
  - Usuários da grade devem obter autorização e certificação para acesso aos recursos disponíveis na grade computacional;
- Falhas nos recursos tanto de processamento como comunicação são mais frequentes que as outras plataformas paralelas;
  - Mecanismos de tolerância a falhas devem tornar essas flutuações do ambiente transparente ao usuário;
- Para utilização eficiente da grade computacional:
  - Gerenciamento da execução da aplicação por meio de políticas de escalonamento da aplicação ou balanceamento de carga;
  - Escalonamento durante a execução da aplicação se faz necessário devido às variações de carga dos recursos da grade.

# Grades Computacionais



# Grades Computacionais



# Computação em Cluster

- Um conjunto de computadores (PCs);
- Não necessariamente iguais => heterogeneidade;
- Filosofia de imagem única;
- Conectadas por uma rede local;
- ❖ Para atingir tais objetivos é necessário uma camada de *software* ou *middleware*.

## **Máquinas com memória compartilhada distribuída (DSM)**

Nos sistemas DSM (Distributed Shared Memory), mesmo com a memória sendo distribuída entre os nós, todos os processadores podem acessar todas as memórias. O espaço de endereçamento único, o compartilhamento de dados e o controle de coerência de cache são conseguidos com software. Podem ser sistemas NUMA com memória entrelaçada distribuída, ou sistemas NORMA (com memórias locais), onde as memórias podem ser ligadas através de adaptadores de rede (AR) a uma rede de interconexão específica, que permite o acesso a memórias remotas. A máquina, nos dois casos, é considerada CC-NUMA ou SC-NUMA dependendo da implementação da coerência de cache. Exemplo: SGI Origin (máximo de 512 processadores).

## COMPARAÇÃO ENTRE OS MODELOS FÍSICOS PARA CONSTRUÇÃO DE MÁQUINAS PARALELAS

Tipo	PVP	SMP	MPP	DSM	NOW
Estrutura	SIMD	MIMD	MIMD	MIMD	MIMD
Comunicação	memória compartilhada	memória compartilhada	troca de mensagem	memória compartilhada	troca de mensagem
Interconexão	Crossbar	barramento ou crossbar	rede específica	rede específica	rede comum
Número de nós	~10	~50	Acima de 100	10 a 2000	4 a 10000
Endereçamento	único	único	múltiplo	único	múltiplo
Processador	Específico	comum	comum	específico ou comum	comum
Acesso à memória	UMA	UMA	NORMA	NUMA	NORMA

NORMA – Non-remote memory access

UMA – Uniform memory access

NUMA – Non-uniform memory Access

# Computação em Grid

- Computação em Cluster foi estendido para computação ao longo dos sites distribuídos geograficamente conectados por redes metropolitanas;

## Grid Computing

- Heterogêneos;
- Compartilhados;
- Aspectos que devem ser tratados:
  - Segurança;
  - Falhas de recursos;
  - Gerenciamento da execução de várias aplicações;

# Computação em Grid

- Modelo ideal:
  - Oferecer desempenho computacional eficiente;
  - De acordo com a demanda;
  - A um custo razoável;
  - Para qualquer um que precisar.
- Basicamente o sucesso da computação em grid depende da comunidade de pesquisadores
  - A possibilidade de construir tal ambiente (hardware e software);
  - Necessidade de atingir tais objetivos;

# Computação em Grid





# Computação em Grid

- *Grid Middleware*: Tem como objetivo facilitar a utilização de um ambiente grid;
  - APIs para isolar usuários ou programas da complexidade deste ambiente;
  - Gerenciar esses sistemas automaticamente e eficientemente para executar aplicações no ambiente grid (*grid enable applications*);

# Computação em Grid

- Como o usuário (dono da aplicação) escolhe?
  - Vários *middlewares* existem, então qual o mais apropriado?
  - Vários ainda estão sendo desenvolvidos;
  - Não há garantia de suporte;
  - Pouca comparação entre os *middlewares*, por exemplo, por desempenho, grau de intrusão;
  - É difícil encontrar grids com o mesmo tipo de software instalado;

# Cloud Computing

- Computação provida como um serviço sobre a internet;
- Infra-estrutura geograficamente distribuída;
- Com algumas características de *autonomic computing*;
  - **Que características são essas?**
- O *middleware* não está embutido na aplicação;
- Exemplos: googleaps, facebook, amazon;
- Exemplo de aplicação: gerenciamento de documentos distribuídos geograficamente.

# *Cloud Computing*

- Modelo econômico de utilização dos serviços;
- Ambiente que provê uma quantidade maior de serviços a baixo custo;
- Baixo consumo de energia;
- Não acontece sem virtualização;