

Fundamentos de Processamento Paralelo e Distribuído

Aula de Introdução e Revisão APD

Prof. Marcelo Veiga Neves

(slides do Prof. César A. F. De Rose)

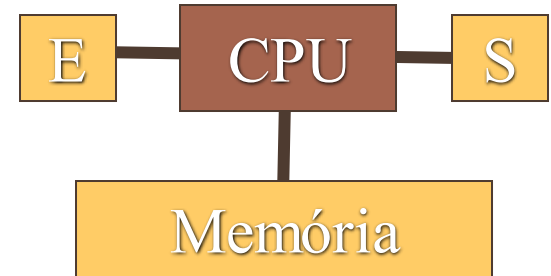
Roteiro

- Histórico e evolução das arquiteturas usadas por estes sistemas
- Revisão das classes de arquiteturas paralelas e distribuídas
- Semelhanças, diferenças e o escopo de aplicação de sistemas concorrentes, paralelos e distribuídos
- Métricas para avaliação destes sistemas

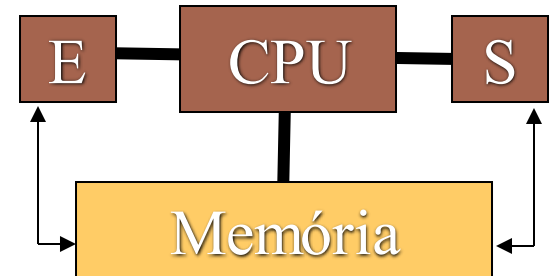


Histórico e Evolução

- Arquitetura Tradicional
 - uma única unidade ativa

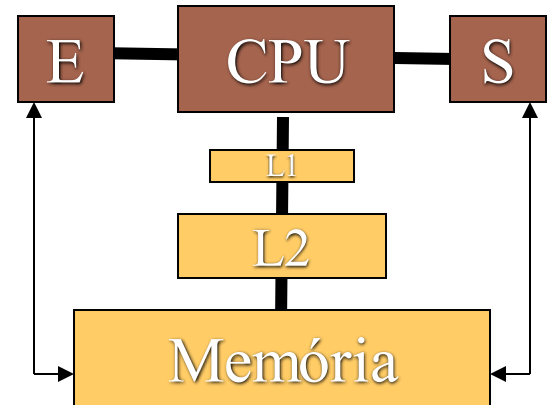


- Unidades de E/S autônomas (Ex: DMA)
 - multiprogramação



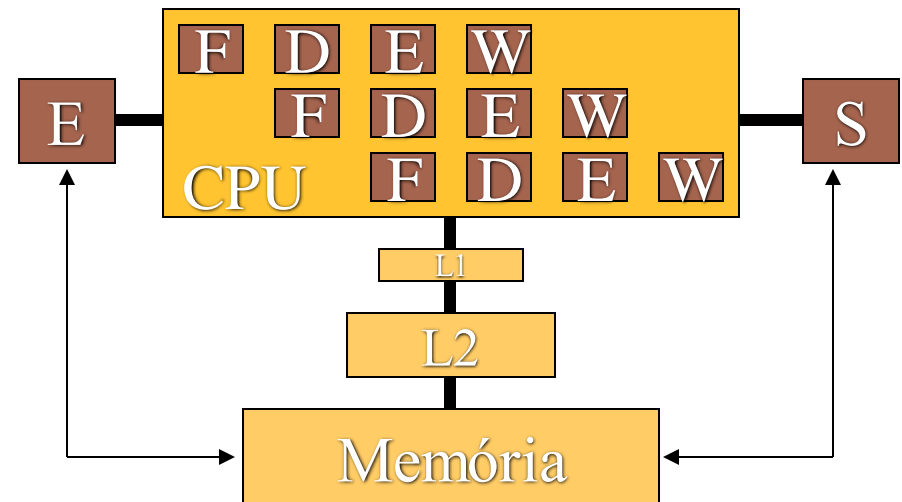
■ Hierarquia de Memória

- acelerar alimentação da CPU
- localidade temporal e espacial
- problema fundamental – I/O

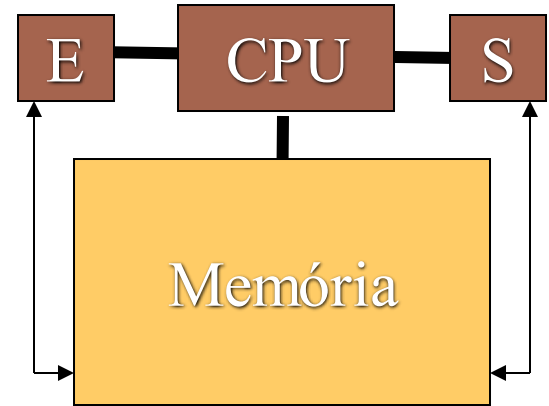


■ Pipeline

- De instrução (superescalar)
- De dados (máquinas vetoriaais)

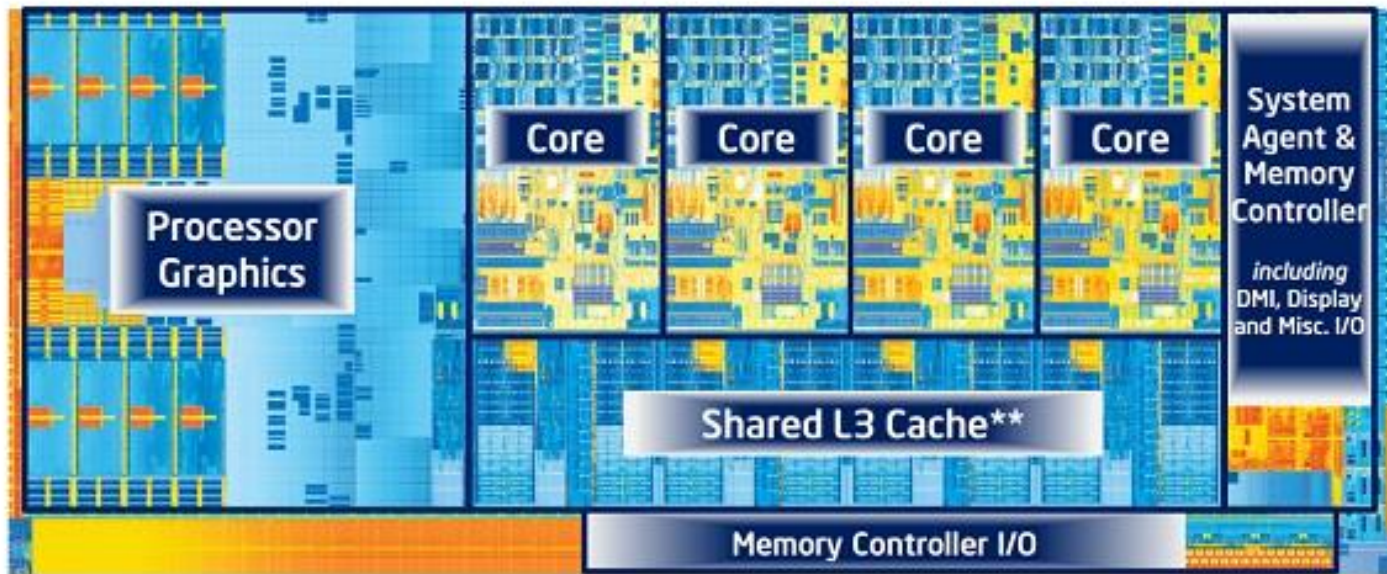
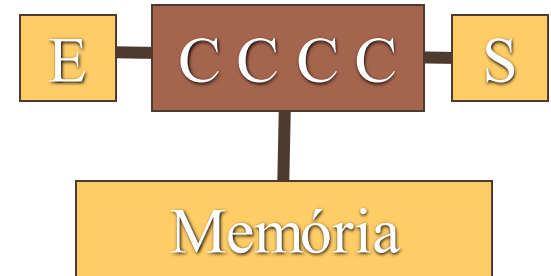


- Memórias não voláteis
 - Reduzir latências de “I/O”
 - SSD, memória única
- Hyperthreading
 - Melhor aproveitar os pipelines superescalares
 - [Video Animação Intel](#)

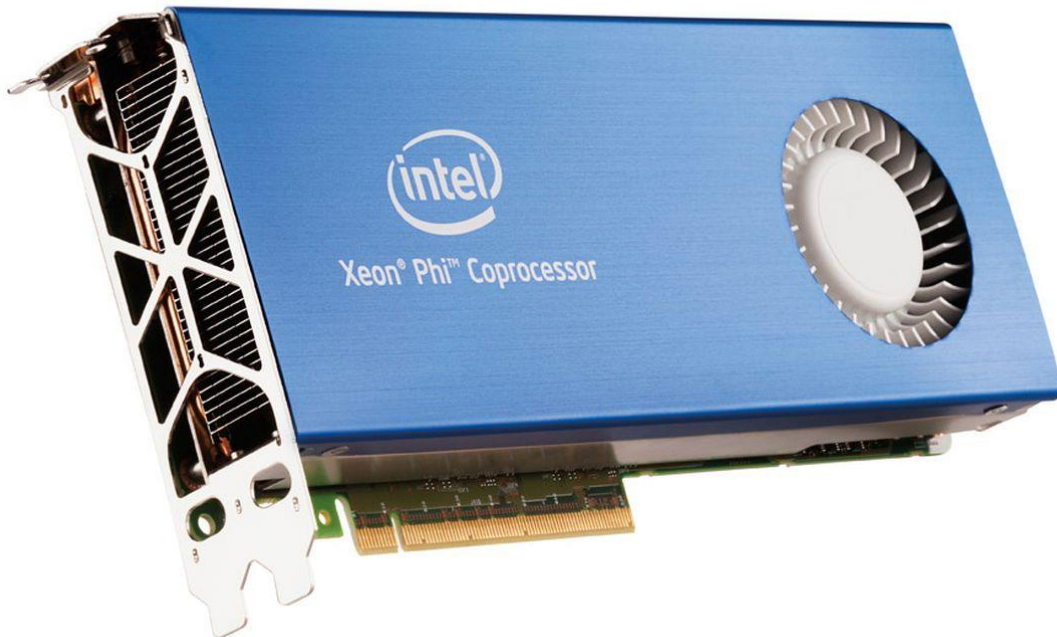


■ Multicore

- Motivação: energia
- Video Animação Intel



- Manycore
 - GPU
 - Unidade geral – GP/GPU
 - Intel Phi



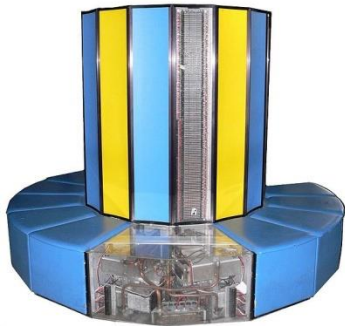


Objetivo desta Evolução

- Acelerar processamento dos dados por parte da CPU
 - Liberando CPU (delegando tarefas)
 - controle do barramento
 - tratamento de E/S
 - Acelerando alimentação da CPU
 - hierarquia de memória
 - Sobrepondo ciclos da CPU
 - pipeline de instrução
- Próximo passo?

Replicação da CPU

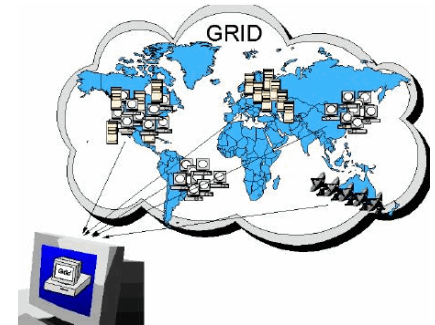
- Acelerar processamento dos dados construindo arquiteturas com múltiplas CPU's
- Arquiteturas Paralelas
- Arquiteturas Distribuídas



Supercomputadores
Vetorial/SMP/NUMA



Clusters of
Workstations

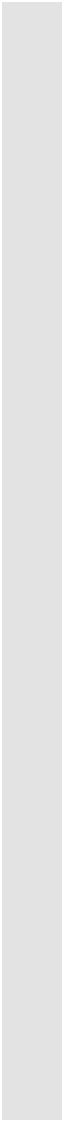


Grid

Escala / Poder Computacional

Acoplamento

Complexidade de uso



Classes de Arquiteturas Paralelas e Distribuídas

Por que estudar classificações ?

- Identificar o critério da classificação
 - por que é importante e quais são as suas implicações
- Analisar todas as possibilidades
 - mesmo as classes que não foram implementadas ou as implementações que não deram certo
- Como se deu a evolução da área
 - como pode evoluir

Classificação de Flynn

- Classificação genérica (1970)
- Diferencia se o fluxo de instruções (*instruction stream*) e o fluxo de dados (*data stream*) são múltiplos ou não

	<i>Single Data</i>	<i>Multiple Data</i>
<i>Single Instruction</i>	SISD	SIMD
<i>Multiple Instruction</i>	MISD	MIMD

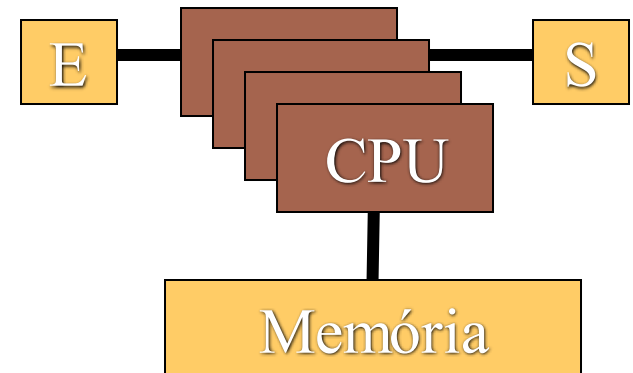
Classificação o Segundo o Compartilha mento de Memória

- Como construir máquinas com vários processadores?



- Compartilho uma memória central
 - Arquitetura tradicional com vários processadores
 - Comunico através da memória
 - Variáveis compartilhadas

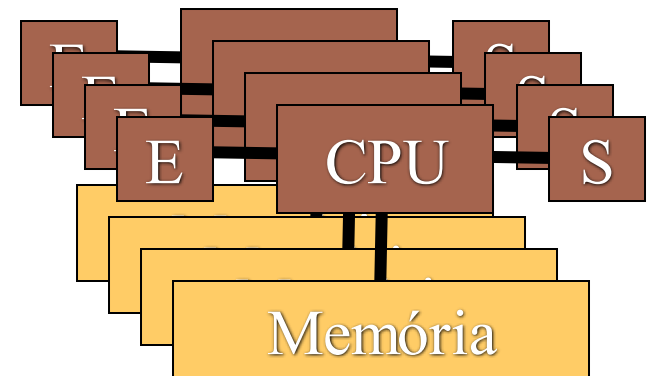
Multiprocessador



■ Não compartilho memória

- Interligação de várias arquiteturas tradicionais
- Cada uma possui sua memória local
- Comunico por troca de mensagens

Multicomputador



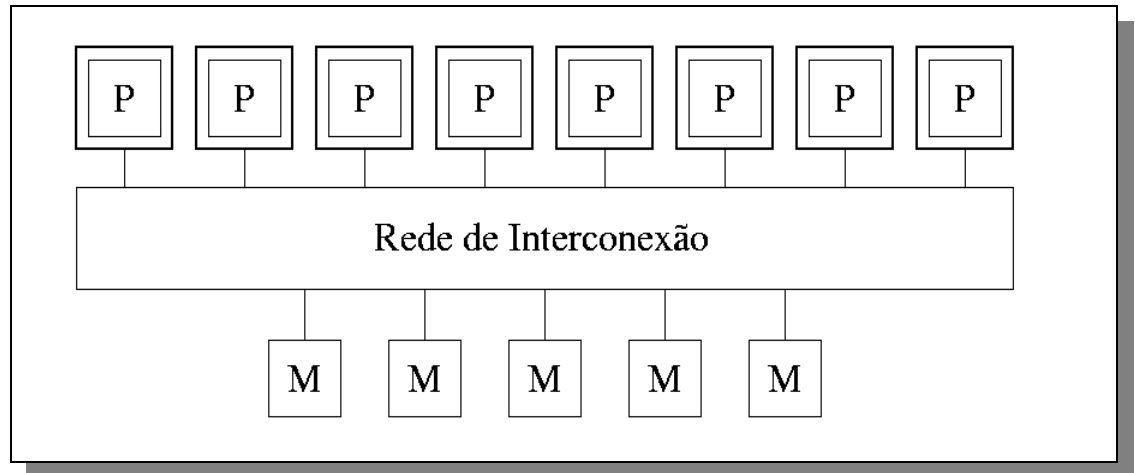
Espaço de Endereçamento

- Área de memória visível ao processador
- Memória compartilhada
 - único espaço de endereçamento
- Memória não compartilhada
 - múltiplos espaços de endereçamento privados
- Memória distribuída
 - localização física da memória
 - oposto: memória centralizada

Classificação Segundo o Tipo de Acesso à Memória

- Multiprocessadores

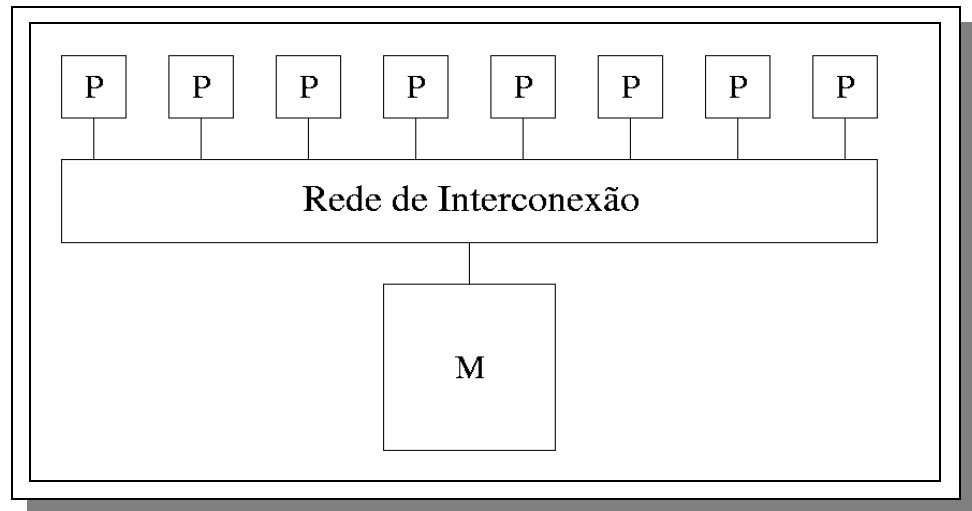
- UMA
- NCC-NUMA
- CC-NUMA
- SC-NUMA
- COMA



Classificação Segundo o Tipo de Acesso à Memória - Multiprocessa- dores

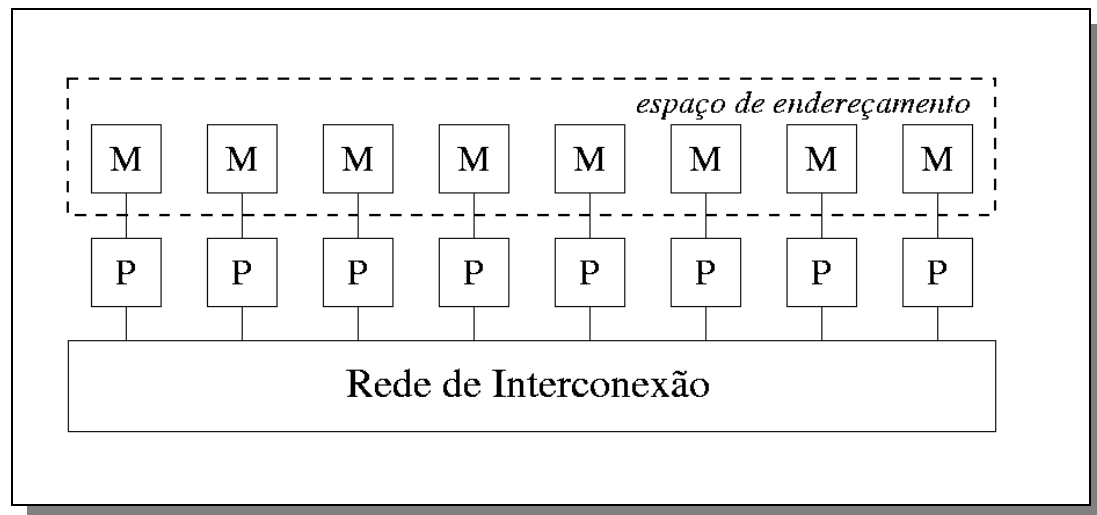
- UMA

- *Uniform Memory Access*
- Memória centralizada (mesma distância de todos os processadores)
- Custo único de acesso
- Precisa tratar coerência de cache



Classificação
Segundo o
Tipo de
Acesso à
Memória -
Multiprocessa-
dores

- NUMA
 - *Non Uniform Memory Access*
 - Único espaço de endereçamento
 - Memória distribuída (distâncias diferentes)
 - Custo não uniforme de acesso à memória



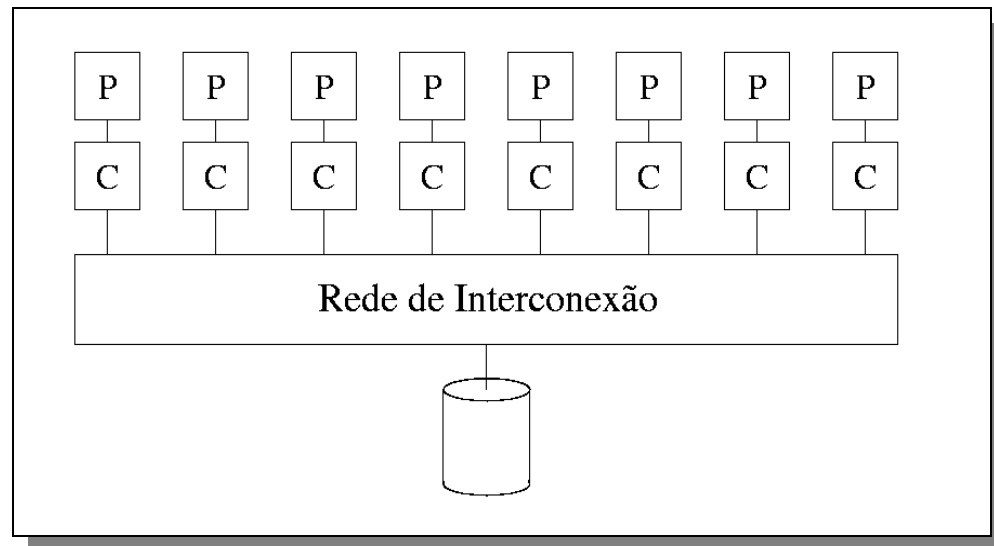
Classificação
Segundo o
Tipo de
Acesso à
Memória -
Multiprocessa-
dores - NUMA

- Em relação ao tratamento do problema de coerência de cache
 - *NCC-NUMA*
 - *non cache-coherent NUMA*
 - *CC-NUMA*
 - *cache-coherent NUMA*
 - implementada em hardware
 - *SC-NUMA*
 - *software-coherent NUMA*
 - implementada em software
 - DSM (Distributed Shared Memory)

Classificação o Segundo o Tipo de Acesso à Memória - Multiproces- sadores

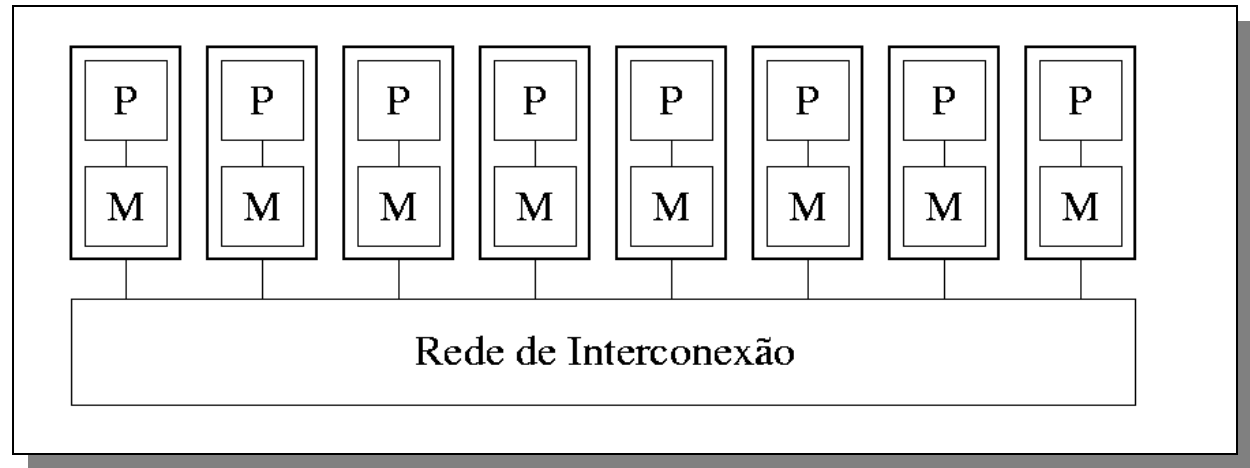
- COMA

- *Cache-only Memory Architecture*
- Memórias locais são caches (coma caches)
- Gerência de caches na MMU



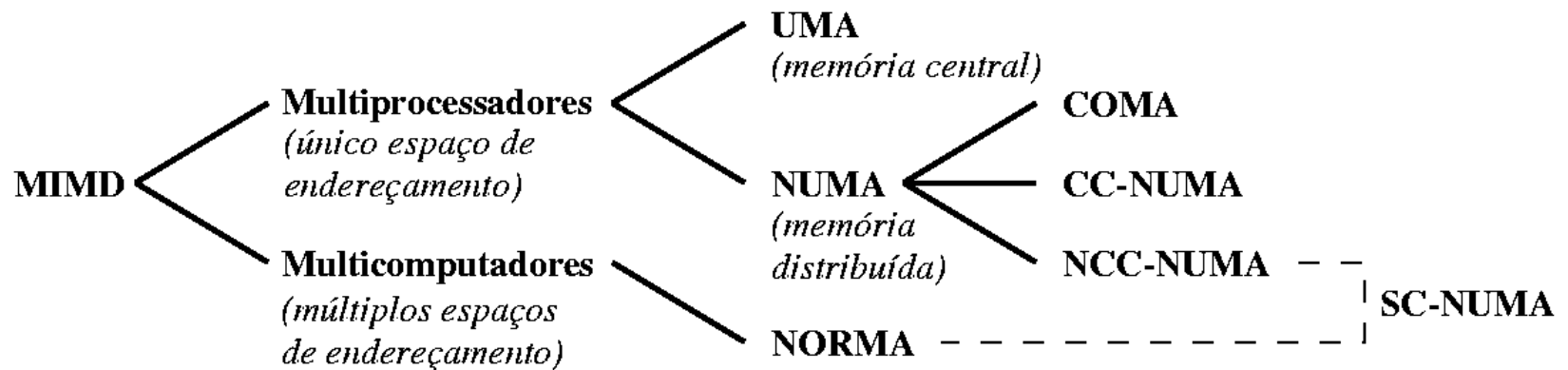
Classificação Segundo o Tipo de Acesso à Memória

- Multicomputadores
 - NORMA (*non-remote memory access*)
 - Apenas acesso local à memória



Plataformas tradicionais para PPD

- PVP - Processadores Vetoriais
- SMP - **Multiprocessadores** Simétricos com memória compartilhada
- MPP - **Multicomputadores** Massivamente Paralelos com **múltiplas memórias locais**
- NOW - Redes de Estações de Trabalho
- COW – Cluster of Workstations

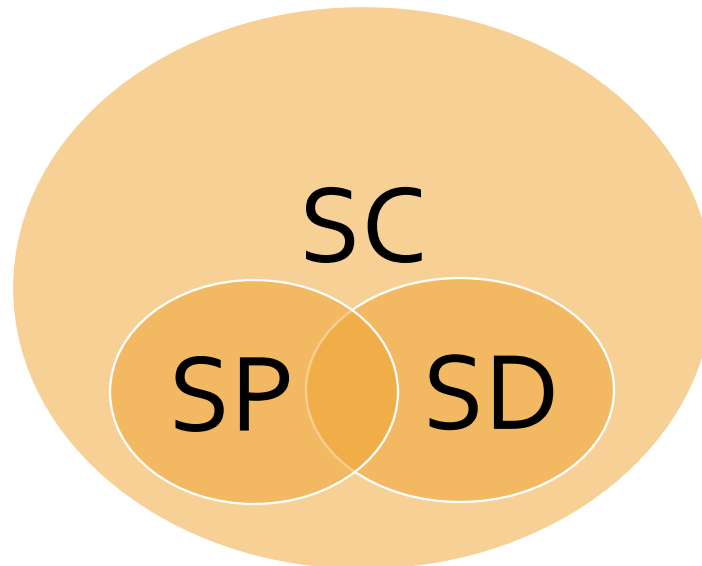




Diferenças e semelhanças entre PP e PD

Sistemas Concorrentes

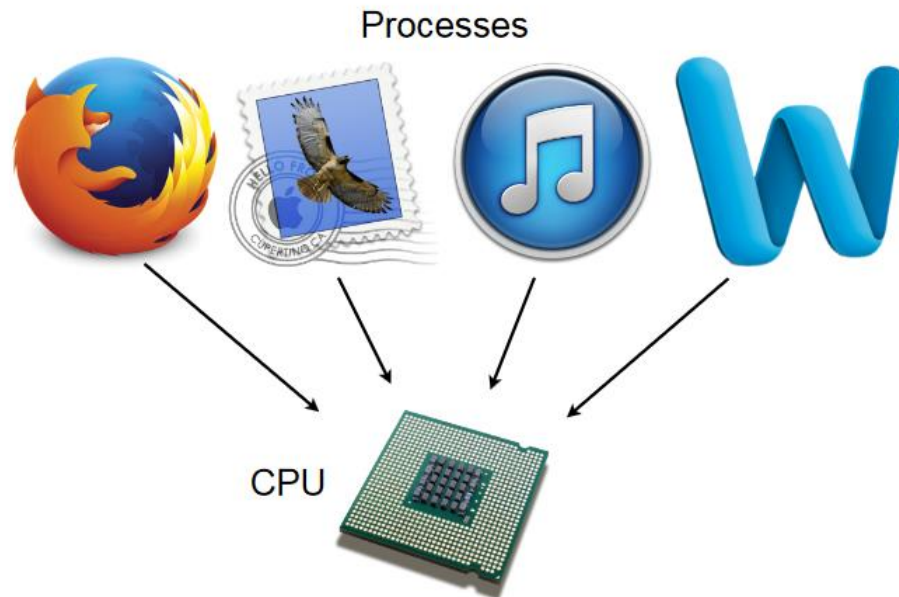
- Tanto SP quanto SD são sistemas concorrentes
 - Compartilham características comportamentais
 - Mas tem objetivos diferentes!



Sistema Concorrente

- Concorrência:
“ability of different parts or units of a program, algorithm, or problem to be executed out-of-order or at the same time simultaneously, without affecting the final outcome”
**Lamport*
- Objetivo principal
 - Melhor aproveitamento dos recursos e ganho de responsividade

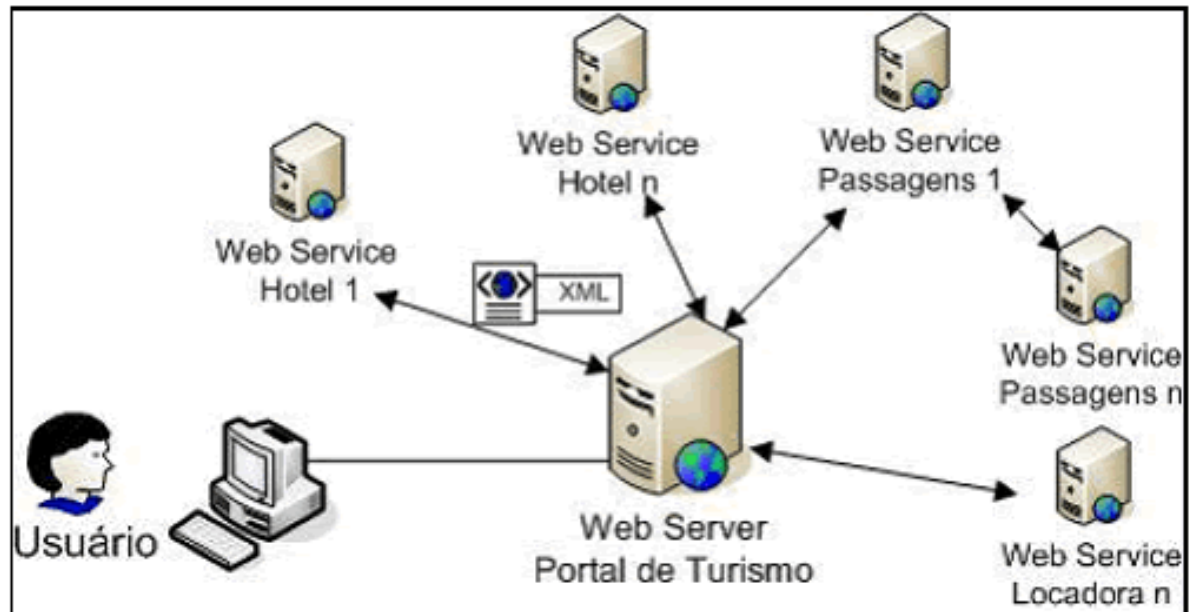
Sistema Concorrente



Sistema Distribuído

- Sistema Distribuído*:
"Coleção de computadores independentes entre si que se apresenta ao usuário como um sistema único e coerente"
**Tanenbaum*
- Objetivo principal
 - Escalabilidade e Tolerância a Falhas
- Objetivos secundários
 - Compartilhamento de Recursos
 - Transparência

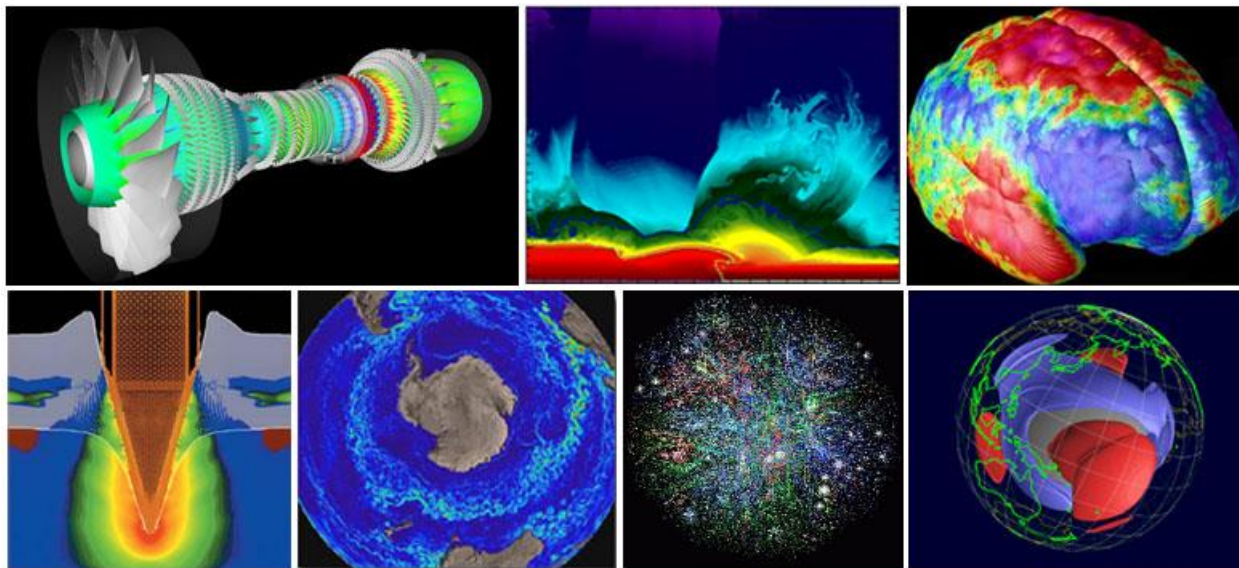
Sistema Distribuído



Sistema Paralelo

- Programação Paralela:
"Divisão de um problema em partes a fim de que estas partes possam ser executadas em paralelo em hardware dedicado para que o problema seja resolvido em menos tempo"
**De Rose*
- Objetivo principal
 - Ganho de Desempenho
- Objetivos secundários
 - Tolerância à Falhas

Sistema Paralelo

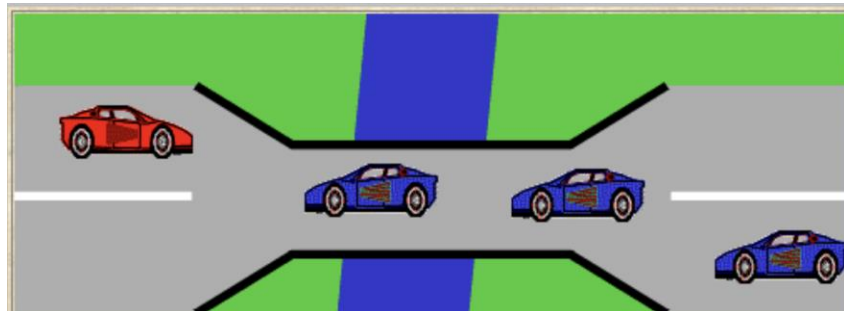




Métricas para Avaliação Destes Sistemas

Sistemas Concorrentes

- Separação (modelagem)
- Responsividade
- Corretude
 - Safety
 - não vai acontecer nada de ruim
 - Liveness
 - Eventualmente vai acontecer algo de bom



Sistemas Distribuídos

- Tempo de resposta
 - Latência (cliente)
- Transações por segundo
 - Provedor do serviço
- Tolerância a falhas
- Segurança

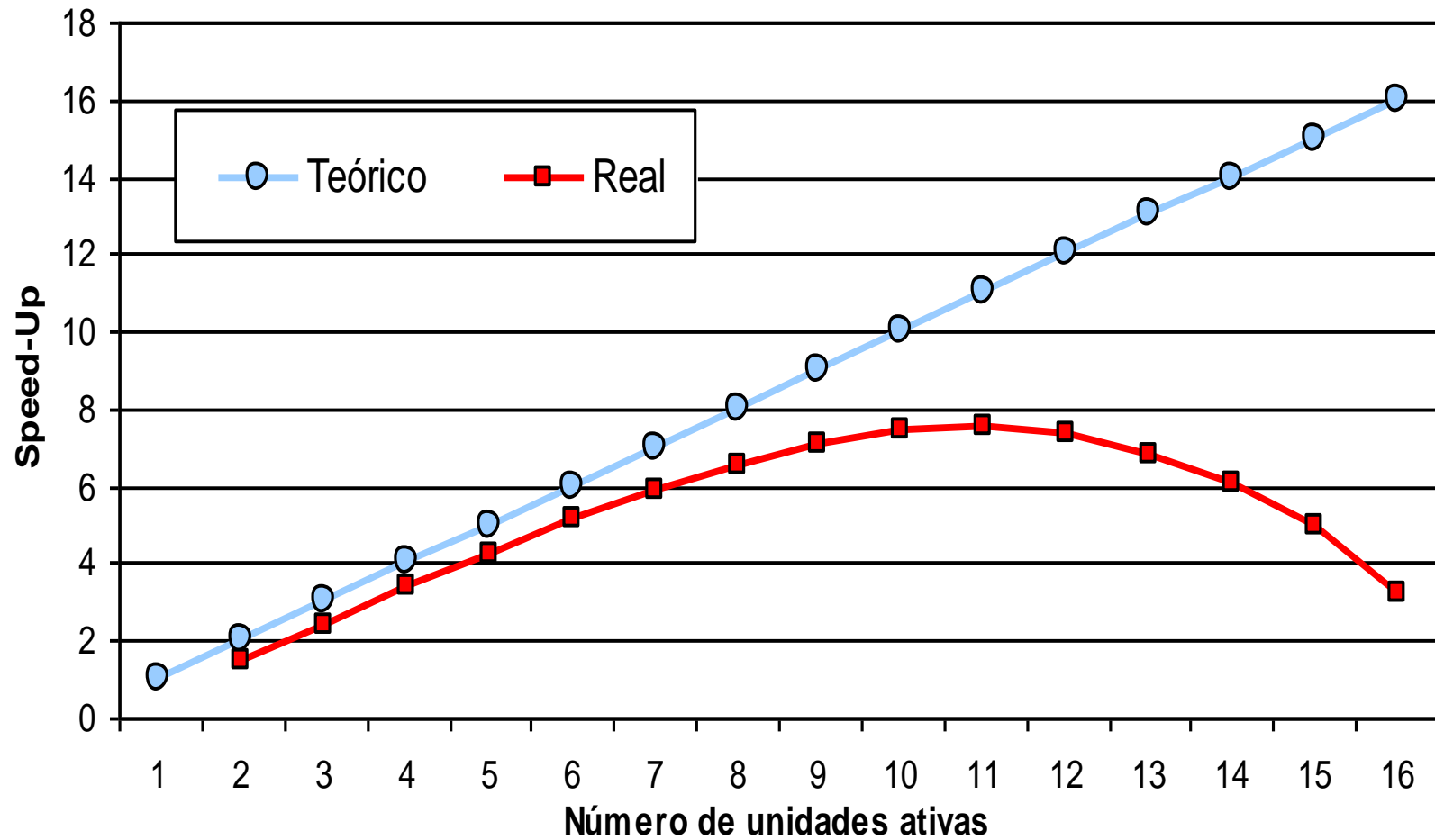
Sistemas Paralelos

- Fator de aceleração
 - Speed-up
- Eficiência

Fator de Aceleração (Speed-Up)

- Indica quantas vezes o programa paralelo ficou mais rápido que a versão sequencial
- É calculado pela razão entre o melhor tempo sequencial e o tempo da versão paralela)
- $SU_p(w) = \frac{T(w)}{T_p(w)}$
- Onde p é o número de unidades ativas utilizadas e w o trabalho que foi calculado

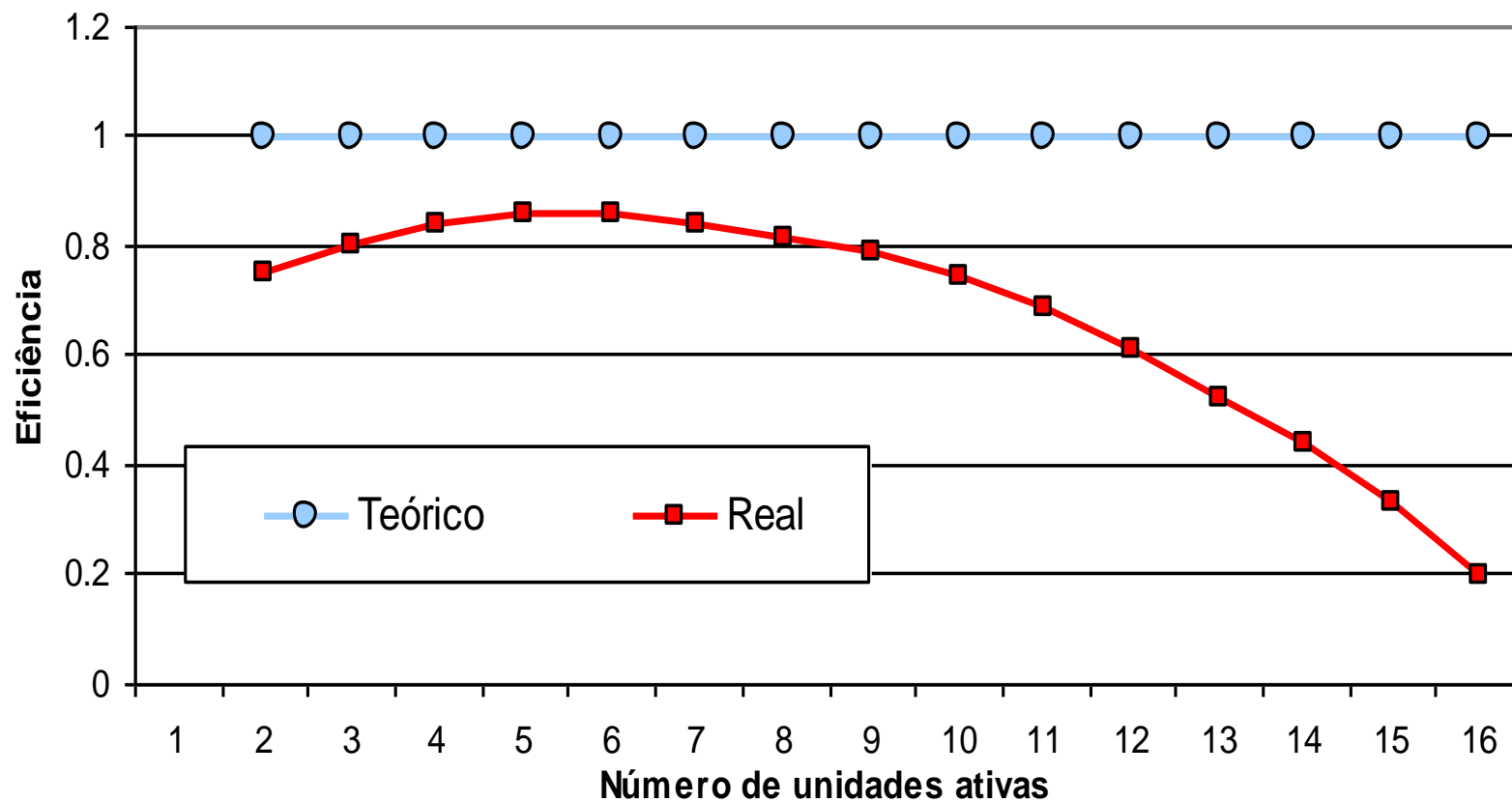
Speed-Up (Fator de Aceleração)



Eficiência

- Indica como foi a taxa de utilização média das unidades ativas utilizadas
- Mostra se os recursos foram bem aproveitados
- É calculado pela razão entre o Speed-Up e o número de unidades ativas utilizadas
- $$E_p(w) = \frac{SU_p(w)}{p}$$
- Onde p é o número de unidades ativas utilizadas e w o trabalho que foi calculado

Eficiência (Speed-Up/Nua)





Dúvidas?