4. Arquiteturas MIMD

Máquinas MIMD (*Multiple Instruction Multiple Data*) são arquiteturas caracterizadas pela execução simultânea de múltiplos fluxos de instruções. Essa capacidade deve-se ao fato de que são construídas a partir de vários processadores operando de forma cooperativa ou concorrente, na execução de um ou vários aplicativos. Essa definição deixa margem para que várias topologias de máquinas paralelas e de redes de computadores sejam enquadradas como MIMD. A diferenciação entre as diversas topologias MIMD é feita pelo tipo de organização da memória principal, memória cache e rede de interconexão. Nas próximas seções serão detalhados os principais aspectos com relação às arquiteturas de máquinas MIMD.

4.1 Compartilhamento de memória

Quando vários elementos processadores são interconectados, a forma como cada um 'enxerga' a memória é decisiva para a definição dos modelos de comunicação.

Memória compartilhada (*shared memory*):
- o espaço de endereçamento é único;
- comunicação através de *load* e *store* nos endereços de memória.

Memória privativa (*multiple private address space*):
- o espaço de endereços é distinto para cada processador;
- comunicação através de troca de mensagens com operações *send* e *receive*.

Essa primeira diferenciação refere-se aos aspectos lógicos da memória. Dependendo do compartilhamento (lógico) de memória, teremos uma arquitetura MIMD do tipo **multiprocessador** ou **multicomputador**. Uma outra maneira de caracterizar as memórias é quanto aos aspectos físicos.

Memória distribuída (*distributed memory*)
- a memória é composta por vários módulos;
- cada módulo está próximo a um processador.

Memória centralizada (*centralized memory*)
- a memória se encontra a mesma distância de todos os processadores;
- pode ser implementada com um ou vários módulos.

4.2 Multiprocessadores

São as arquiteturas em que todos os processadores tem acesso ao mesmo espaço de endereços na memória. Assim, a comunicação entre processos é muito simples, bastando para tanto, utilizar operações do tipo *load* e *store*. Essa estrutura é semelhante à colocação de múltiplos processadores em uma máquina von Neumann tradicional. Os múltiplos

processadores são conectados à memória através de uma rede de interconexão. A Fig. 4.1 apresenta um modelo de arquitetura de um multiprocessador.

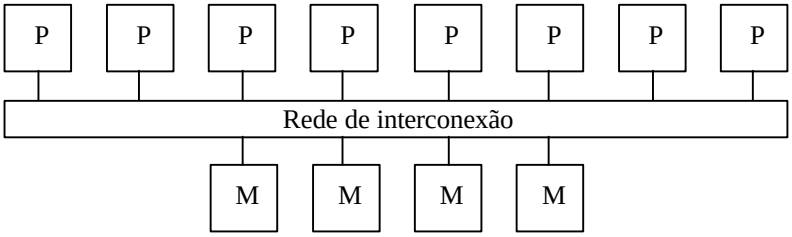


Figura 4.1 Arquitetura de um multiprocessador

Os multiprocessadores ainda podem ser classificados quanto a distância dos processadores à memória, e quanto aos esquemas de coerência de cache.

UMA (*Uniform Memory Access*)

Neste tipo de máquina, o tempo para o acesso aos dados na memória é o mesmo para todos os processadores a para todas as posições da memória. Essas arquiteturas também são chamadas de SMP (*Symmetric MultiProcessor*). A forma de interconexão mais comum neste tipo de máquina é o barramento e a memória geralmente é implementada com um único módulo. O principal problema com este tipo de arranjo é que o barramento e a memória tornam-se gargalos para o sistema, que fica limitado a uma única transferência por vez. A Fig. 4.2 mostra uma arquitetura do tipo SMP.

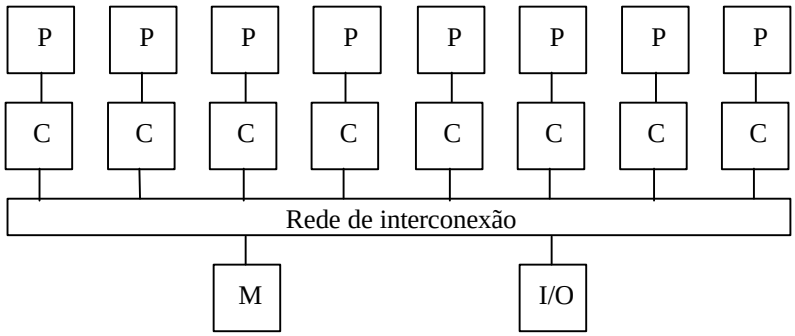


Figura 4.2 Arquitetura de um multiprocessador do tipo SMP (UMA)

Na Fig. 4.2 estão mostradas as memórias cache de cada processador. Essas memórias são utilizadas para esconder a latência no acesso à memória principal e para diminuir o tráfego no barramento. Como várias cópias de um mesmo dado podem ser manipuladas simultaneamente nas caches de vários processadores, é necessário que se garanta que os processadores sempre acessem a cópia mais recente. Esta garantia é chamada de coerência de cache (*cache coherence*), e máquinas UMA geralmente lidam com este problema diretamente em hardware. Um dos protocolos de coerência de cache mais populares é chamado de *snooping*, ou ‘bisbilhoteiro’. Neste caso, quando uma dado compartilhado por várias caches é alterado por algum processador, todas as demais cópias são invalidadas ou então atualizadas.

NUMA (*Non-Uniform Memory Access*)

Neste tipo de multiprocessadores, a memória geralmente é distribuída e portanto implementada com múltiplos módulos. Cada processador está associado a um módulo, mas o acesso aos módulos ligados a outro processador é possível. O espaço de endereçamento é comum a todos os processadores e a latência para ler ou escrever na memória pertencente a um outro processador é maior que a latência para o acesso à memória local. A Fig. 4.3 mostra a arquitetura de uma máquina NUMA.

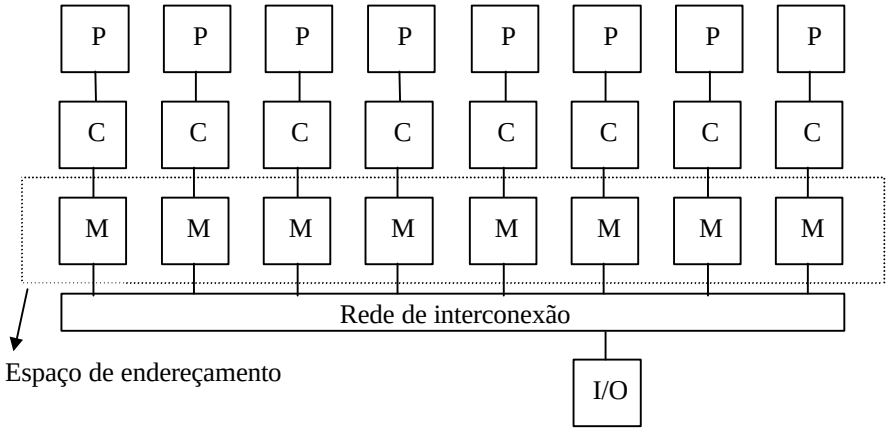


Figura 4.3 Arquitetura de um multiprocessador do tipo NUMA

As máquinas NUMA também estão sujeitas aos problemas de coerência de cache, e conforme a solução implementada existem variações deste tipo de arquitetura.

NCC-NUMA (*Non-Cache Coherent NUMA*)

Nesse tipo de máquina, não existe garantia de coerência nos dados da memória cache, ou simplesmente não existe cache.

CC-NUMA (*Cache Coherent NUMA*)

Nesse tipo de máquina, a coerência de cache é garantida pelo hardware.

SC-NUMA (*Software Coherent NUMA*)

Neste caso, a coerência de cache é garantida em software e essa implementação recebe o nome de DSM (*Distributed Shared Memory*). O espaço de endereçamento único é conseguido através de uma abstração implementada em software que pode tomar duas formas. Na primeira, os mecanismos de gerência de memória do sistema operacional são modificados para suprir as faltas de página ou segmento a partir da rede de interconexão (e não do disco). Quando um processador precisa de um dado em um espaço de endereçamento alheio, o sistema operacional encarrega-se de encontrar o mesmo e disponibilizá-lo para o processador. A grande vantagem é que as aplicações não precisam ser modificadas. Na segunda forma, alterações são feitas nos compiladores e bibliotecas de funções para que reflitam nas aplicações o modelo de memória utilizado. As aplicações são

então modificadas para a incorporação de primitivas de coerência, compartilhamento de dados e sincronização.

COMA (*Cache Only Memory Architecture*)

São multiprocessadores baseados em memórias cache de alta capacidade, em que a coerência é conseguida em hardware com a atualização simultânea em múltiplos nós dos dados alterados. Esse tipo de arquitetura é bastante complexo e faz com que estas máquinas tenham um custo elevado.

4.3 Multicomputadores

Essas máquinas caracterizam-se pelo fato de que cada processador enxerga somente a sua própria memória. Para a troca de mensagens e dados é preciso o envio de requisições através da rede de interconexão. Os multicomputadores também são chamados de sistemas de troca de mensagens (*message passing systems*). Com estas características, tais máquinas paralelas podem ser implementadas através de um conjunto de máquinas autônomas, ou seja, computadores tradicionais. A Fig. 4.4 mostra uma arquitetura de multicomputador.

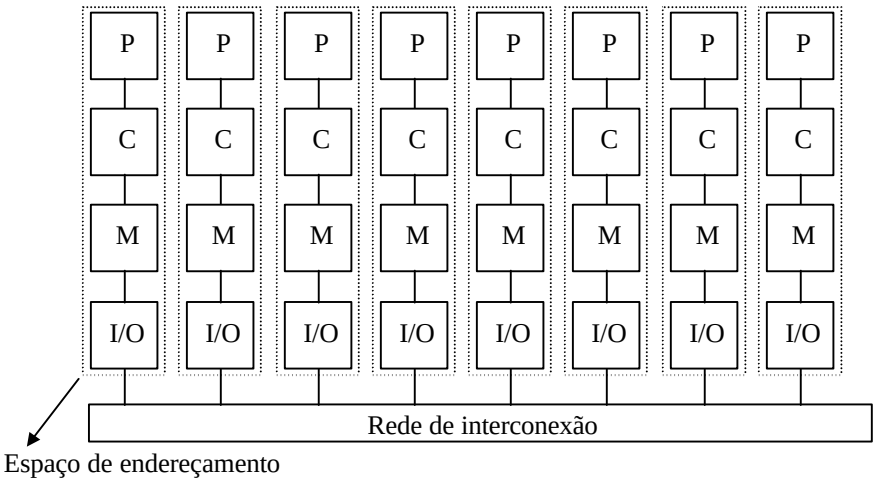


Figura 4.4 Arquitetura de um multicomputador

Quanto ao compartilhamento de memória, essas máquinas são classificadas como **NORMA** (*NO Remote Memory Access*). Multicomputadores podem ser divididos em duas categorias.

MPP (*Massively Parallel Processors*)

São multicomputadores compostos por um grande número de processadores, fortemente acoplados através de uma rede de alta velocidade. Geralmente são arquiteturas de custo elevado pois utilizam processadores específicos e redes de interconexão proprietárias.

COW (*Cluster of Workstations*)

Também chamadas de **NOW** (*Network of Workstations*), essas máquinas são construídas a partir de computadores comuns (PCs) ligados por redes de interconexão tradicionais.

4.4 Organização da memória principal

Como visto nas seções anteriores, a memória principal em uma máquina paralela desempenha um papel fundamental (principalmente em multiprocessadores). Existem várias formas de se fazer a conexão desta memória de maneira a privilegiar aspectos como custo, taxa de transferência e latência. A organização da memória em um módulo único é problemática devido à possibilidade de múltiplos acessos por vários processadores. O ideal seria a possibilidade de cada processador acessar todas as posições da memória sem a ocorrência de ciclos de espera. Uma forma de aumentar a capacidade de acessos simultâneos à memória é a implementação de memórias entrelaçadas (interleaved). O entrelaçamento permite que vários módulos operem em paralelo atendendo a requisições de vários processadores.

O entrelaçamento pode ser feito de 3 formas diferentes havendo entre os mesmos uma diferença na distribuição dos dados nos diversos módulos. Considere uma memória com 2^n endereços. No primeiro tipo de entrelaçamento (Fig. 4.5a), a parte mais significativa do endereço (com m bits) é utilizada para a especificação do módulo (2^m módulos) ao qual o restante do endereço se refere. No segundo tipo de entrelaçamento (Fig. 4.5b), a parte menos significativa do endereço (com m bits) é utilizada para a especificação do módulo (2^m módulos) ao qual o restante do endereço se refere.

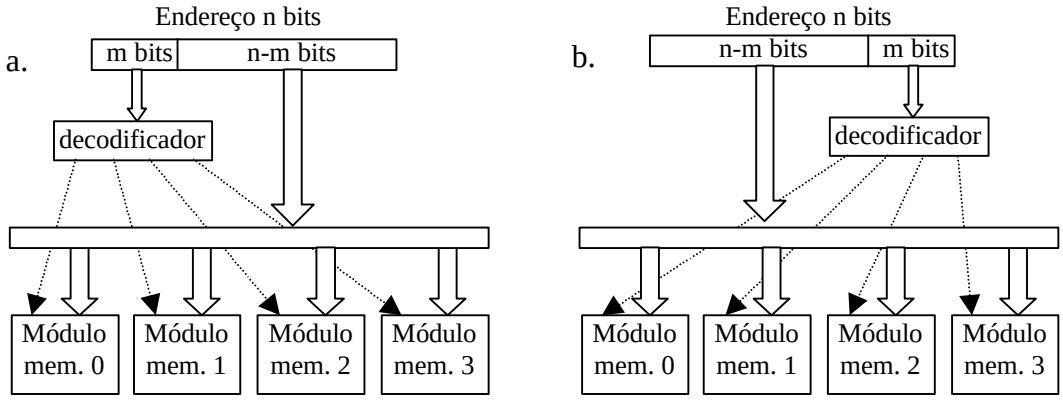


Figura 4.5 Seleção dos módulos por endereço mais (a) e menos (b) significativo

No entrelaçamento pela parte mais significativa dos endereços as principais características são as seguintes:

- facilidade de expansão da memória;
- a falha em um módulo afeta uma região específica de endereços;
- maior possibilidade de tráfego para um mesmo módulo em arquiteturas pipeline, vetoriais e multiprocessadores SIMD, devido a concentração dos dados em um mesmo módulo.

No entrelaçamento pela parte menos significativa dos endereços as principais características são as seguintes:

- os conflitos por acesso ao mesmo módulo ficam reduzidos pois os dados geralmente ficam distribuídos ao longo de vários módulos;
- a falha em um módulo afeta toda a memória do sistema.

Estas duas alternativas de entrelaçamento podem ser combinadas em uma terceira forma em que r bits da parte menos significativa do endereço e m-r bits da parte mais significativa são utilizados para a seleção do módulo. Através da escolha de r pode-se controlar o grau de tolerância as falhas e o grau de conflito nos acessos.

Além do entrelaçamento é preciso considerar o meio físico de conexão dos processadores com as memórias. Um barramento simples não permite a exploração do paralelismo no acesso às memórias e uma conexão mais adequada tende a ter alto custo, o qual aumenta com o aumento do número de processadores ou de módulos de memória. Uma alternativa para a conexão de vários processadores a várias memórias é a utilização de memórias multiportas, as quais incorporam circuitos para a operação com vários processadores. O resultado é uma memória que pode ser ligada diretamente aos processadores sem a utilização de uma rede de conexão.

A utilização de memórias multiportas é limitada por questões de custo e escalabilidade (o número de portas, e portanto de processadores, é limitado). Uma boa saída é ligação de cada memória a apenas um conjunto de processadores.

4.5 Redes de interconexão

Critérios de avaliação:

- Escalabilidade: possibilidade de acréscimo de dispositivos sem a necessidade de alteração das características da rede;

- Desempenho: está relacionado com as distâncias envolvidas e om o número de operações simultâneas. O desempenho tem como métricas a latência e a taxa de transferência. A primeira corresponde ao tempo necessário para a transferência dos dados e a segunda representa a quantidade de dados que podem ser comunicados por unidade de tempo. Uma outra questão com impacto no desempenho é se a rede é unidirecional ou bidirecional;

- Custo: basicamente é proporcional ao desempenho desejado e ao número de ligações existentes;

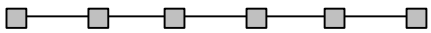
- Confiabilidade: especifica a capacidade de comunicação da rede mediante falha em alguma ligação. Está associada com a existência de caminhos redundantes entre os componentes;

- Funcionalidade: indica os serviços extras (além da comunicação) oferecidos pela rede como *buffering* (armazenamento temporário), ordenação e roteamento automático por hardware.

Redes estáticas

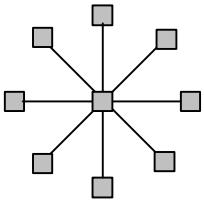
São as que especificam uma ligação direta dedicada entre dois componentes quaisquer. Muito utilizada em multicomputadores. O número de ligações diretas de cada componente define o **grau do nó**. A maior distância (em número de ligações) entre dois componentes quaisquer é chamada de **diâmetro** da rede.

Array linear:



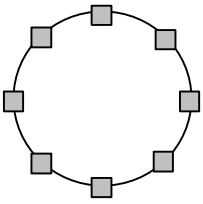
- sem caminhos alternativos

Estrela:



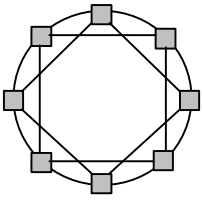
- tráfego intenso no nó central
- problemas no nó central bloqueiam a rede

Anel:



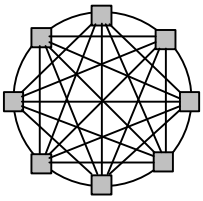
- baixo custo
- diâmetro cresce de forma linear com os nós
- sem caminhos alternativos
- tráfego intenso

Anel chordal:



- menos tráfego no anel central
- caminho alternativos

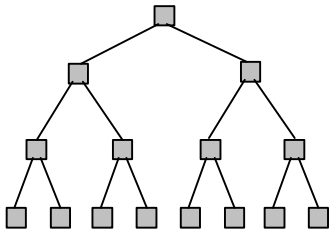
Totalmente conectada:



- alto custo
- grau de nó = número de nós - 1
- diâmetro 1 (ideal)

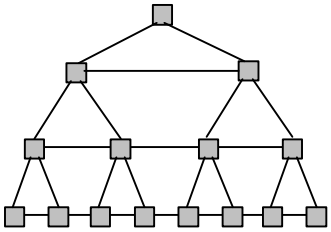
Outro critério para a avaliação de uma rede é a sua adequação a um algoritmo específico. Uma rede do tipo árvore binária, por exemplo, é ideal para a execução de algoritmos do tipo divisão e conquista (*divide and conquer*).

Árvore binária:



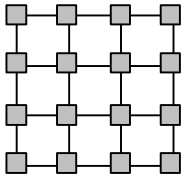
- diâmetro cresce de forma linear com a altura *h*
- grau de nó máximo 3
- sem caminhos alternativos
- nó raiz é um gargalo

X-Tree:



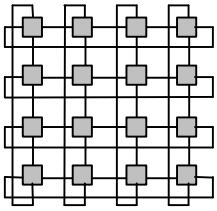
- caminhos alternativos
- grau de nó máximo 5

Malha bidimensional:



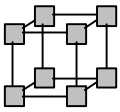
- grau de nó máximo 4
- facilidade de incremento de elementos

Torus:



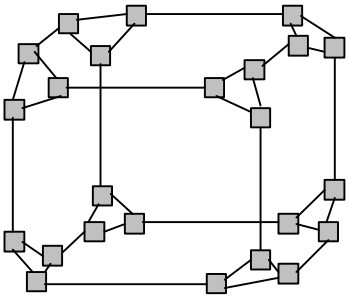
- grau de nó 4
- diâmetro reduzido em relação ao número de nós

Hipercubos:



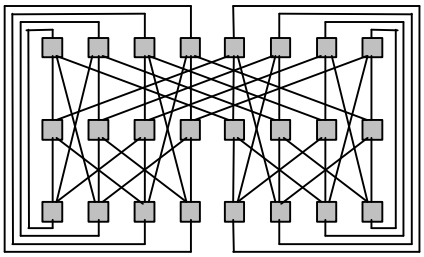
- escalabilidade restrita a potências de 2
- diâmetro = grau de nó
- diâmetro cresce logaritmicamente
- grau de nó = dimensão do cubo (exemplo = 3)

Cubo CCC (*Cube Connected Cycles*):



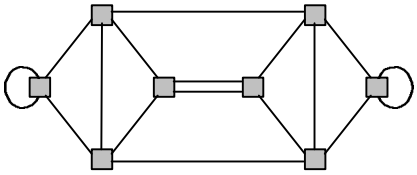
- hipercubo em que cada nó é um anel
- hipercubo de dimensão *d* = anel com *d* nós
- diâmetro cresce logaritmicamente
- grau de nó 3 para qualquer diâmetro

Butterfly:



- grau de nó 4
- diâmetro menor que um cubo CCC
- diâmetro cresce logaritmicamente
- o exemplo é de dimensão 3

Grafo de DeBrujn:



- grau de nó 4
- grafo de dimensão *d* = 2 x *d* nós
- diâmetro cresce logaritmicamente
- o exemplo é de um grafo de dimensão 3

Redes dinâmicas

Corresponde às redes em que as conexões são feitas sob demanda. Não existem ligações fixas entre os componentes. São as mais utilizadas em multiprocessadores. Uma questão importante nas redes dinâmicas é a possibilidade de serem bloqueantes ou não, ou seja, é a possibilidade de conexão entre dois elementos da rede impedir a conexão entre dois outros.

As redes dinâmicas podem ser de três tipos:

- Barramento
- Matriz de chaveamento
- Rede multinível

O barramento é a forma mais simples de conexão mas tem como grande desvantagem o compartilhamento do mesmo meio físico por parte de todos os elementos do sistema. Este fato caracteriza o barramento como altamente bloqueante e de baixa confiabilidade. O fato de ser bloqueante limita a escalabilidade do barramento que é

indicado para sistemas com menos de 50 processadores. Uma forma de minimizar as desvantagens deste tipo de conexão é a utilização de mais de um barramento na conexão dos elementos.

A matriz de chaveamento, ou *crossbar switch*, é uma alternativa não bloqueante de interconexão. Quaisquer elementos podem ser interligados dinamicamente mas esta característica é originada na grande disponibilidade de hardware, o que se reflete em um alto custo do sistema para um grande número de processadores. A escalabilidade fica limitada apenas pelos aspectos econômicos. Uma possível solução para a questão dos custos é a conexão de várias matrizes menores para a implementação de redes com grande número de processadores, mas esta é uma alternativa que torna a rede bloqueante.

As redes multinível são uma variação das matrizes de chaveamento, de maneira que se possa utilizar matrizes de chaveamento padrão (2 x 2, por exemplo) para a interconexão dos elementos. Tais matrizes são organizadas em diversos níveis de conexões, de forma a minimizar a possibilidade de conflito na ligação entre dois componentes quaisquer.

Roteamento de mensagens

A ligação entre dois elementos quaisquer em uma rede normalmente é indireta, ou seja, existem nós intermediários no caminho do nó origem para o nó destino. Desta forma é preciso definir como será feita a condução das mensagens. Existem duas formas básicas de conduzir as mensagens:

Chaveamento de circuito (*circuit switching*): neste caso, em um primeiro instante, o caminho de conexão é estabelecido e somente após a mensagem é enviada. Isto é similar ao que ocorre com as mensagens do sistema telefônico. Não é comum em arquiteturas paralelas.

Chaveamento de pacotes (packet switching): as mensagens não seguem um caminho pré-definido ao longo dos nós da rede. A cada nó atingido pela mensagem um novo nó de destino é escolhido. Isto é importante por não haver reserva de caminho (o que é uma escolha bloqueante) e por que os algoritmos de roteamento podem ser mais flexíveis e podem operar mais rapidamente.

Uma vez entendida a forma de condução da mensagem, é importante discutir as políticas de roteamento, que dizem respeito à maneira como os pacotes serão manipulados pelos nós intermediários. A seguir são apresentadas as duas políticas principais de roteamento de mensagens.

Store-and-forward: podemos considerar que um pacote seja dividido em células, as quais serão transferidas a cada ciclo de comunicação da rede. Na política *store-and-forward*, todas as células de um pacote devem ser recebidas por um nó intermediário para que o pacote comece a ser repassado para o nó seguinte.

Cut-through: esta política é semelhante a uma comunicação *pipeline* das células de um pacote. Tão logo uma célula seja recebida por um nó intermediário, ela pode ser repassada para o nó seguinte. Assim, as diferentes células de um pacote circulam simultaneamente por diferentes nós da rede de conexão.