Plataformas de Programação Paralela

Programação Paralela

Aula 5
Alessandro L. Koerich

Pontifícia Universidade Católica do Paraná (PUCPR)
Ciência da Computação – 6º Período

Aula Anterior

- Taxonomia de Arquiteturas
  - SISD
  - SIMD
  - MISD
  - MIMD
- Organização Lógica
  - Acessando um espaço de dados compartilhado
  - Troca de mensagens

Plano de Aula

- Organização Física de Plataformas Paralelas
  - Arquitetura ideal
  - Arquiteturas convencionais
  - Topologias de rede

Computador Paralelo Ideal

- Uma extensão natural do modelo serial de computação (RAM) consiste em:
  - p processadores
  - Memória global de tamanho ilimitado acessível a todos os processadores
  - Todos os processadores acessam o mesmo espaço de endereçamento
  - Compartilham um *clock* comum, mas podem executar instruções diferentes em cada ciclo

Computador Paralelo Ideal

- Este modelo ideal é chamado de PRAM (*Parallel Random Access Machine*)
- Como PRAMs permitem acesso concorrente a várias posições de memória, dependendo de como estes acessos simultâneos são tratados, PRAMs podem ser divididos em 4 subclasses.
  - EREW
  - CREW
  - ERCW
  - CRCW

Subclasses PRAM

- EREW: *exclusive–read, exclusive–write*
  - O acesso a uma posição de memória é exclusivo
  - Operações *Read* e *Write* concorrentes não são permitidas
  - É o modelo PRAM mais fraco, permitindo concorrência mínima no acesso a memória

Subclasses PRAM

- CREW: *concurrente–read, exclusive–write*
  - São permitidos múltiplos acessos de leitura a uma posição de memória
  - Acessos múltiplos de escritura a uma posição de memória são serializados

Subclasses PRAM

- ERCW: *exclusive–read, concurrente–write*
  - São permitidos múltiplos acessos de escrita leitura a uma posição de memória
  - Acessos múltiplos de leitura a uma posição de memória são serializados

Subclasses PRAM

- CRCW: *concurrente–read, concurrente–write*
  - São permitidos múltiplos acessos de leitura e escrita leitura a uma posição de memória
  - Este é o modelo PRAM mais poderoso

Complexidade Arquitetural

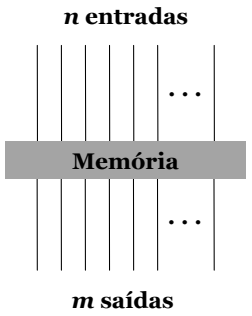
- Considerando a implementação de um EREW PRAM como um computador de memória compartilhada com:
  - *p* processadores
  - memória global de *m* palavras
- Os processadores estão conectados a memória através de um conjunto de chaves (*switches*)
- Estes *switches* determinam a palavra da memória sendo acessada por cada processador

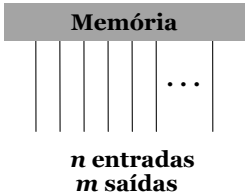
Complexidade Arquitetural

- Em um EREW PRAM, cada um dos *p* processadores no conjunto pode acessar qualquer palavra da memória
- Para garantir a conectividade, o número total de chaves (*switches*) deve ser *m.p*
- Para um tamanho de memória razoável, a construção de uma rede de chaveamento (*switching*) desta complexidade é muito caro
- Então..... Modelos de computação PRAM são impossíveis de serem realizados na prática

Redes de Interconexão: Organização Física

- Fornecem mecanismos para a transferência de dados entre nós de processamento e módulos de memória





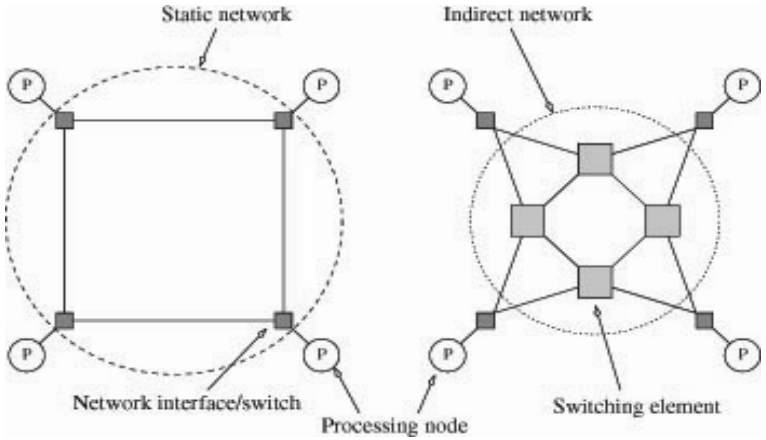
Redes de Interconexão: Organização Física

- Redes de interconexão podem ser construídas tipicamente usando *links* e *switches*
- **Link**: meio físico composto por um conjunto de fios ou fibras capazes de transportar informação
- Influência das características ⇦ capacitância, comprimento.

Redes de Interconexão: Organização Física

- Classificação: estáticas ou dinâmicas
  - Estáticas: *links* de comunicação ponto a ponto entre nós de processamento. a.k.a. ⇨ redes diretas
  - Dinâmicas: são construídas utilizando *switches* e *links* de comunicação. *Links* de comunicação estão conectados dinamicamente pelos *switches* para estabelecer caminhos entre nós de processamento e bancos de memória. a.k.a ⇨ redes indiretas

Redes de Interconexão: Organização Física



Rede Estática
- 4 elementos de processamento (nós)
- Configuração *Mesh*: cada nó ⇨2 outros nós

Rede Dinâmica
- 4 nós conectados via uma rede de *switches* a outros nós

Redes de Interconexão: Organização Física

- *Switch* – um conjunto de portas de entrada e um conjunto de portas saída
- Funcionalidade mínima: mapeamento das portas de entrada para as portas de saída
- Mas também podem fazer:
  - Buferização interna: quanto uma porta de saída requisitada está ocupada
  - Roteamento: para aliviar o congestionamento na rede
  - *Multicast*: mesma saída em portas múltiplas

Redes de Interconexão: Organização Física

- O mapeamento das portas de entrada para as portas de saída pode ser feito utilizando–se uma variedade de mecanismos:
  - Barras transversais físicas (*crossbars*)
  - Multiplexador–Demultiplexador
  - Bus multiplexados
- A conectividade entre nós e a rede é feita através de uma interface de rede.

Redes de Interconexão: Organização Física

- A interface de rede tem portas de entrada e saída, que colocam os dados dentro e fora da rede.
- A interface de rede tem as responsabilidades de:
  - Empacotar os dados;
  - Computar informações da rota;
  - Buferizar dados chegando e partindo de modo a compatibilizar as velocidades da rede e elementos de processamento
  - Checagem de erro.

Topologias de Rede

- Uma grande variedade de topologias tem sido utilizadas para interconectar redes.
- Objetivo: buscar um compromisso entre o custo, escalabilidade e performance.

Topologias de Rede

- Tipos:
  - Baseadas em Bus
  - Crossbar (barra transversal)
  - Multiestágio
  - Completamente Conectadas
  - Conectadas em Estrela
  - Arranjo Linear
  - Malha
  - Malhas *k–d*
  - Baseadas em Árvores

Redes Baseadas em BUS

- Tipo mais simples de rede
- Consiste em um meio compartilhado que é comum a todos os nós
- **BUS (ou Barramento)** : propriedade desejável ⇨ custo da rede cresce linearmente com o número de nós *p*.
- Este custo esta associado com a interface do BUS.

Redes Baseadas em BUS

- A distância entre dois nós quaisquer na rede é constante.
- Bus também são ideais para o broadcast de informações entre os nós
- Como o meio de transmissão é compartilhado, existe um overhead comparado a transferência de mensagens ponto a ponto.

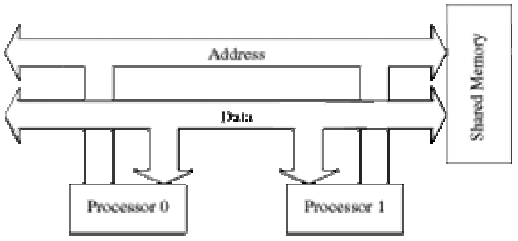
Redes Baseadas em BUS

- A limitação da largura de banda impõe uma limitação na performance global quando o número de nós cresce.
- Tipicamente, máquinas baseadas em bus são limitadas a dúzias de CPUs.
- Ex: Servidores *Sun Enterprise* e *Pentium Intel multiprocessador*

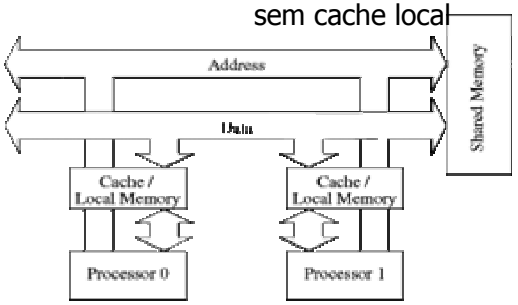
Redes Baseadas em BUS

- As exigências da largura de banda podem ser reduzidas utilizando–se um cache para cada nó.
- Dados privados são “cacheados” nos nós e somente dados remotos são acessados através do BUS.

Redes Baseadas em BUS



sem cache local



com cache local

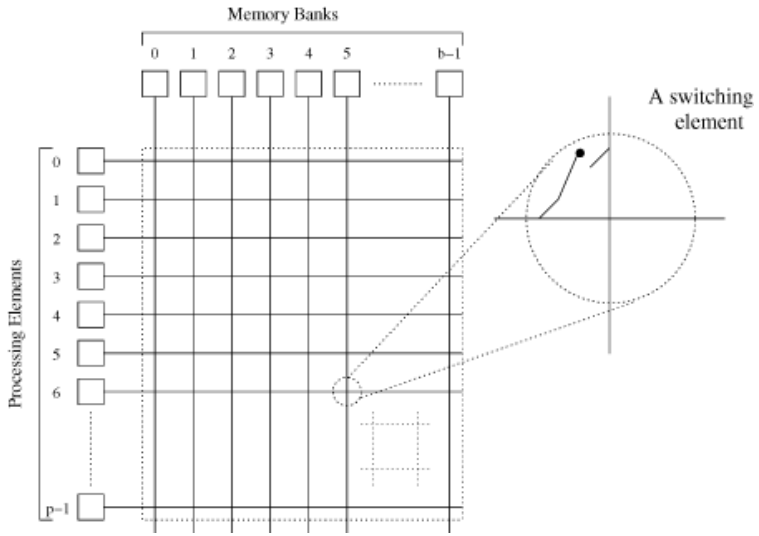
Redes *Crossbar* (ou barra transversal)

- Uma maneira simples de conectar *p* processadores a *b* bancos de memória.
- Uma rede ***crossbar*** utiliza um ***grid*** de *switches*.
- É uma rede **"*não blocante*"**
- A conexão de um nó de processamento à um banco de memória não bloqueia a conexão entre qualquer outro nó de processamento e outro banco de memória.

Redes *Crossbar*

- O número total de nós de chaveamento necessários para implementar tal rede é *p.b*.
- É razoável assumir que o número de bancos de memória (*b*) é pelo menos *p*.
- Caso contrário, haverão alguns nós de processamento que não serão capazes de acessar um banco de memória.
- Com o crescimento do número de nós de processamento, é difícil de obter um chaveamento rápido ⇨ não muito escaláveis em termos de custo

Redes *Crossbar*



- Ex: *Sun Ultra HPC Server, Fujitsu VPP500, TOP1*

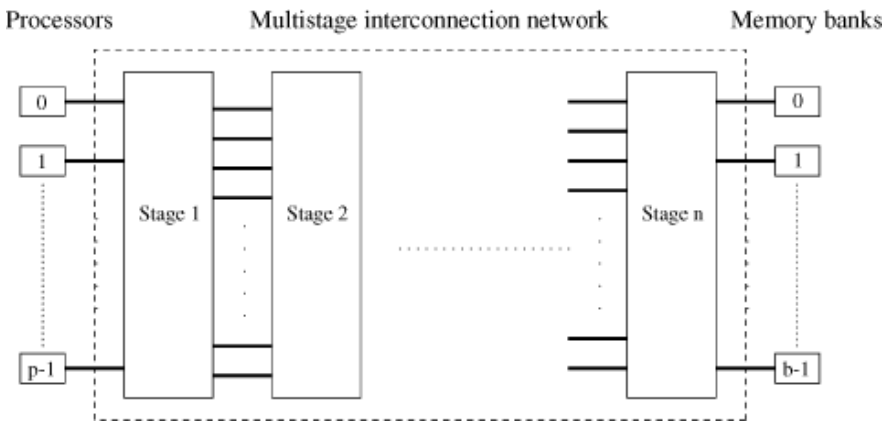
Redes *Crossbar* x *Bus*

Crossbar	*Bus* Compartilhado
Escalável em termos de performance	Não escalável em termos de performance
Não escalável em termos de custo	Escalável em termos de custo

Redes Multiestágios

- Uma classe intermediária de rede que recai entre *bus* e *crossbar*
- Consiste de *p* processadores e *b* bancos de memória
- Uma rede de conexão multiestágios comumente utilizada é a rede Omega.

Redes Multiestágios



Um esquema de uma rede de interconexão multiestágios típica.

Redes Multiestágios

- A rede Omega consiste em *log p* estágios, onde *p* é o número de nós de processamento (entradas) e também o número de bancos de memória (saídas).
- Cada estágio da rede consiste em um padrão de interconexão que conecta *p* entradas e *p* saídas.

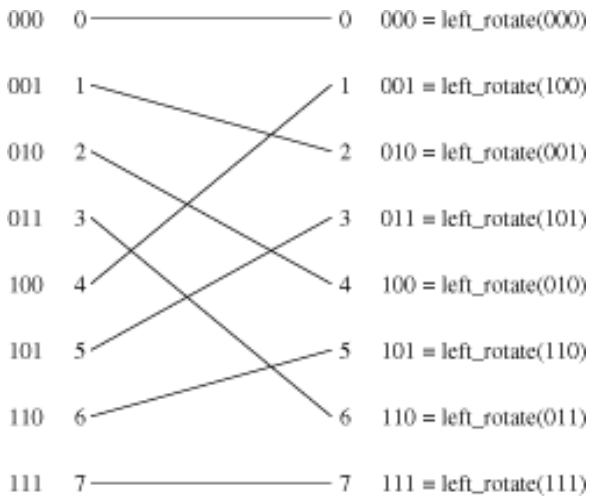
Redes Multiestágios

- Existe um link entre entradas *i* e saídas *j* se o seguinte for verdadeiro:

$$j = \begin{cases} 2i & 0 \le i \le p/2 - 1 \\ 2i + 1 - p & p/2 \le i \le p - 1 \end{cases}$$

- Esta equação representa uma operação de rotação à esquerda sobre a representação binária de *i* para obter *j*.
- Este padrão de interconexão é chamado de "*perfect shuffle*"

Redes Multiestágios



- "*perfect shuffle*" interconectando 8 entradas e saídas.

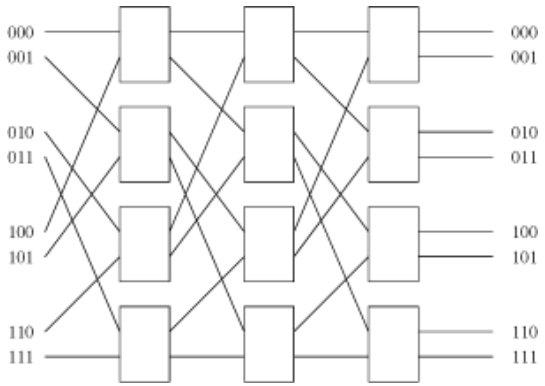
Redes Multiestágios

- Cada *switch* tem dois modos de conexão:

- Conexão *pass–through*: entradas passam direto a saída;
- Conexão *cross–over*: entradas são cruzadas e então enviadas a saída

Redes Multiestágios

- Uma rede Ômega tem $p/2$ x $\log p$ nós de chaveamento.
- O custo de tal rede cresce com $p \log p$.



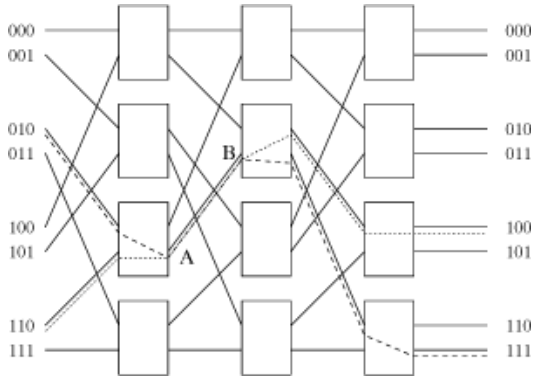
- Uma rede Ômega completa conectando 8 entradas e 8 saídas.

Redes Multiestágios

Roteamento de dados:

- *s* quer escrever em *t*
  - 1º nó de chaveamento:
    - se os bits mais significantes de *s* e *t* forem iguais ⇨ modo *pass–through*.
    - se forem diferentes ⇨ modo *cross–over*
  - Outros nós: mesmo critério, porém utilizando o próximo bit mais significativo

Redes Multiestágios



- Roteamento do processador 2 (010) para banco de memória 7 (111) e processador 6 (110) para o banco de memória 4 (100)

Redes Multiestágios

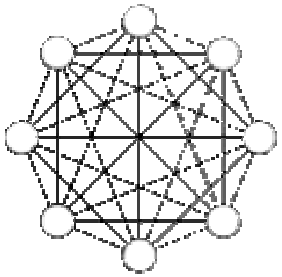
- Importante: Link de comunicação *AB* é utilizado por ambos caminhos de comunicação.
- O acesso de um desabilita o acesso do outro.
- Rede com bloqueio

Redes Multiestágios

- Exemplos:
  - BBN *Butterfly* (1989);
  - NYU *Ultracomputer* (1983);
  - IBM *RP3* (1985).

Redes Completamente Interconectadas

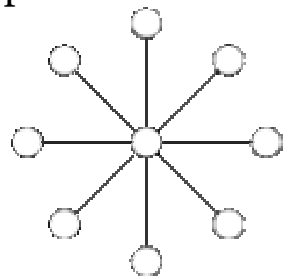
- Cada nó tem um *link* de comunicação direto com todos os outros nós na rede.



- Esta rede é ideal no sentido que um nó pode enviar uma mensagem para outro nó em uma única etapa, pois existe um link entre eles.

Redes Conectadas em Estrela

- Um processador atua como processador central. Todos os outros processadores tem um link de comunicação conectando–os ao processador central.



- O processador central é o gargalo na topologia estrela.

Arranjos Lineares, Malhas e Malhas *k*-d

- Devido ao grande número de *links* nas redes completamente conectadas, redes esparsas são geralmente utilizadas na construção de computadores paralelos

⇨ Arranjos lineares e hipercubos

Arranjos Lineares, Malhas e Malhas *k*-d

- Arranjo linear: rede estática na qual cada nó (exceto os dois nós extremos) possui dois vizinhos, um a direita e um a esquerda.

- Uma extensão simples do arranjo linear é a topologia em anel ou *tórica* 1D (*torus*). Esta topologia tem uma conexão entre as extremidades do arranjo linear.

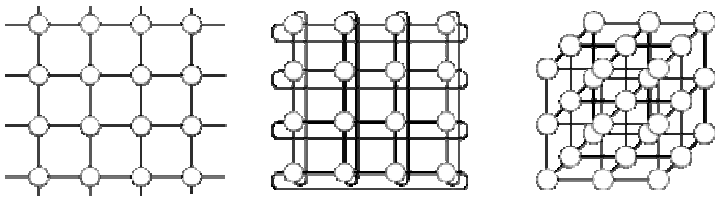
Arranjos Lineares, Malhas e Malhas *k*-d

- Uma malha 2D é uma extensão do arranjo linear para 2 dimensões.
- Cada dimensão tem $p^{1/2}$ nós onde um nó é identificado por um par (i, j).
- Cada nó (exceto os da periferia) está conectado a quatros outros nós cujos índices diferem em 1 em qualquer dimensão

Arranjos Lineares, Malhas e Malhas *k*-d

- Uma malha 2D tem a propriedade de poder ser arranjada em um espaço 2D, tornando–a atrativa do ponto de vista das ligações.
- Malhas 2D são freqüentemente utilizadas para interconectar máquinas paralelas
- Malhas 2D podem crescer com ligações em anel formando topologias tóricas 2D (*torus*).

Arranjos Lineares, Malhas e Malhas *k*-d



- Cubo 3D é uma generalização de malhas 2D para 3 dimensões.
- Cada nó em um cubo 3D está conectado à seis outros nós (exceto os da periferia), dois ao longo de cada dimensão.

Arranjos Lineares, Malhas e Malhas *k*-d

- Uma variedade de simulações físicas comumente executadas em máquinas paralelas podem ser mapeadas naturalmente para topologias de rede 3D (modelamento de estruturas).
- Por esta razão, cubos 3D são utilizados comumente como redes de interconexão em computadores paralelos.

  Exemplo: *Cray T3E*

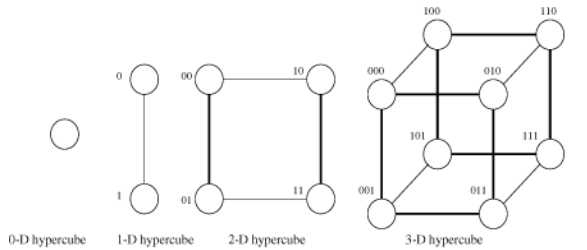
Arranjos Lineares, Malhas e Malhas *k*-d

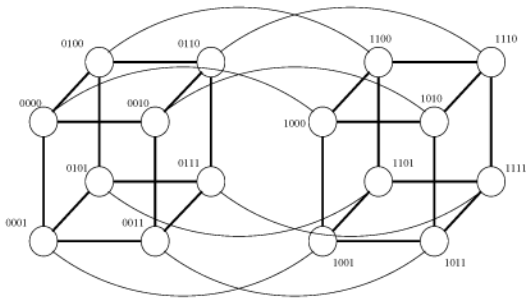
- A classe geral de malhas *k–d* refere–se as topologias que consistem em *d* dimensões com *k* nós ao longo de cada dimensão.
- **Topologia Hipercubo**: temos 2 nós ao longo de cada dimensão e *log p* dimensões.
- A construção de um hipercubo é ilustrada a seguir....

Arranjos Lineares, Malhas e Malhas *k*-d

- Um hipercubo de dimensão 0 (zero) consiste de 2^0 nós.
- Um hipercubo de dimensão 1 é construído a partir de dois hipercubos de dimensão 0 (zero).
- Em geral, um hipercubo de dimensão *d* é construído pela conexão dos nós de 2 hipercubos de dimensão (*d*–1).

Arranjos Lineares, Malhas e Malhas *k*-d





Arranjos Lineares, Malhas e Malhas *k*-d

- Exemplo de máquinas Malha (Mesh):
  - *Cray T3E.*
- Exemplo de máquinas Malha 2D (Mesh):
  - *Intel Paragon XP/S* (1991);
  - *Mosaic C* (1992).
- Exemplo de máquinas Malha 3D (Mesh):
  - *MIT J – Machine* (1992).
- Exemplo de máquinas hipercubo:
  - *Cosmic Cube* (1985);
  - *nCUBE2* (1990);
  - *Intel iPSC–1, iPSC–2,* e *iPSC/860*;
  - *SGI Origin 2000.*

Redes Baseadas em Árvores

- Em uma rede em árvore há somente um único caminho entre um par de nós.
- Arranjos lineares e redes conectadas em estrela são casos particulares de redes em árvore.
- **Redes estáticas em árvore** têm um elemento de processamento em cada nó.
- **Redes dinâmicas em árvore** têm nós intermediários de chaveamento e as folhas são os nós de processamento.

Redes Baseadas em Árvores

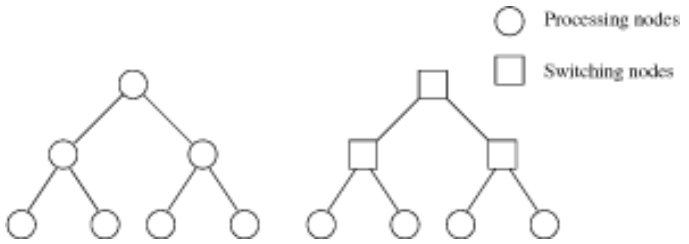


Figura: Redes completas em árvore binária, sendo estática e dinâmica.

- Para rotear uma mensagem em uma árvore, o nó fonte manda uma mensagem árvore acima, até que ela atinja o nó da raiz da menor subárvore, contendo tanto os nós fonte e destino.

Redes Baseadas em Árvores

- Rede em árvore sofrem de gargalos de comunicação nos níveis superiores da árvore.
- Ex: quando vários nós do lado direito querem se comunicar com nós do lado esquerdo.
- Este problema pode ser minimizado em árvores dinâmicas aumentando–se o número de links de comunicação e nós de chaveamento próximos a raiz. (*Fat Tree*)

Redes Baseadas em Árvores

Figura: Exemplo de uma *"Fat Tree"*

Redes Baseadas em Árvores

- Exemplos:
  - DADO (1986) utiliza árvore binária completa de profundidade 10;
  - *Thinking Machines* CM–5 (1991) utiliza uma *fat–tree*.

Próxima Aula

- Avaliação de Redes de Interconexão Estáticas
- Avaliação de Redes de Interconexão Dinâmicas
- Coerência de *Cache* em Sistemas com Multiprocessadores.