Sistemas Distribuídos

Conceitos HW e SW

Edeyson Andrade Gomes
www.edeyson.com.br

Roteiro da Aula

Roteiro da Aula

- Conceitos de Hardware
- Conceitos de Software
- Combinações de SW e HW

Conceitos de Hardware

Conceitos de HW

- Múltiplas CPUs
  - Diferentes formas de organização do hardware
    - Interconexão
    - Comunicação

Conceitos de HW

- Esquemas de classificação
  - Sistemas com várias CPUs
  - Taxonomia de Flynn
    - Mais citada
    - Rudimentar
    - Características essenciais:
      - Número de fluxos de instruções
      - Número de fluxos de dados

Taxonomia de Flynn

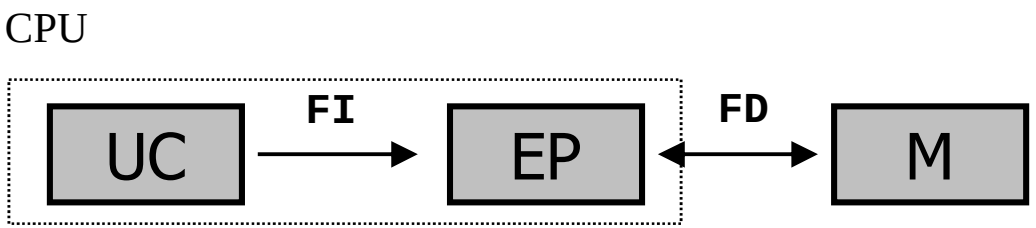
- **SISD** (Single Instruction, Single Data streams)
  - Computadores com um único processador
  - Microprocessadores tradicionais
  - Mainframes
- **SIMD** (Single Instruction, Multiple Data streams)
  - Processadores vetoriais
  - Supercomputadores
  - Uma instrução X dados em paralelo

Taxonomia de Flynn

- **MISD** (Multiple Instruction, Single Data streams)
  - Não existe computador que atenda esse modelo
- **MIMD** (Multiple Instruction, Multiple Data streams)
  - Grupo de computadores independentes
  - Sistemas paralelos e distribuídos

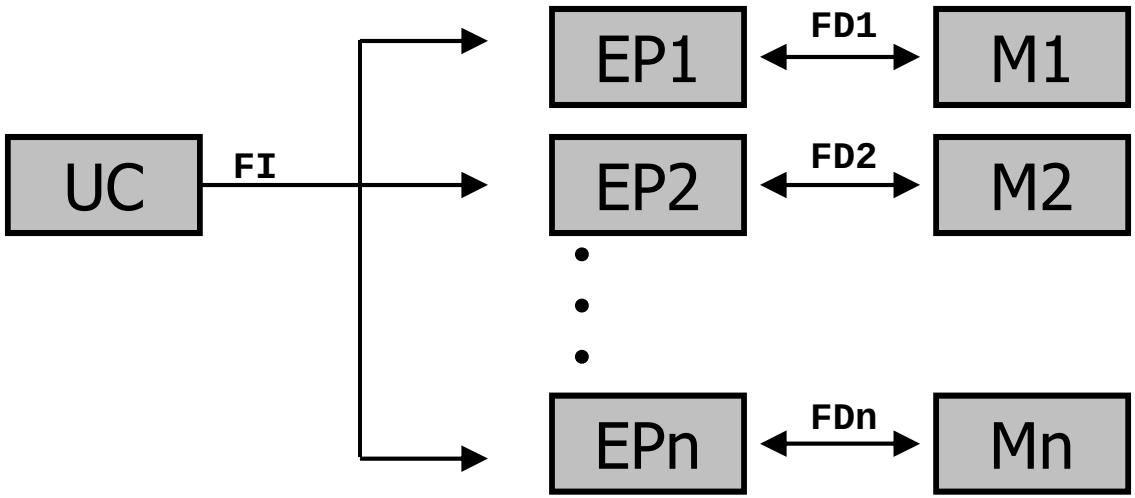
Modelos

- SISD



Modelos

- SIMD
  - Processadores Vetoriais

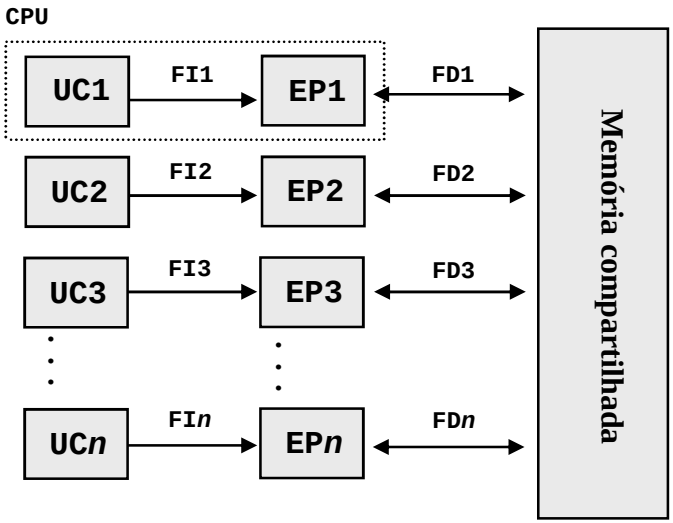


Modelos

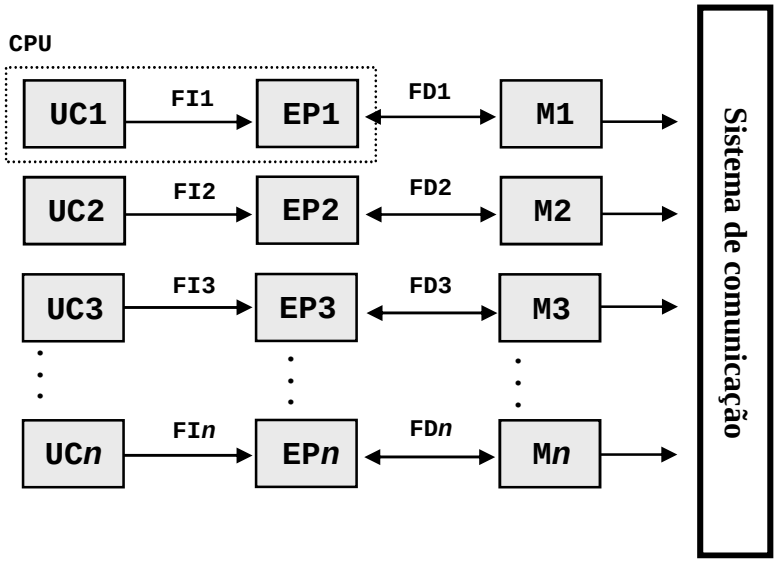
- MIMD
  - Multicomputadores
  - Multiprocessadores

Modelos

MIMD compartilhada



MIMD distribuída



MIMD

- Tipos de arquiteturas MIMD utilizadas
  - Multiprocessador Simétrico (SMP)
    - *Arquitetura MIMD com memória compartilhada*
  - Acesso Não-Uniforme à Memória (NUMA)
    - *Arquitetura MIMD com memória compartilhada*
  - Agregado de Computadores (*Cluster*)
    - *Arquitetura MIMD com memória distribuída*

Classificação quanto ao HW

- MIMD pode ser dividido em:
  - Multi-processadores
    - Memória compartilhada (pública)
      - Espaço de endereçamento único
  - Multi-computadores
    - Memória não compartilhada (privada)
      - Múltiplos espaços de endereçamento

Classificação quanto ao HW

- Arquiteturas de interconexão de rede:
  - Barramento (BUS)
    - Rede única, cabo, etc
      - Conexão de todas as máquinas
  - *Switch*
    - Conexão individual de máquina a máquina
    - Sistema de telefonia

Classificação quanto ao HW

- Multi-processadores baseados em barramento
  - várias CPUs
    - Barramento comum
      - Rápido
    - Memória única
    - Ex:
      - Motherboard com várias CPUs

Classificação quanto ao HW

- Barramento comum
  - Linhas de endereço, dados e controle
    - Leitura da memória:
      - CPU coloca endereço nas linhas de endereço
      - Ativa um sinal na linha de controle correspondente
      - Memória responde com conteúdo nas linhas de dados

Barramento Comum

- Memória Coerente
  - CPU A escreve
  - CPU B lê (Tb > Ta)
    - CPU B lê dado da CPU A

- Congestionamento no barramento gera queda no desempenho
  - Sobrecarga (várias CPUs)

Barramento Comum

- Desempenho melhorado com caches privadas
  - Cada CPU possui sua CACHE
    - Diminui acessos ao barramento
    - Número de CPUs pode crescer
    - Incoerência de Memória
      - Mesmo dado em caches diferentes
      - Alteração?

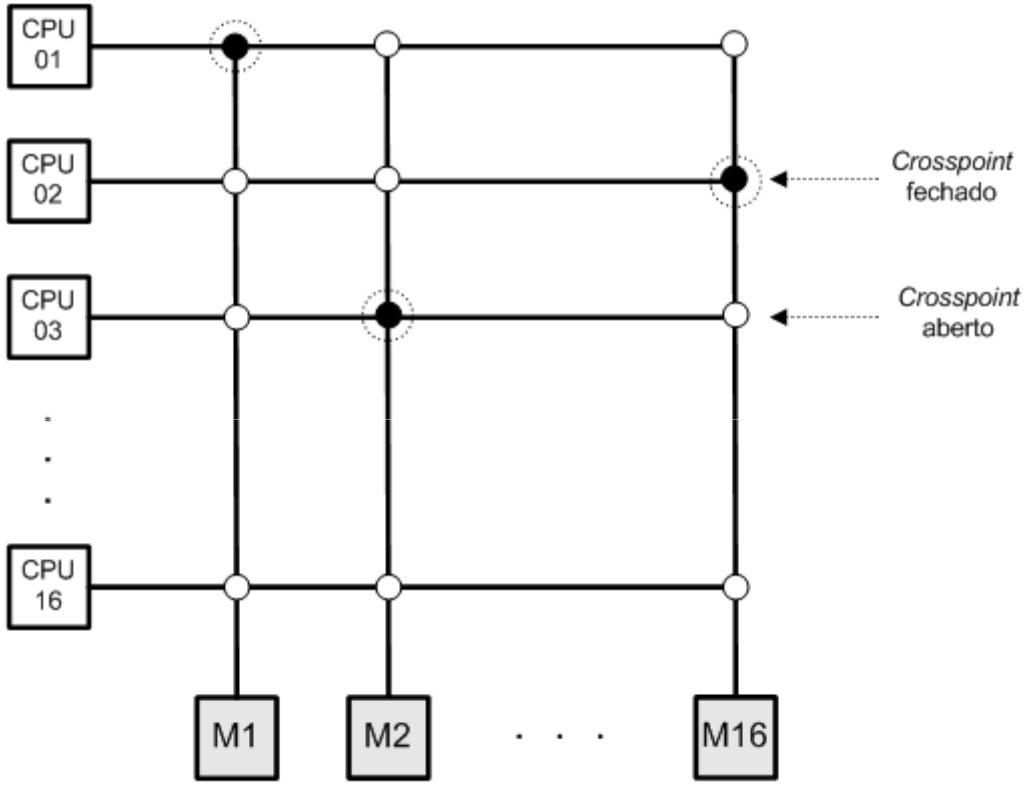
Gerência de Cache

- Gerência de Cache Write-Through
  - Leitura na Cache
  - Escrita na Cache e na memória
- Snoopy Cache
  - Monitorar o barramento
  - Escrita em endereço armazenado
    - Troca
- Snoopy Write-Through Cache
  - Coerente
  - Transparente

Switch

- Multi-processadores baseados em *Switch*
  - Suporte a multi-processadores com mais de 64 CPUs
- Crosspoint Switch
  - Estados
    - Aberto ou fechado
- Muitos processadores podem acessar diferentes módulos de memória ao mesmo tempo
  - Módulos têm acesso não simultâneo por várias CPUs

Switch



Sistema C.mpp: rede crossbar 16 × 16

(16 proc. PDP11 e 16 módulos de memória) (HWANG, 1993)

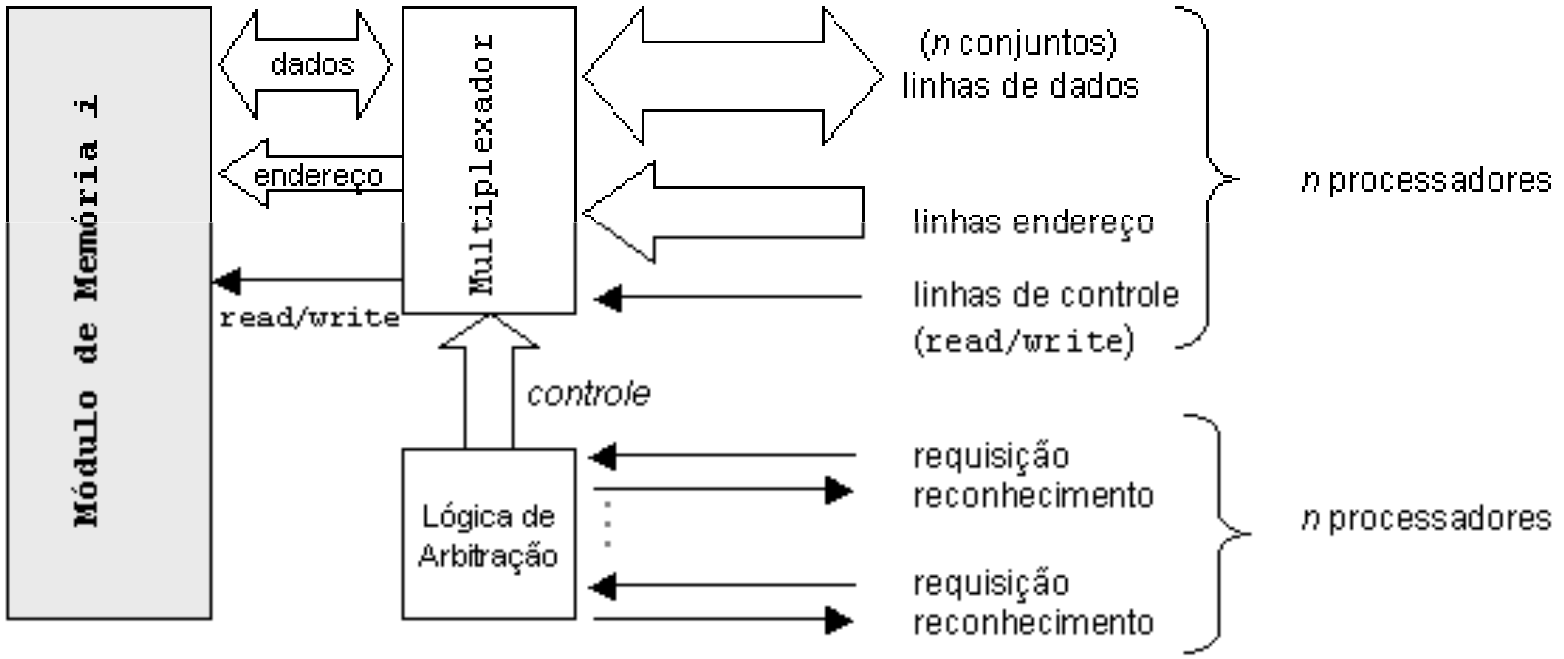
Switch

- Desvantagens
  - ***n*** CPUs e ***n*** módulos de memória
    - n^2 crosspoint switches
    - Custo proibitivo se ***n*** cresce

Switch

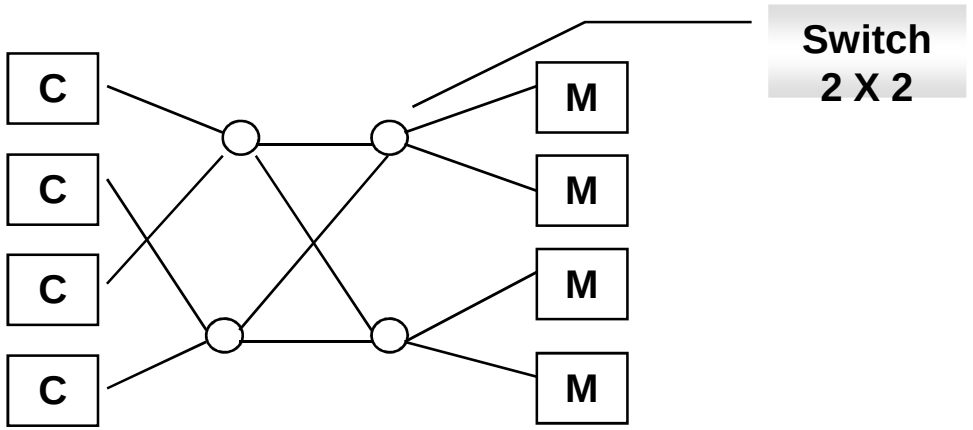
- Características
  - CPUs são ligadas aos módulos de memórias por meio de crosspoints (switches que são eletronicamente abertos/fechados para conectar um par processador-memória);
  - Acessos à memória em paralelo, desde que para módulos de memória distintos.

Switch

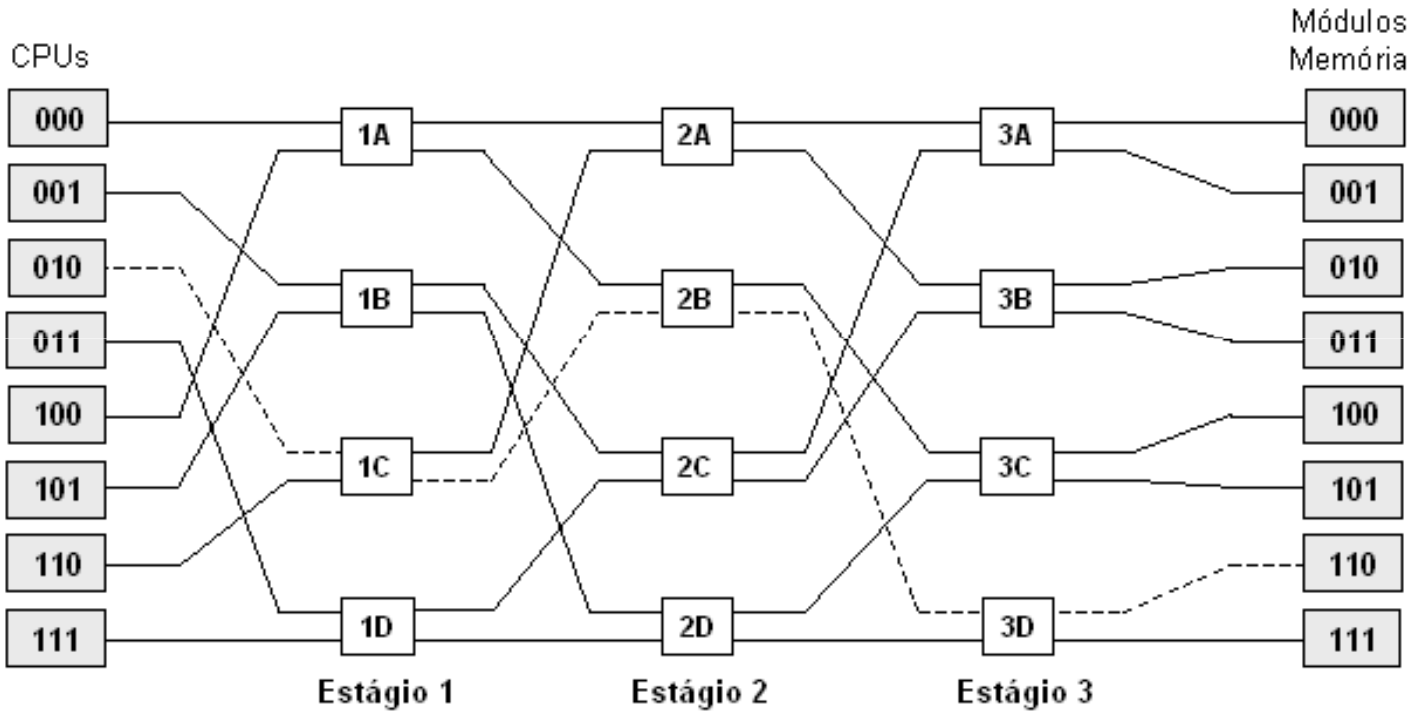

Módulo de Memória i
dados
endereço
read/write
Multiplexador
(n conjuntos)
linhas de dados
linhas endereço
linhas de controle
(read/write)
n processadores
controle
Lógica de
Arbitração
requisição
reconhecimento
requisição
reconhecimento
n processadores

Redes Omega

- Redes alternativas de *Switch*
  - *Exemplo:*
    - *4 switches 2 X 2 com 2 entradas e 2 saídas cada.*
    - *Toda CPU pode acessar qualquer módulo de memória*



Redes Multiestágio



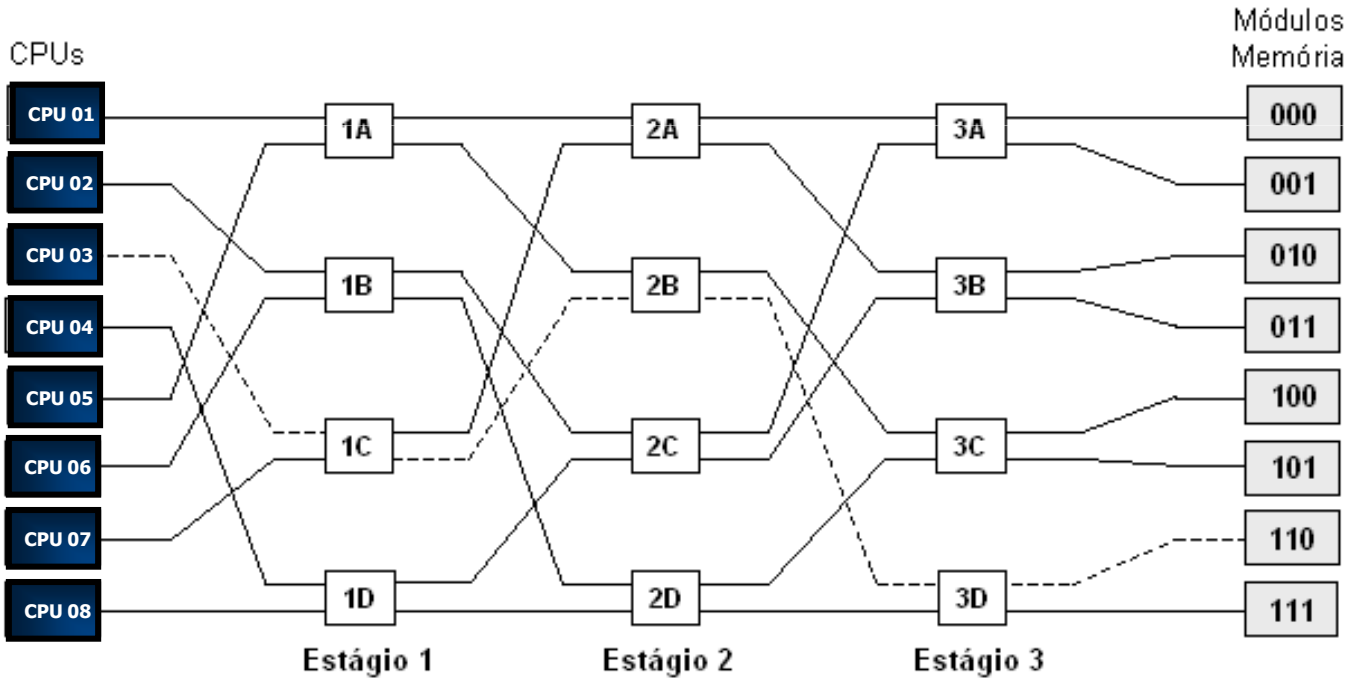
Rede Omega (switches 2x2)

Redes Omega

- Caso geral de Redes Omega
  - n CPUs e n módulos de memória
    - $Log_2 n$ estágios de switches
    - Cada estágio contém n/2 switches
    - Total: (n $log_2 n$) /2 switches
  - Performance
    - Número de estágios determina Overhead
    - Switchs mais rápidos
      - Maior Custo

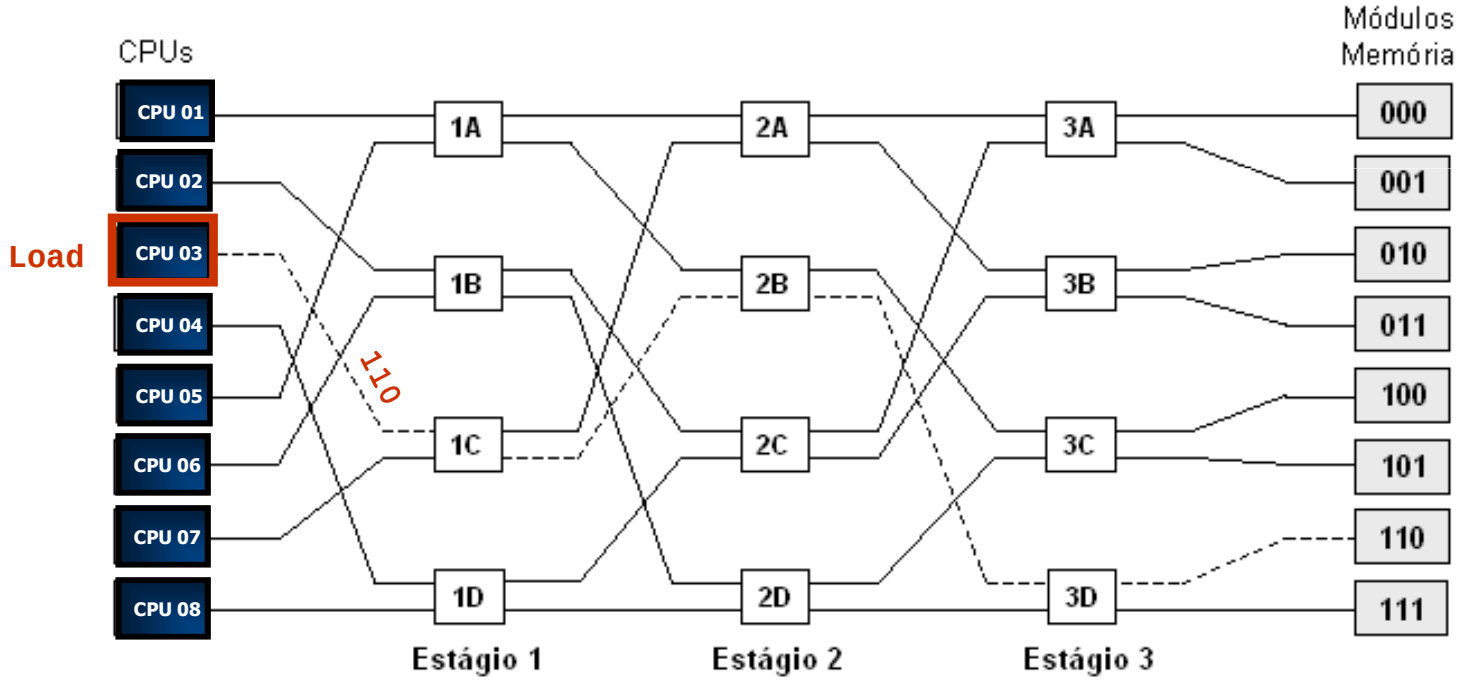
Redes Omega

Sistema com 512 MB de RAM (= endereços de 29 bits) divididos em 8 módulos de memória (= módulos de 64 MB). Portanto, os 3 primeiros bits especificam o módulo de memória, os outros 26 especificam um endereço dentro do módulo.



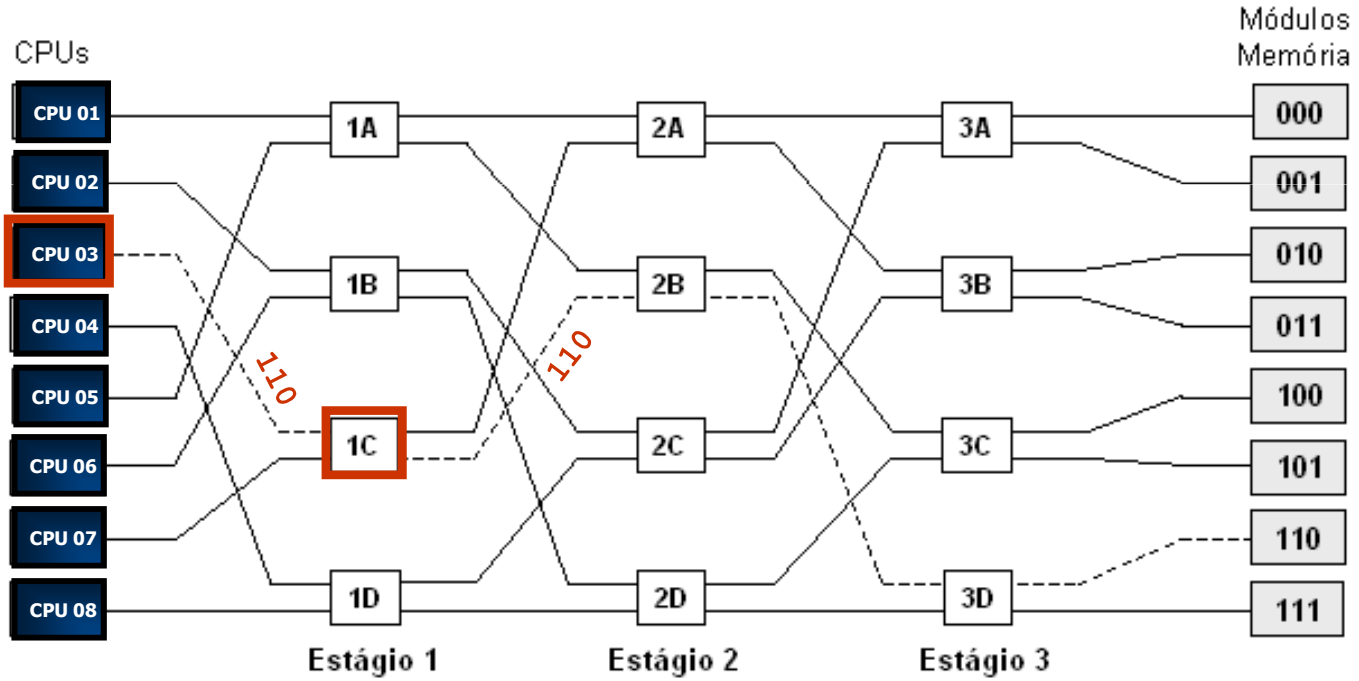
REDES OMEGA

CPU 03 executa instrução LOAD (110 00000000000000000000000001), i.e., para um endereço que está "contido" no módulo 110.



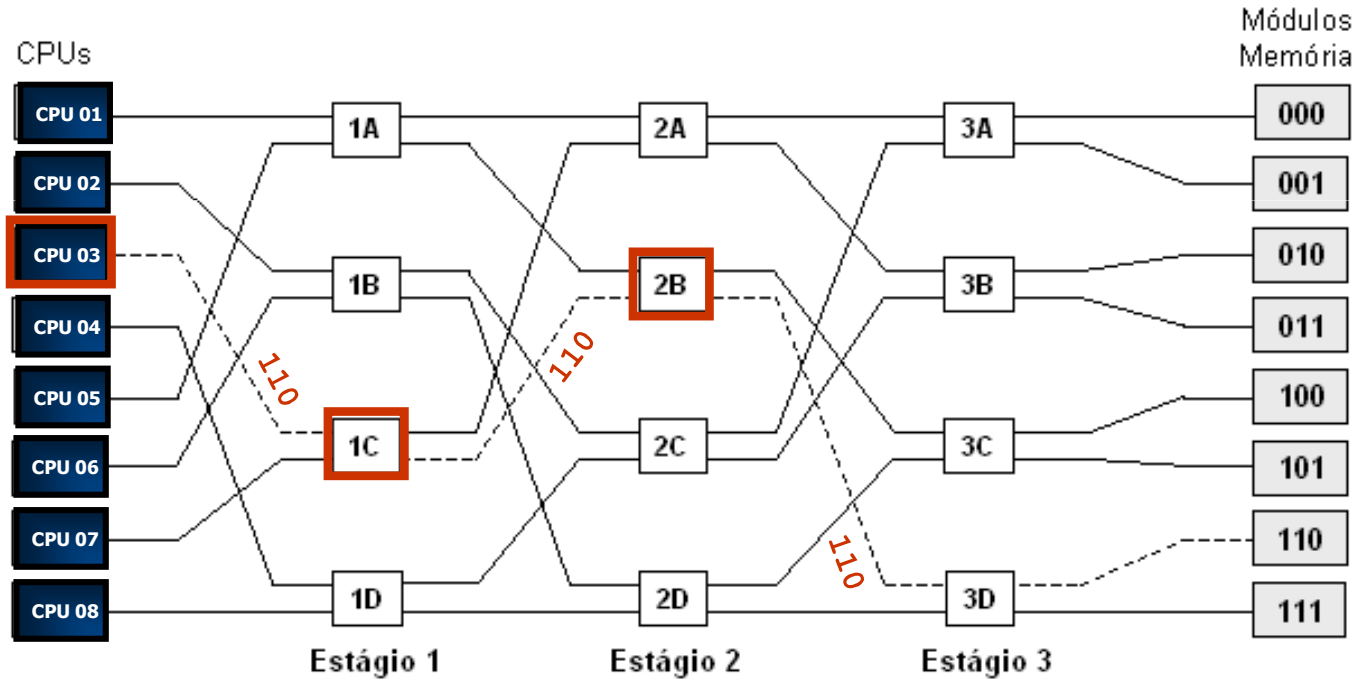
REDES OMEGA

O *switch 1C* verifica o 1º digito do campo módulo (um *switch* no 1º estágio da rede verifica o 1º dígito do campo *módulo*), neste caso 1. Um valor 1 indica que a requisição de leitura deve prosseguir pela saída inferior do *switch*, neste caso indo para o *switch 2B*.



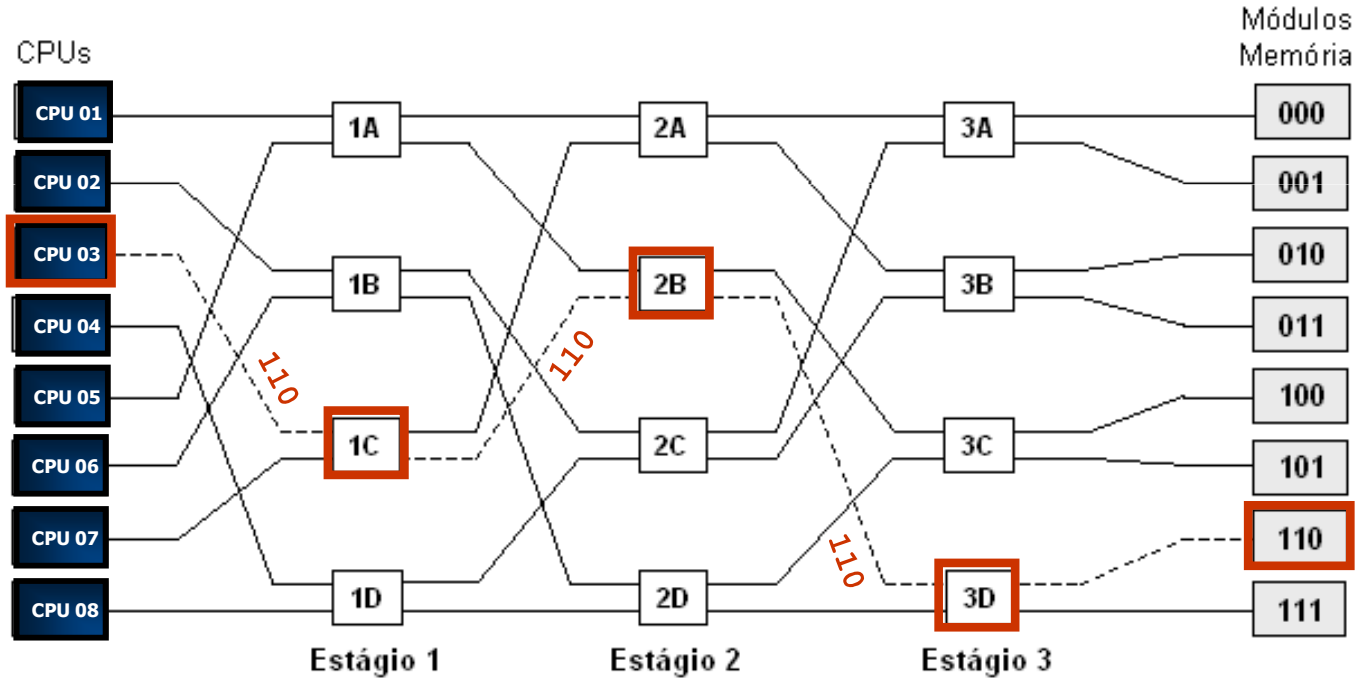
Redes Omega

O *switch 2B*, no estágio 2 da rede, analisa o segundo dígito do campo *módulo*, que também é 1, e a requisição prossegue através do *switch 3D*.



Redes Omega

O *switch 3D*, no estágio 3 da rede, verifica o terceiro dígito do campo *módulo,* que é 0 (zero). Um valor 0 indica que a operação deve prosseguir pela a saída superior do *switch*. Então, o módulo 110 é "alcançado".



Redes Omega

- Para a construção da rede anterior foram utilizados 12 *switches* (para uma rede $n \times n$ utilizam-se $log_2\ n$ estágios e $n/2$ *switches* por estágio, num total de $(log_2\ n) * n/2$ *switches*). Para a construção de uma rede crossbar seriam necessários n^2 switches.
- A desvantagem de uma rede multiestágio em relação à *crossbar*, é que nem sempre dois acessos podem ocorrer em paralelo, mesmo que sejam para módulos diferentes.

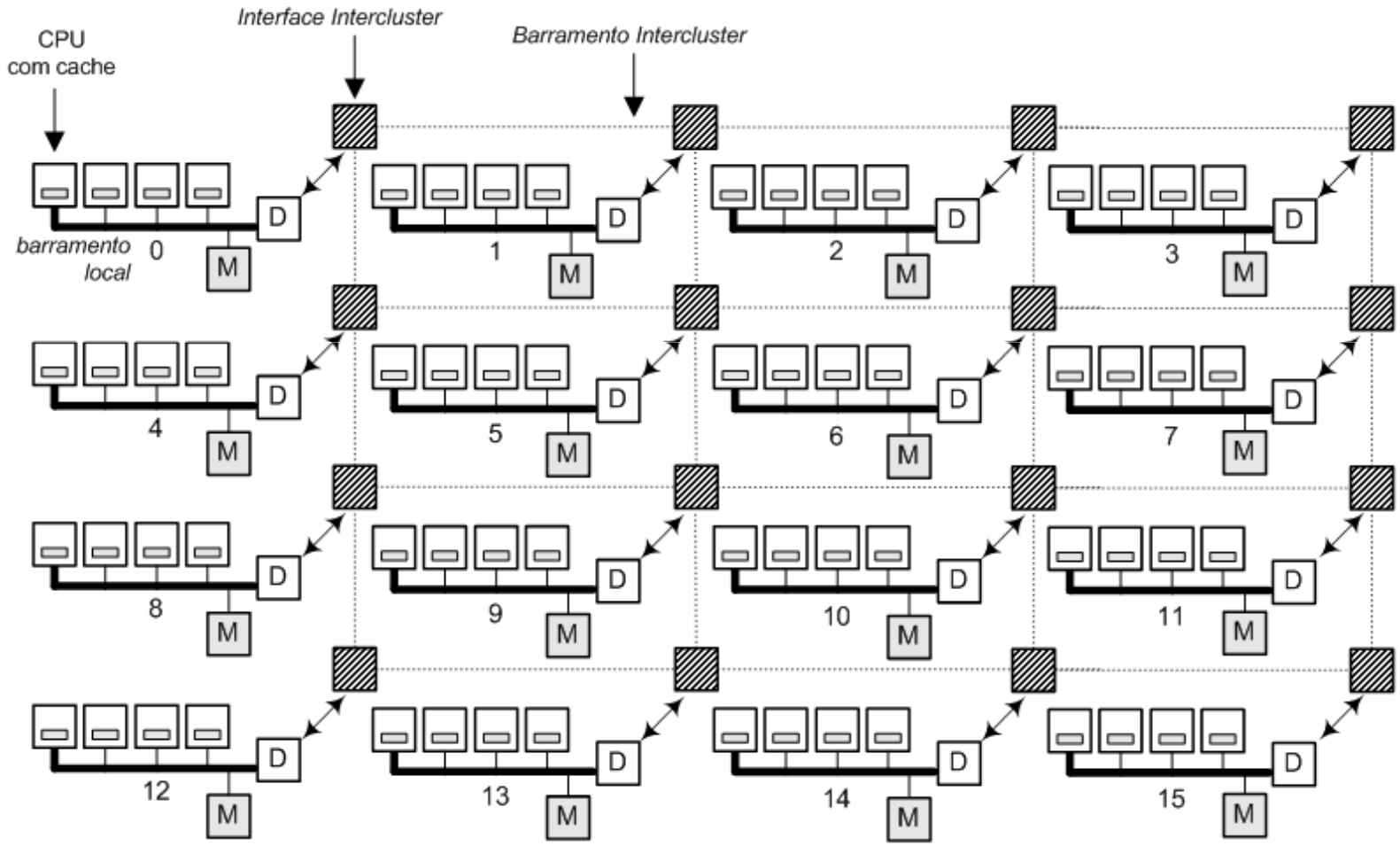
Sistemas Hierárquicos

- Redução de Custo
  - Cada CPU possui memória
    - Acesso local rápido e em outra CPU é lento
    - NUMA (NonUniform Memory Access)
  - Máquinas Numa
    - Tempo de acesso médio melhor que redes Omega
    - Maior complexidade
      - Alocação de programas, dados, etc.
      - Maximizar acesso local

NUMA

- Assim como em SMPs, neste tipo de arquitetura existem dois ou mais processadores, que compartilham uma memória global (= um único espaço de endereçamento).
- Em um sistema NUMA os processadores são organizados em nós.
- Cada nó possui 1 ou mais processadores, com sua(s) própria(s) memória(s) cache (um, dois, ou mais níveis) e alguma memória principal conectados por um barramento ou outro sistema de interconexão.

NUMA


CPU
com cache
Interface Intercluster
Barramento Intercluster
barramento
local
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
D
M

NUMA

- A **principal característica** de uma arquitetura NUMA é o acesso não uniforme à memória, ou seja, embora todos os processadores possam acessar todas as posições de memória, os tempos de acesso variam de acordo com o endereço acessado.

NUMA

- O acesso a uma posição de memória local (memória no mesmo nó do processador que está realizando o acesso) é mais rápido do que o acesso a uma posição de uma memória remota.

- Assim, o sistema operacional de uma máquina NUMA deveria, sempre que possível, escalonar as *threads* de um processo entre os processadores do nó da memória usada pelo processo.

NUMA

- O programador usando uma arquitetura NUMA também deveria explorar a localidade.

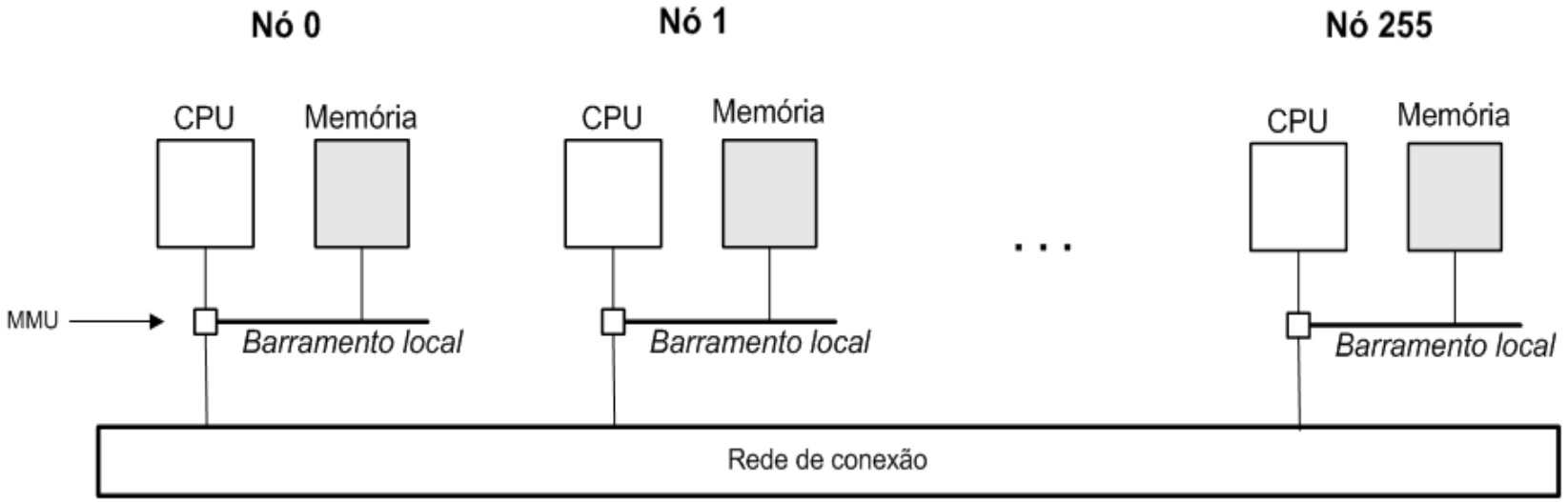
- O sistema operacional Windows, por exemplo, oferece uma série de funções através da API Win32 para programação em sistemas NUMA.

NUMA

- Existem dois tipos de arquiteturas NUMA
  - NC-NUMA
    - *NUMA que não utiliza cache*
  - CC-NUMA (Cache Coherent NUMA)
    - *NUMA com cache (e coerência de cache)*

NC-NUMA

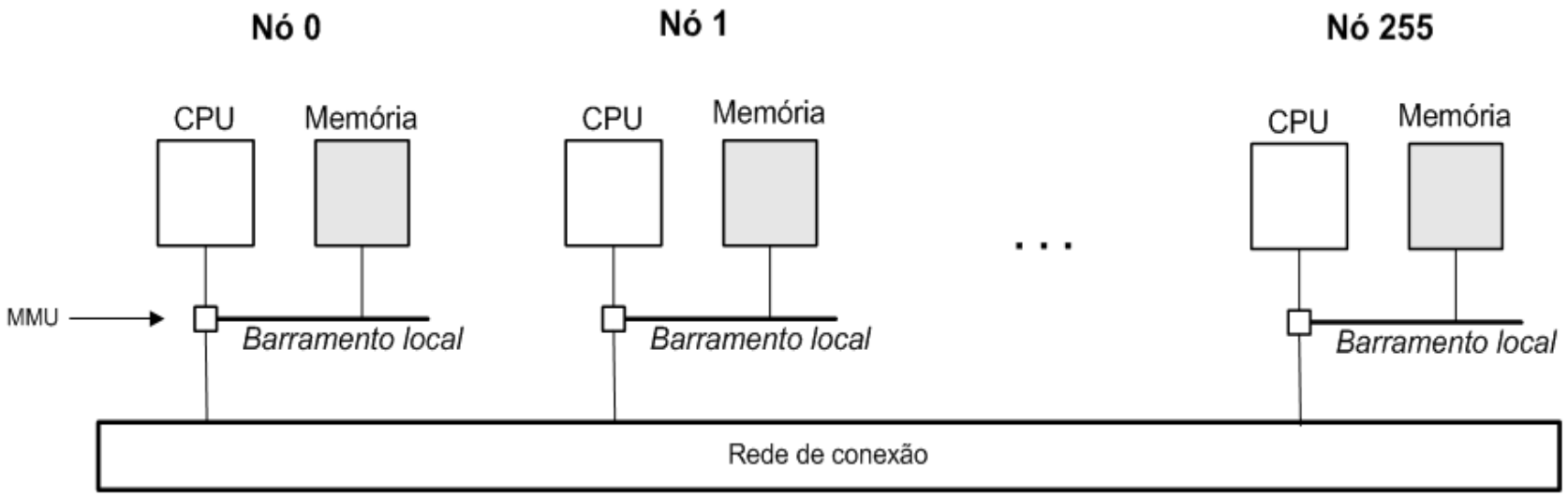
Uma das primeiras máquinas NC-NUMA foi a Carnegie-Mellon Cm. Era composta por uma coleção de CPUs LSI-11 (uma versão em chip único do PDP-11), cada uma com uma memória acessível diretamente através de um barramento local. Os nós eram interconectados por um barramento de sistema.



NC-NUMA Carnegie-Mellon Cm

NC-NUMA

A MMU em cada referência à memória verificava se o endereço pertencia à memória local. Se sim, a requisição era feita à memória local, via barramento local. Caso contrário era feita ao nó remoto, através do barramento de sistema.



NC-NUMA Carnegie-Mellon Cm

NC-NUMA

- A coerência dos dados em máquinas NC-NUMA é garantida porque não existe *caching*;
- Porém, um dado no lugar "errado" gera muitas penalidades. Se forem feitas três referências seguidas a uma posição de memória remota, são necessárias três buscas através do barramento de sistema.

NC-NUMA

- Para amenizar este problema, usam-se esquemas implementados em software. Por exemplo: um daemon pode fazer estatísticas de uso do sistema.
- Se for verificado que uma página parece estar no lugar “errado”, ele remove esta página da memória para que da próxima vez ocorra uma page fault ela possa ser alocada em um novo nó.

CC-NUMA

- Um sistema NUMA com coerência de cache é chamado CC-NUMA. A existência de *cache* requer algum protocolo de coerência.
- O método mais popular para a coerência de cache em sistema CC-NUMA é baseado no conceito de diretório.
- O diretório funciona como uma espécie de banco de dados que indica a localização das várias porções de memória, assim como o *status* das caches.

CC-NUMA

A figura abaixo ilustra um sistema CC-NUMA com 256 nós, 1 processador por nó, 16 MB por nó (256 x 16 = 4GB RAM = endereços de 32 bits). A memória dos nós é acessada em blocos de 64 bytes.

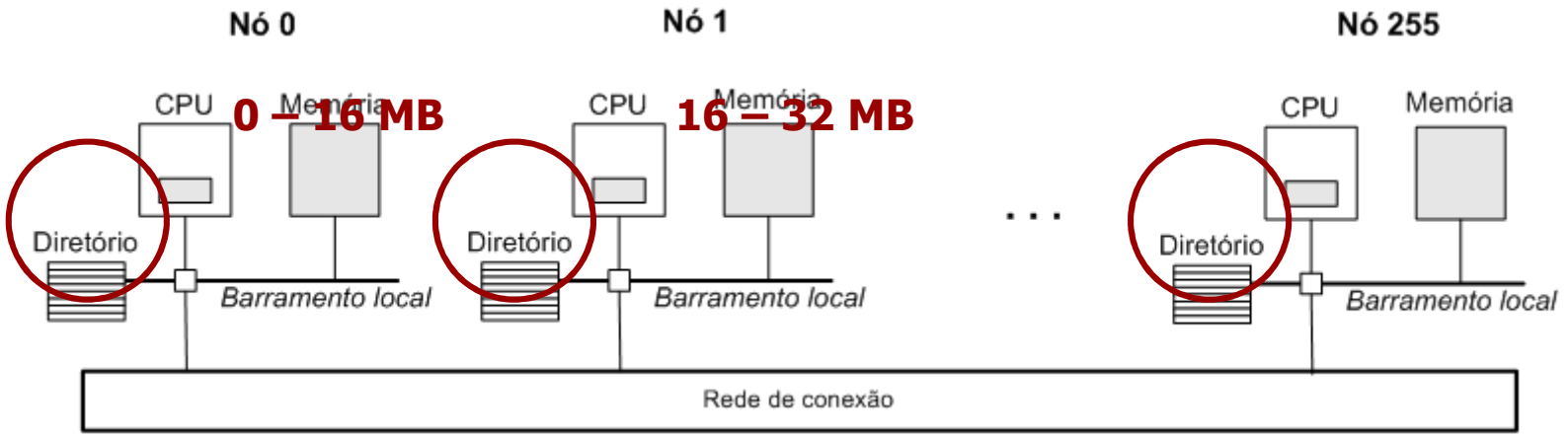
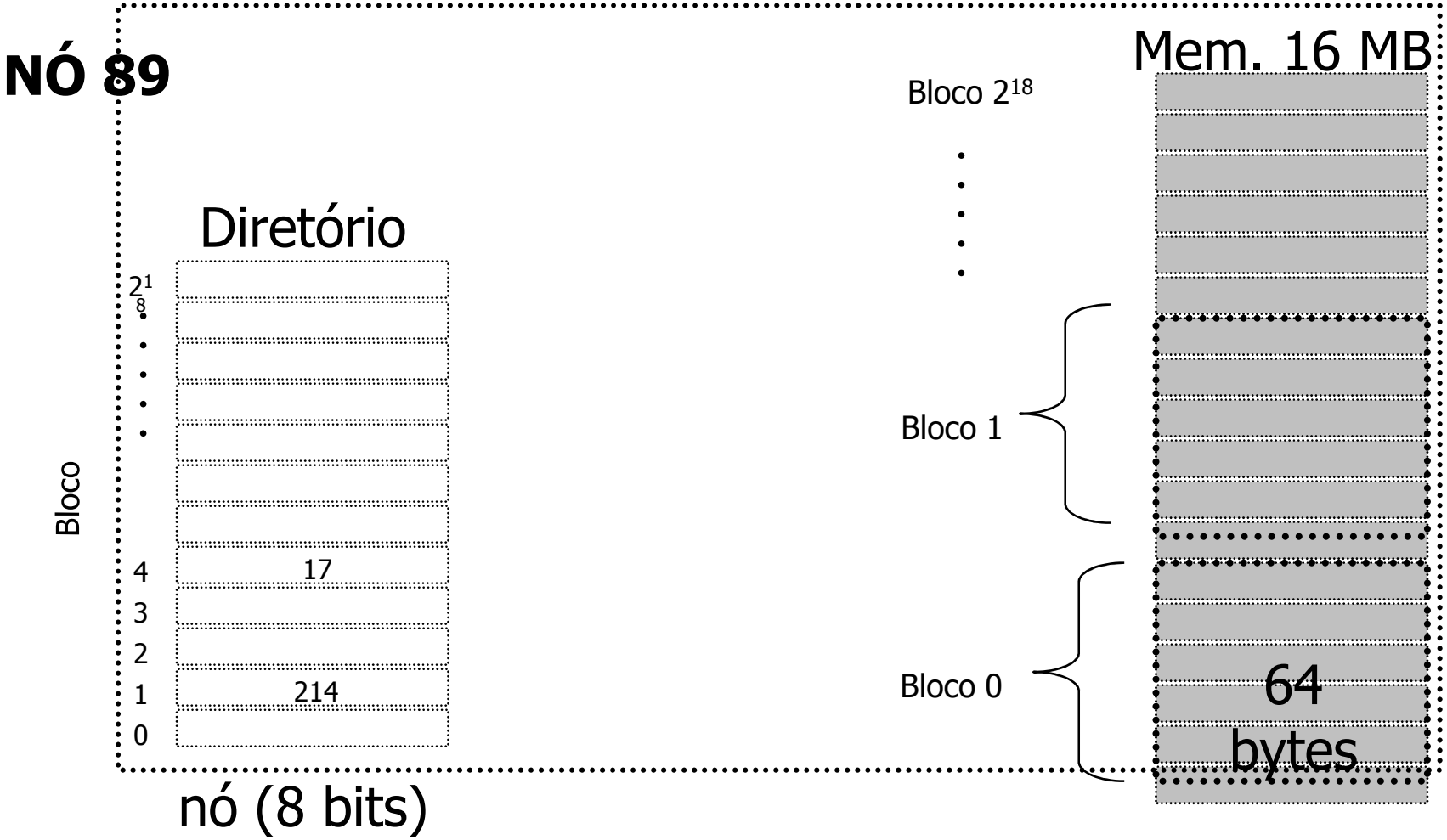


Diagrama de um sistema CC-NUMA

CC-NUMA

Coerência baseada em diretórios



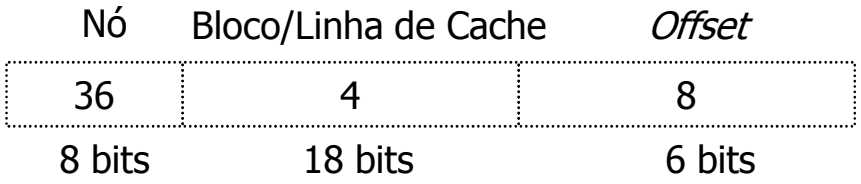
CC-NUMA

Coerência baseada em diretórios

(1) CPU 20 executa a instrução

```
LOAD 0x24000108
```

(2) MMU converte endereço lógico em físico.



(3) Verifica sua memória cache

(4) Não encontrando envia requisição para nó 36 solicitando bloco 4

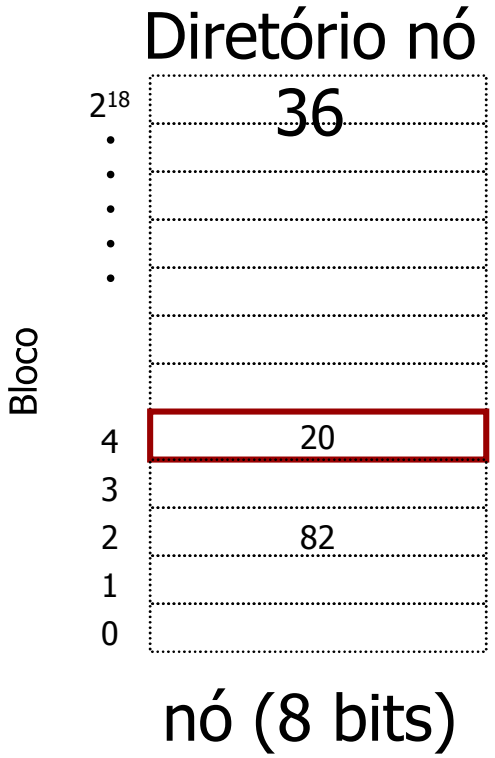
CC-NUMA

Coerência baseada em diretórios

(5) Hardware do nó 36 verifica diretório

(6) Entrada do bloco 4 vazia, portanto, ele não se encontra em cache

(7) Hardware do diretório 36 envia o bloco 4 à CPU 20 e atualiza o diretório



Problema de Custo

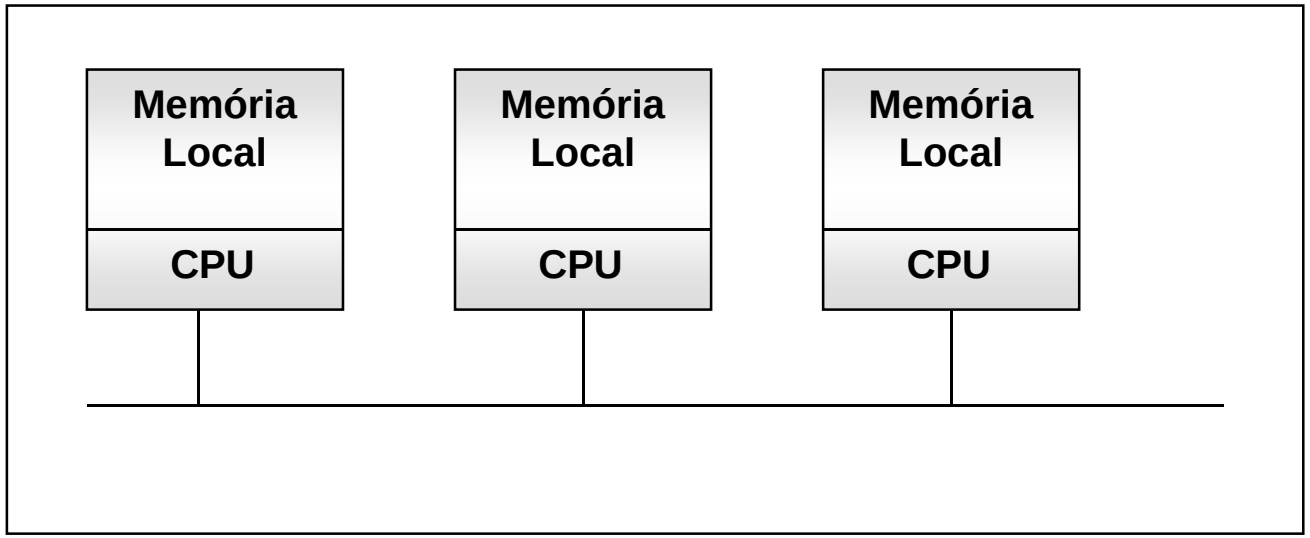
- Construção de grandes multi-processadores
  - Fortemente acoplados
  - Memória Compartilhada
    - Difícil e Caro

- **Solução?**
  - **Multi-computadores**

Multi-computadores

- Baseados em Barramento
  - Memória Privada
    - Cada CPU possui sua memória local
  - Comunicação CPU-CPU
    - Menor tráfego que CPU-Memória
    - Requisitos de performance
      - Rede local (10-100 Mbps 1Gbps)
      - Barramento

Multi-computadores



Multi-computadores

- Baseados em Switch
  - Malhas
    - Problemas de natureza bidimensional
      - Teoria dos Grafos
      - Crescimento dos caminhos
        - Raiz quadrada do número de CPUs

Multi-computadores

- Baseados em Switch
  - Hipercubos
    - Cubo n-dimensional
      - Cada CPU com n conexões para outras CPUs
      - Crescimento logarítmico
      - Conexão de CPUs próximas (vizinhas)
    - Caminho das mensagens
      - Menor que com Malhas
      - Logarítmico
    - Milhares de CPUs

Conceitos de Software

Conceitos de Software

- Software fracamente acoplado
  - Máquinas e usuários independentes entre si
  - Interação limitada
    - Sistema Distribuído
      - Grupo de computadores (PCs)
      - Rede Local
      - Compartilhamento

Conceitos de Software

- Software fracamente acoplado
  - Falha na rede
    - Computadores individuais operam com falta dos recursos compartilhados
      - Funcionalidades:
        - Arquivos
        - Impressoras

Combinações de SW e HW

Combinações de SW e HW

- SW fracamente acoplado em HW fracamente acoplado (multi-computador)
  - Combinação mais comum
  - Sistema operacional de rede
    - Máquinas com alto grau de autonomia
    - Poucos requisitos de sistemas
      - Heterogeneidade
    - Estações de trabalho + LAN

Combinações de SW e HW

- SW fracamente acoplado em HW fracamente acoplado (multi-computador)
  - Cada estação tem seu próprio Sistema Operacional
  - Execução local de comandos e programas
  - Conexão remota a outras máquinas
    - Terminais remotos

Combinações de SW e HW

- SW fracamente acoplado em HW fracamente acoplado (multi-computador)
  - Compartilhamento de dados através de sistema de arquivos distribuído
  - Comunicação mais conveniente
    - Compartilhamento de informação
    - Servidores de arquivos
      - Diferentes sistemas nos clientes
        - Visões dos arquivos?
        - Transparência
        - Padrão para troca de mensagens

Combinações de SW e HW

- SW fortemente acoplado em HW fracamente acoplado (multi-computador)
  - Meta:
    - "Ilusão" de que a rede é um sistema único, ao invés de uma coleção de computadores
      - Imagem de sistema único
      - Uniprocessador Virtual
    - Transparência
      - Multi-computadores são transparentes ao usuário

Combinações de SW e HW

- SW fortemente acoplado em HW fracamente acoplado (multi-computador)
  - Mecanismo global de comunicação entre processos
    - Comunicação local = comunicação remota
  - Esquema de proteção global
  - Igual gerência de processos
    - Interface para manipulação de processos
      - Criar, destruir, iniciar, parar,...
    - System Calls para ambiente distribuído
      - Não basta serem iguais
        - Mesma Interface

Combinações de SW e HW

- SW fortemente acoplado em HW fracamente acoplado (multi-computador)
  - Visão igual do sistema de arquivos
    - Restrições, nomenclatura, direitos, ...
  - Mesma interface para System Calls
    - Kernel idêntico nas diversas CPUs
      - Conseqüência lógica
    - Facilidade na coordenação de atividades
      - Cooperação entre os diversos kernels

Combinações de SW e HW

- SW fortemente acoplado em HW fortemente acoplado
  - Multi-processador de tempo compartilhado
  - Arquiteturas de propósito específico
    - Databases machines
  - Característica principal: fila única de tarefas
    - O escalonador executa numa região crítica, evitando que dois processadores escolham o mesmo processo simultaneamente

Combinações de SW e HW

- SW fortemente acoplado em HW fortemente acoplado
  - Sistema de arquivos simples
  - É comum o uso de uma memória cache única
  - Comunicação através de memória compartilhada
  - Cópia única do sistema operacional em execução

Conclusão