SSC0611
Arquitetura de Computadores

24ª Aula – Arquiteturas Paralelas

Arquitetura MIMD com Memória Distribuída

Profa. Sarita Mazzini Bruschi
sarita@icmc.usp.br

Comunicação em MIMD com Memória Distribuída

- A transmissão de mensagens é uma maneira de comunicação em máquinas MIMD com Memória Distribuída
- Uma mensagem pode ser definida como um bloco de informações relacionadas que trafegam entre os processadores
- Como nas máquinas MIMD com memória compartilhada, os programas são particionados em pequenas partes, cada uma podendo ser executada por um processador de maneira concorrente

Comunicação em MIMD com Memória Distribuída

- O tamanho do processo que está sendo executado é determinado pelo programador e pode ser descrito através de sua granulosidade:

$$Granulosidade = \frac{tempo\ de\ computação}{tempo\ de\ comunicação}$$

Comunicação em MIMD com Memória Distribuída

- Granulosidade grossa (*coarse granularity*)*:* cada processo gasta uma grande quantidade de tempo processando instruções sequenciais
- Granulosidade média (*medium granularity*): balanço entre computação e comunicação
- Granulosidade fina (*fine granularity*): cada processo executa poucas instruções sequenciais

Comunicação em MIMD com Memória Distribuída

- Fator importante: Rede de Interconexão
- Fatores de projeto das redes de interconexão:
  - *Bandwidth* do link de comunicação
    - Número de bits que podem ser transmitidos por unidade de tempo (bits/segundo)
    - Links podem ser unidirecionais ou bidirecionais e transmitir um ou vários bits de cada vez
  - Latência da rede
    - Tempo para completar a transferência de uma mensagem
    - Pode ser estimada a partir do tempo de determinação do caminho, que depende do número de nós, e do tempo de transmissão, que depende do tamanho da mensagem

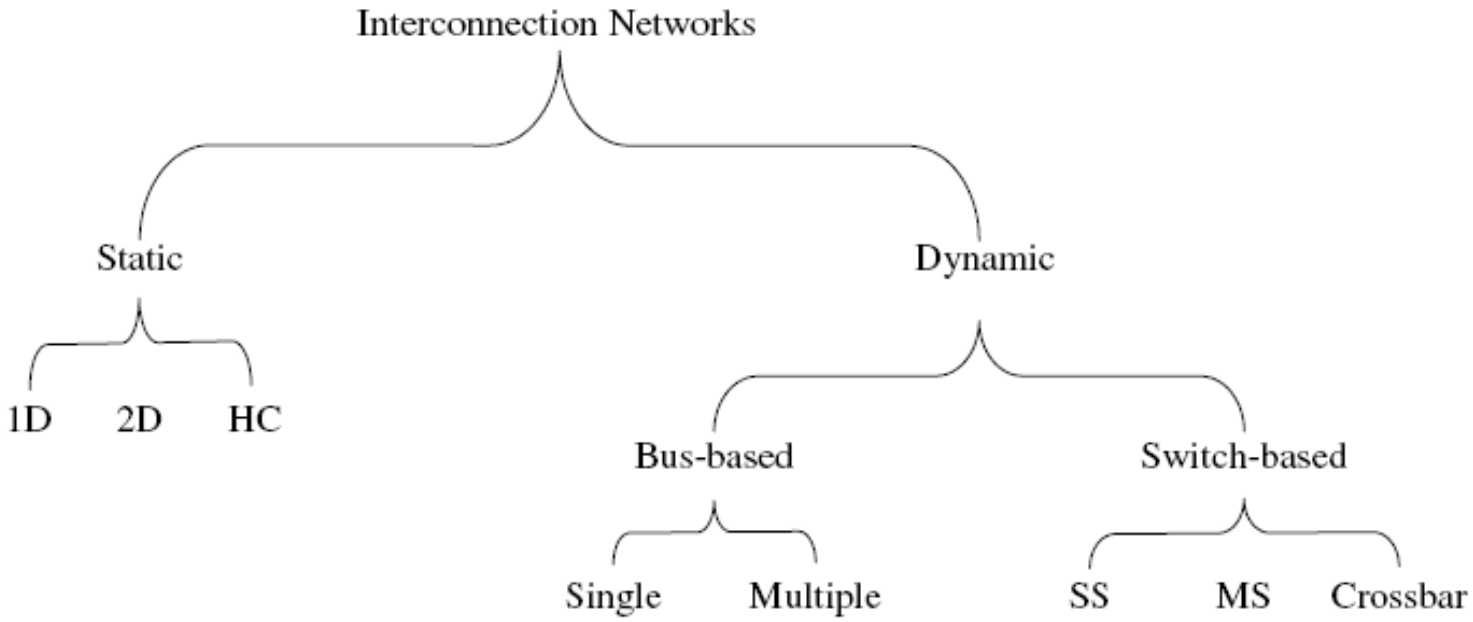
Redes de Interconexão

- Critérios utilizados para a classificação das redes:
  - Modo de operação
    - Síncrona: um único clock é usado por todos os componentes do sistema
    - Assíncrona: não precisa de um clock global (utiliza protocolos de *handshaking*)
  - Estratégia de controle
    - Centralizada: uma única unidade de controle central é utilizada para supervionar e controlar a operação de todos os componentes do sistema.
    - Descentralizada: a função de controle é distribuída entre os componentes do sistema.

Redes de Interconexão

- Critérios utilizados para a classificação das redes:
  - Técnicas de comutação:
    - Circuito: um caminho completo tem que ser estabelecido antes de se iniciar a transmissão, a qual se manterá fixa durante todo o período de comunicação
    - Pacote: comunicação feita através de mensagens, divididas em unidades menores denominadas pacotes
  - Topologia
    - Estática: existem caminhos fixos e diretos entre os nós
    - Dinâmica: conexões entre entradas e saídas são feitas utilizando elementos de conexão, denominados chaves. Dependendo da configuração desses elementos, diferentes interconexões podem ser estabelecidas

Redes de Interconexão
Topologias



1D – Uma dimensão
2D – Duas dimensões
HC – Hiper Cubo

SS – Single-stage
MS – Multi-stage

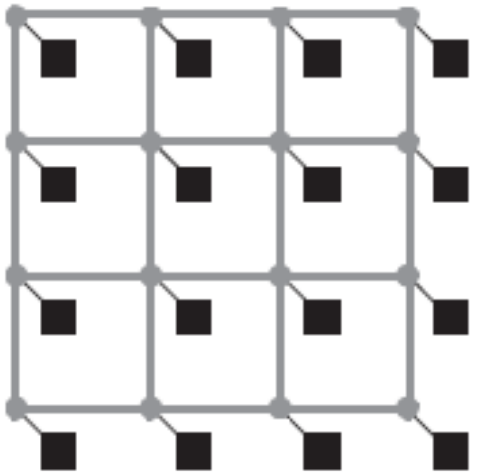
Redes de Interconexão
Topologias

- Critérios para avaliação de uma topologia:
  - **Número total de ligações entre os componentes**
  - **Grau do nó**: quantas ligações diretas cada componente possui
  - **Diâmetro**:maior distância entre dois componentes quaisquer da rede

Redes de Interconexão
Topologias - Exemplos

- Topologia estática
  - Anel

  - Malha bi-dimensional

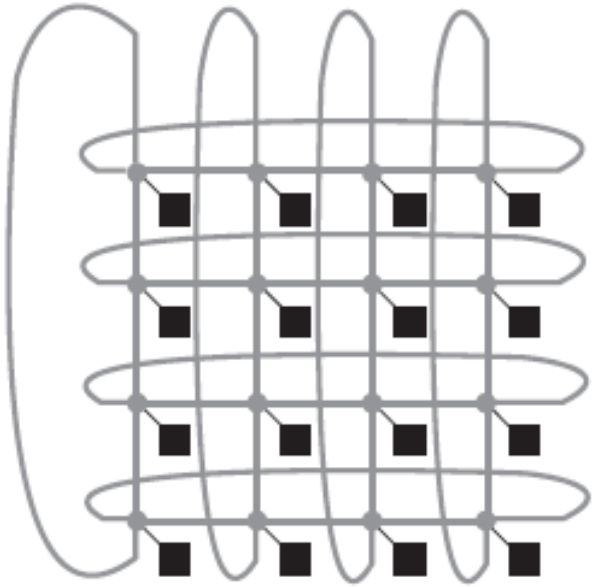


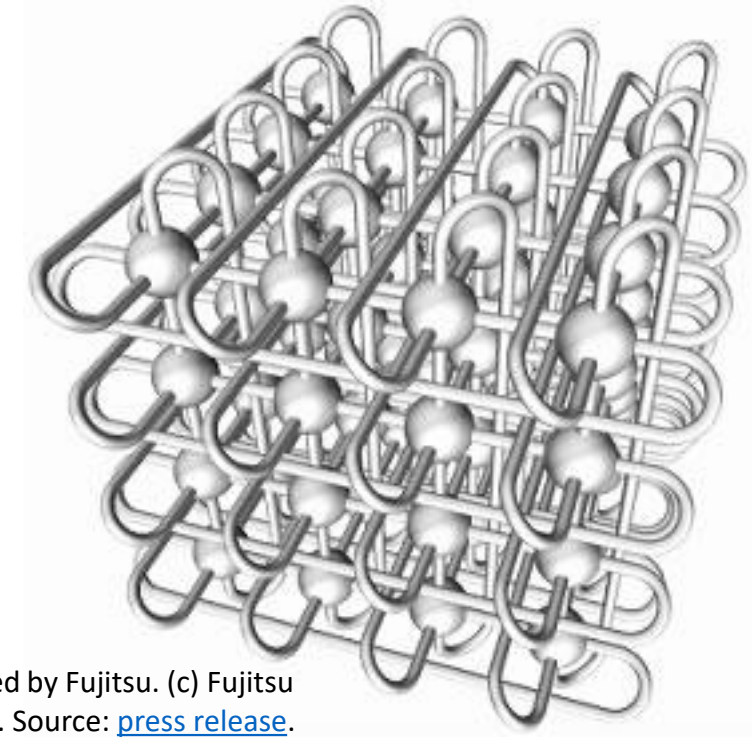
Processador/Memória

Chaves

Redes de Interconexão
Topologias - Exemplos

- Topologia estática
  - Torus (malha bi-dimensional)



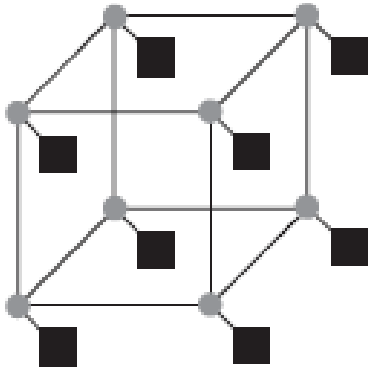


3D torus rendered by Fujitsu. (c) Fujitsu and RIKEN, 2009. Source: press release.

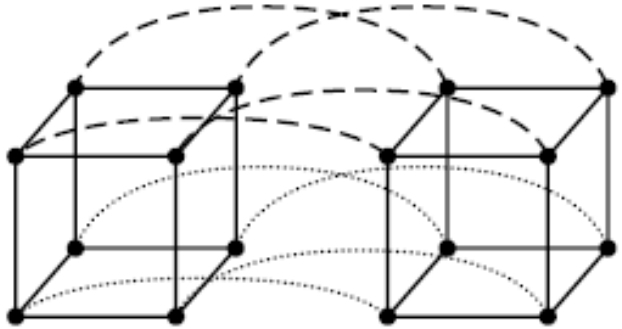
http://www.fujitsu.com/global/about/businesspolicy/tech/k/whatis/network/index.html

Redes de Interconexão
Topologias - Exemplos

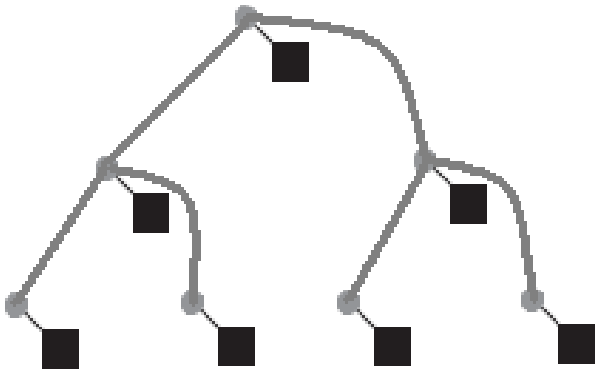
- Topologia estática
  - Cubo



Hipercubo 4D

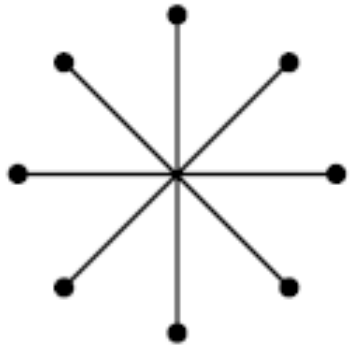


  - Árvore binária

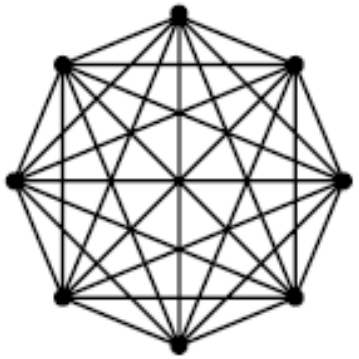


Redes de Interconexão
Topologias - Exemplos

- Topologia estática
  - Estrela



  - Totalmente conectada

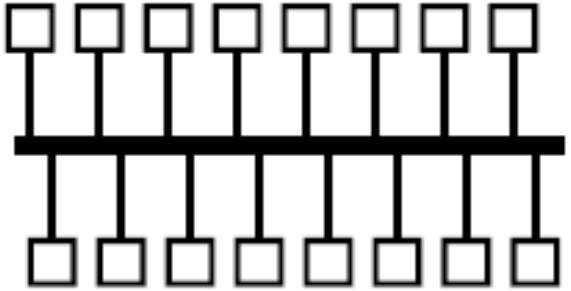


Redes de Interconexão
Topologias - Comparações

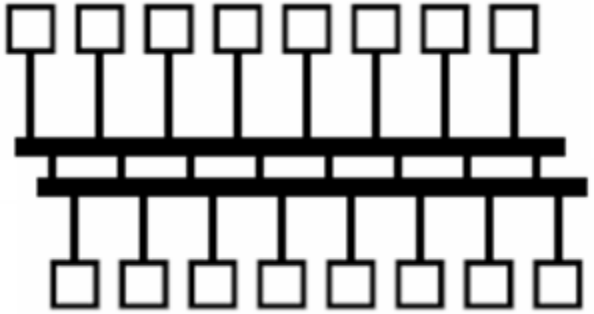
	Número ligações	**Grau do nó**	**Diâmetro**
Anel	*n*	*2*	*n/2*
Estrela	*n*	*n – 1*	*2*
Totalmente conectada	$(n^2 – n)/2$	*n – 1*	*1*

Redes de Interconexão
Topologias - Exemplos

- Topologia Dinâmica
  - Barramento
    - Single



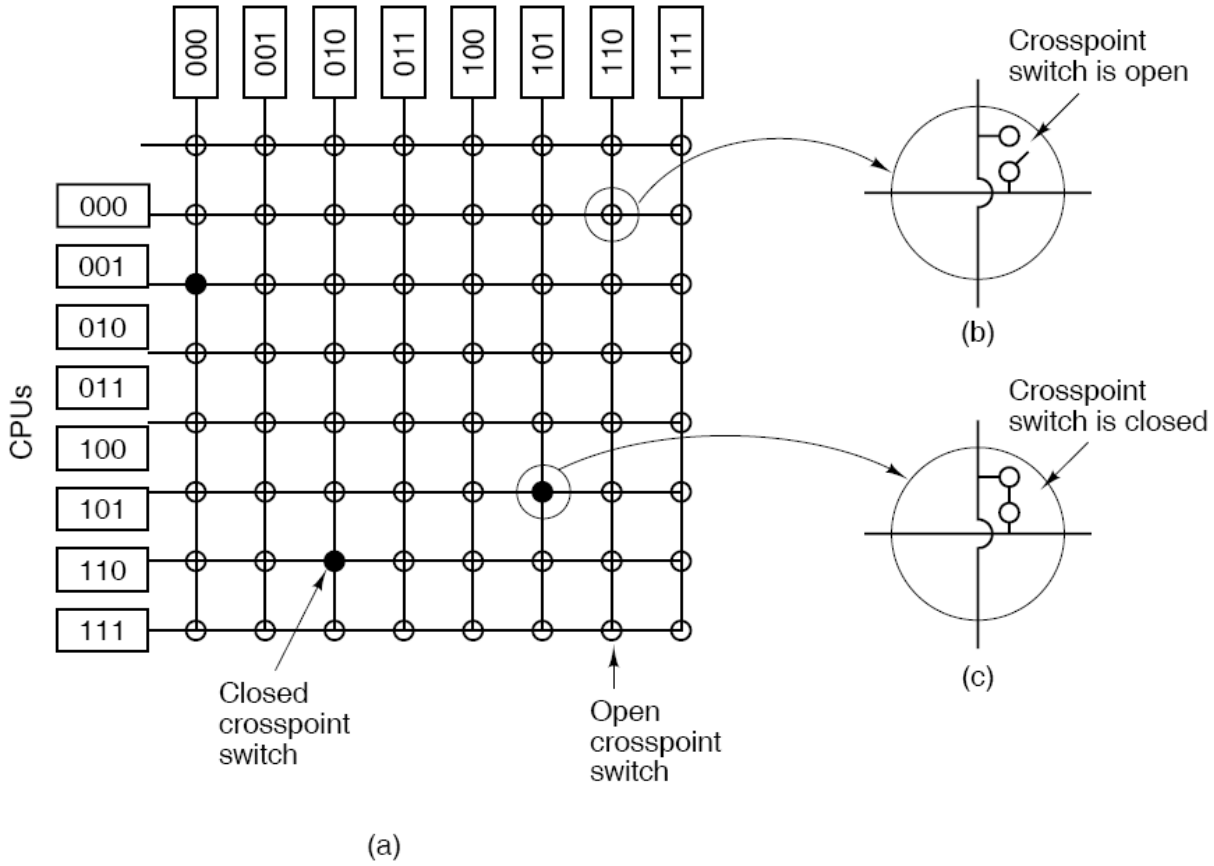
    - Multiple



Redes de Interconexão
Topologias - Exemplos

- Topologia Dinâmica
  - Crossbar

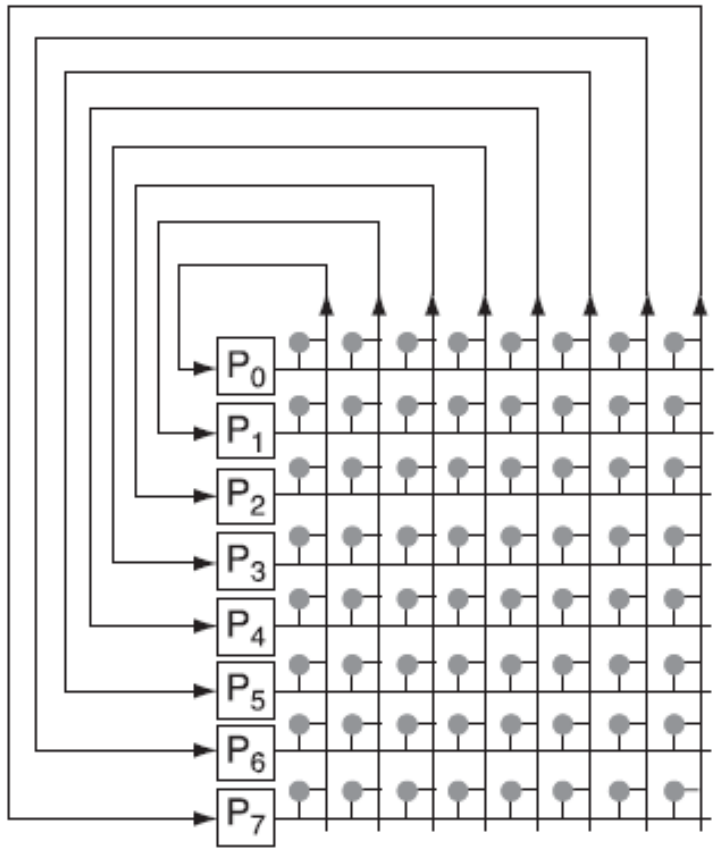
Proc/Mem



Redes de Interconexão
Topologias - Exemplos

- Topologia Dinâmica
  - Crossbar

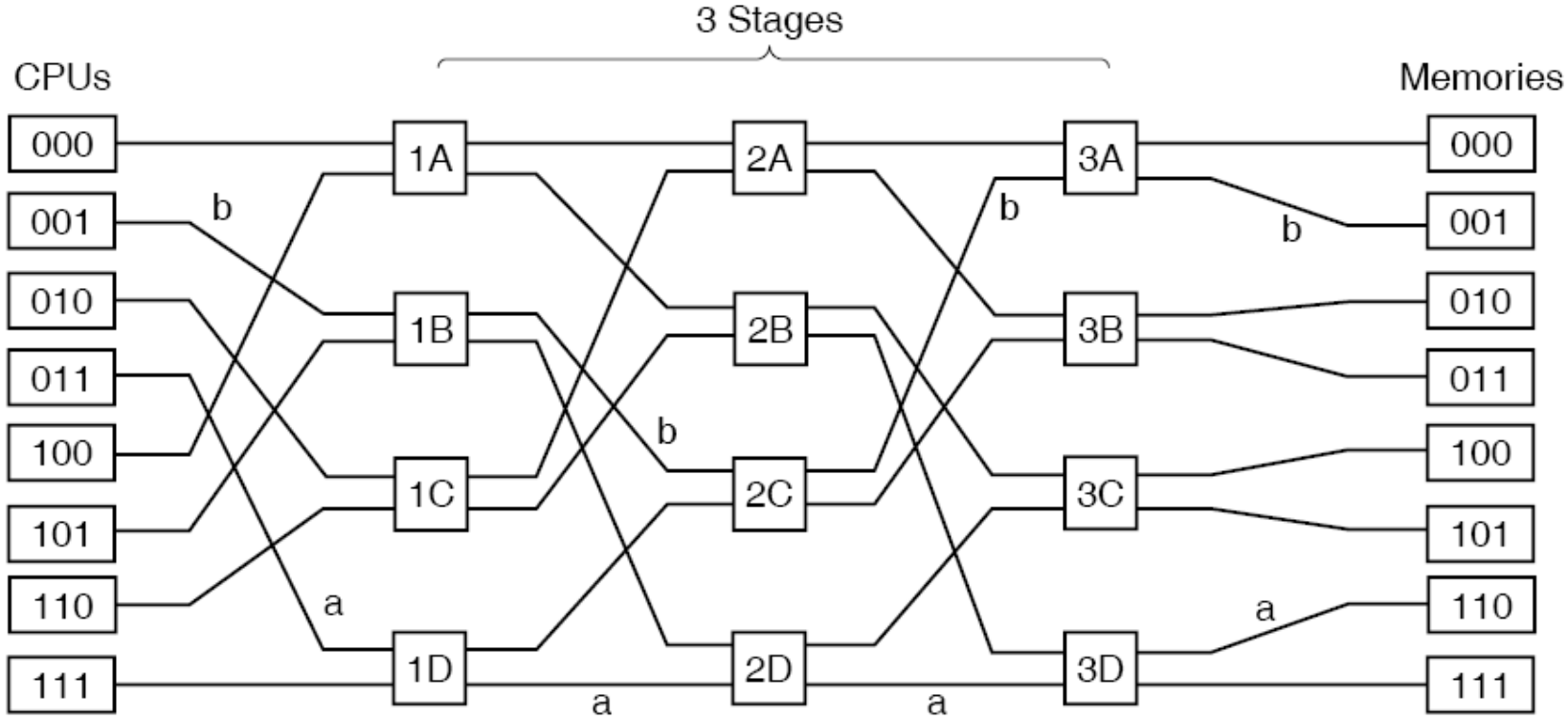
  Proc/Proc



a. Crossbar

Redes de Interconexão
Topologias - Exemplos

- Topologia Dinâmica
  - Omega (*Multistage*) – Proc/Mem

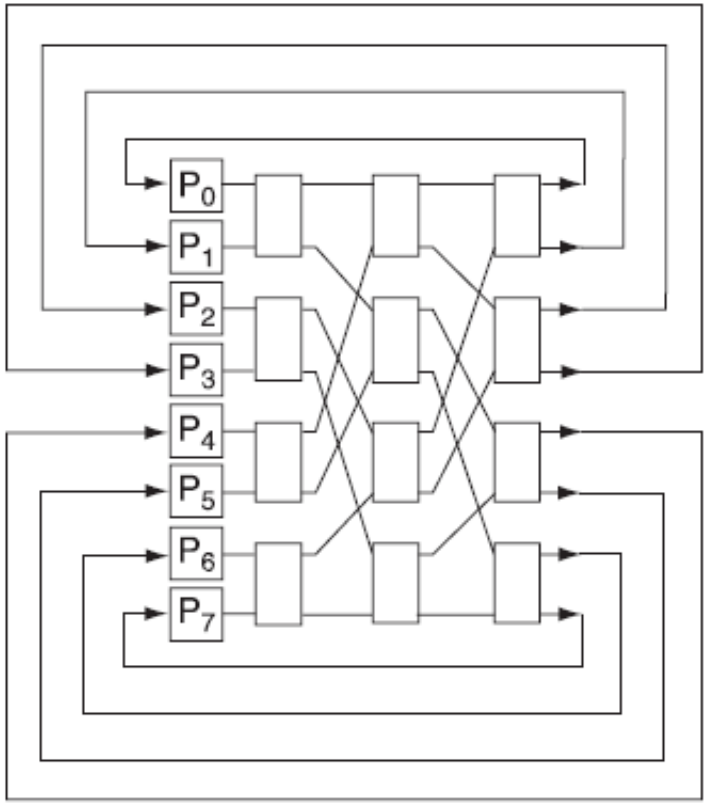


Redes de Interconexão
Topologias - Exemplos

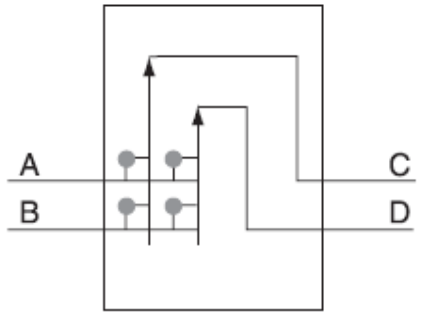
- Topologia Dinâmica
  - Omega

  (*Multistage*)

  Proc/Proc



b. Rede Omega



Comunicação em MIMD com Memória Distribuída

- O Sistema de Comunicação provê tipicamente os seguintes serviços para as aplicações:
  - transmissão da mensagem do processador origem ao processador destino;
  - garantia de que as mensagens entre um par de processadores origem e destino serão entregues na ordem em que foram enviadas;
  - confiabilidade na transmissão das mensagens
  - o sistema de comunicação tenha uma camada de software para prover, através de protocolos implementados em software, os serviços necessários a partir dos recursos de hardware disponíveis

Comunicação em MIMD com Memória Distribuída

- As mensagens são transmitidas através de primitivas *send* e *receive*
- Cada mensagem percorre o seguinte caminho:
  - nó de processamento origem
  - interface de rede do nó de origem
  - rede de interconexão
  - interface de rede do nó destino
  - nó de processamento destino
- A interação do nó de processamento com a sua interface de rede é realizada através de chamadas ao sistema operacional.
- As primitivas podem ser classificadas como:
  - Síncronas ou assíncronas
  - Bloqueantes ou não bloqueantes

Comunicação em MIMD com Memória Distribuída

- Comunicação Síncrona
  - O processo que deseja enviar uma mensagem faz uma requisição inicial de autorização para enviar a mensagem ao processo destinatário.
  - Depois de receber essa autorização, a operação de envio ("send") é realizada.
  - A operação de envio só se completa quando a operação de recepção (“receive”) correspondente retorna uma mensagem de “acknowlegement”.
  - Portanto, ao se concluir a operação de envio, os buffers e estruturas de dados utilizados na comunicação podem ser reutilizados.

Comunicação em MIMD com Memória Distribuída

- Comunicação Síncrona
  - Este modo de comunicação é simples e seguro, contudo elimina a possibilidade de haver superposição entre o processamento da aplicação e o processamento da transmissão das mensagens.
  - Requer um protocolo de quatro fases para a sua implementação do lado do transmissor: pedido de autorização para transmitir; recebimento da autorização; transmissão propriamente dita da mensagem; e recebimento da mensagem de "*acknowledgement*".

Comunicação em MIMD com Memória Distribuída

- Comunicação Assíncrona Bloqueante
  - A operação de envio ("*send*") só retorna o controle para o processo que a chamou após ter sido feita a cópia da mensagem a ser enviada de um buffer da aplicação para um buffer do sistema operacional.
  - Portanto, ao haver o retorno da operação de envio, a aplicação está livre para reutilizar o seu buffer, embora não haja nenhuma garantia de que a transmissão da mensagem tenha completado ou vá completar satisfatoriamente.

Comunicação em MIMD com Memória Distribuída

- Comunicação Assíncrona Bloqueante
  - A operação de envio bloqueante difere da operação de envio síncrona, uma vez que não é implementado o protocolo de quatro fases entre os processos origem e destino.
  - A operação de recepção ("*receive*") bloqueante é semelhante à operação de recepção síncrona, só retornando para o processo que a chamou após ter concluído a transferência da mensagem do buffer do sistema para o buffer especificado pela aplicação. A diferença em relação a operação de recepção síncrona é que a mensagem de “acknowledgement” não é enviada.

Comunicação em MIMD com Memória Distribuída

- Comunicação Assíncrona não Bloqueante
  - Na primeira fase, a operação de *"send"* retorna imediatamente, tendo apenas dado início ao processo de transmissão da mensagem e a operação de *"receive"* retorna após notificar a intenção do processo de receber uma mensagem
  - As operações de envio e recepção propriamente ditas são realizadas de forma assíncrona pelo sistema
  - O retorno das operações de *“send”* e de *“receive”* nada garante e não autoriza a reutilização das estruturas de dados para nova mensagem

Comunicação em MIMD com Memória Distribuída

- Comunicação Assíncrona não bloqueante
  - Na segunda fase, é verificado se a operação de troca de mensagens iniciada anteriormente pelas primitivas *"send"* e*"receive"* já foi concluída
  - Somente após esta verificação é que o processo que realizou o envio de dados pode alterar com segurança sua área de dados original.
  - Este modo de comunicação é o que permite maior superposição no tempo entre computações da aplicação e o processamento da transmissão das mensagens.

Comunicação em MIMD com Memória Distribuída

- A latência total na comunicação por troca de mensagens ser definida pela soma das seguintes parcelas:
  - latência total = overhead de software + overhead da interface da rede + latência da rede de interconexão
- O *overhead* de software é devido a um conjunto de fatores:
  - complexidade do protocolo de comunicação,
  - processamento de interrupções
  - troca de contexto devido às chamadas ao sistema
  - cópias dos dados entre "buffers", etc

Arquitetura MIMD com Memória Distribuída

- As arquiteturas MIMD com memória distribuída, ou multicomputadores, podem variar na estrutura e no tamanho
- Dois estilos podem ser observados:
  - MPP – *Massively Parallel Processors*
  - Clusters

MPP – *Massively Parallel Processors*

- Supercomputadores muito caros
- Utilizado em computação científica, engenharia e indústria para cálculos de larga escala ou para uma grande quantidade de transações por segundo
- Utilizado também em *data warehousing* (manipulação de uma grande quantidade de dados) e *big data*

MPP – *Massively Parallel Processors*

- Composto por CPUs padrões
  - Intel Pentium
  - Sun UltraSPARC
  - IBM PowerPC
- O que faz a diferença nos MPPs é a rede de interconexão proprietária de alta performance projetada para mover mensagens com baixa latência a uma alta *bandwidth*
- Outra característica é a alta capacidade de E/S
- Hardware e software especiais para tratar tolerância a falhas (monitoramento do sistema, detecção e recuperação das falhas)
- Exemplo: BlueGene

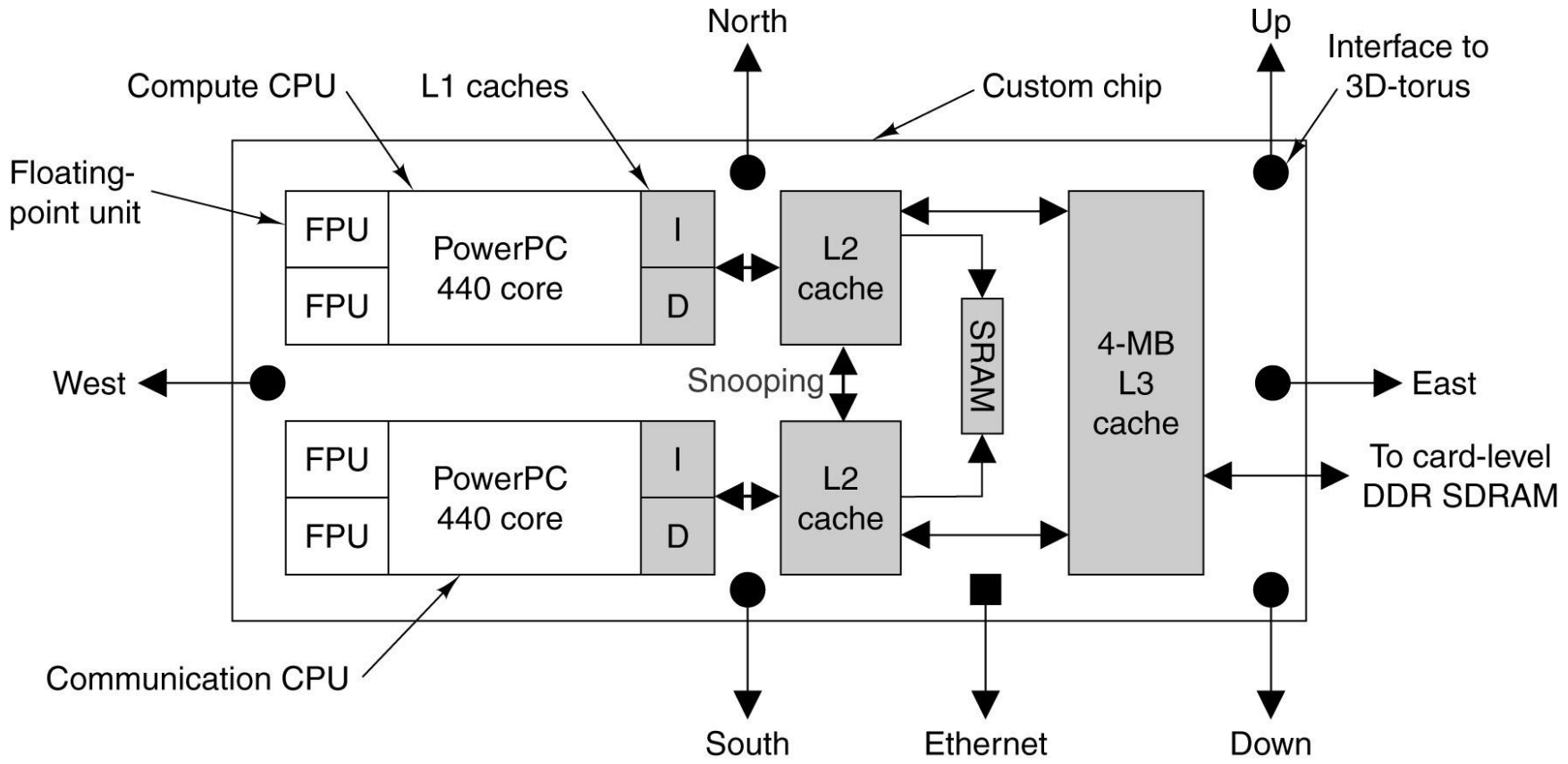
MPP – *Massively Parallel Processors*
BlueGene

- Sistema projetado pela IBM, em 1999, para resolver problemas de computação intensiva relacionados à vida
- Primeiro: BlueGene/L
  - Início em 2001
  - Uma parceria da IBM com o Departamento de Energia dos EUA
  - Projetado não só ser o MPP mais rápido, mas também para ser o mais eficiente em termos de teraflops/dolar, teraflops/watt, teraflops/m^3
  - Um quarto (16384 processadores) disponível em nov/2004
  - http://domino.research.ibm.com/comm/research_projects.nsf/pages/bluegene.index.html

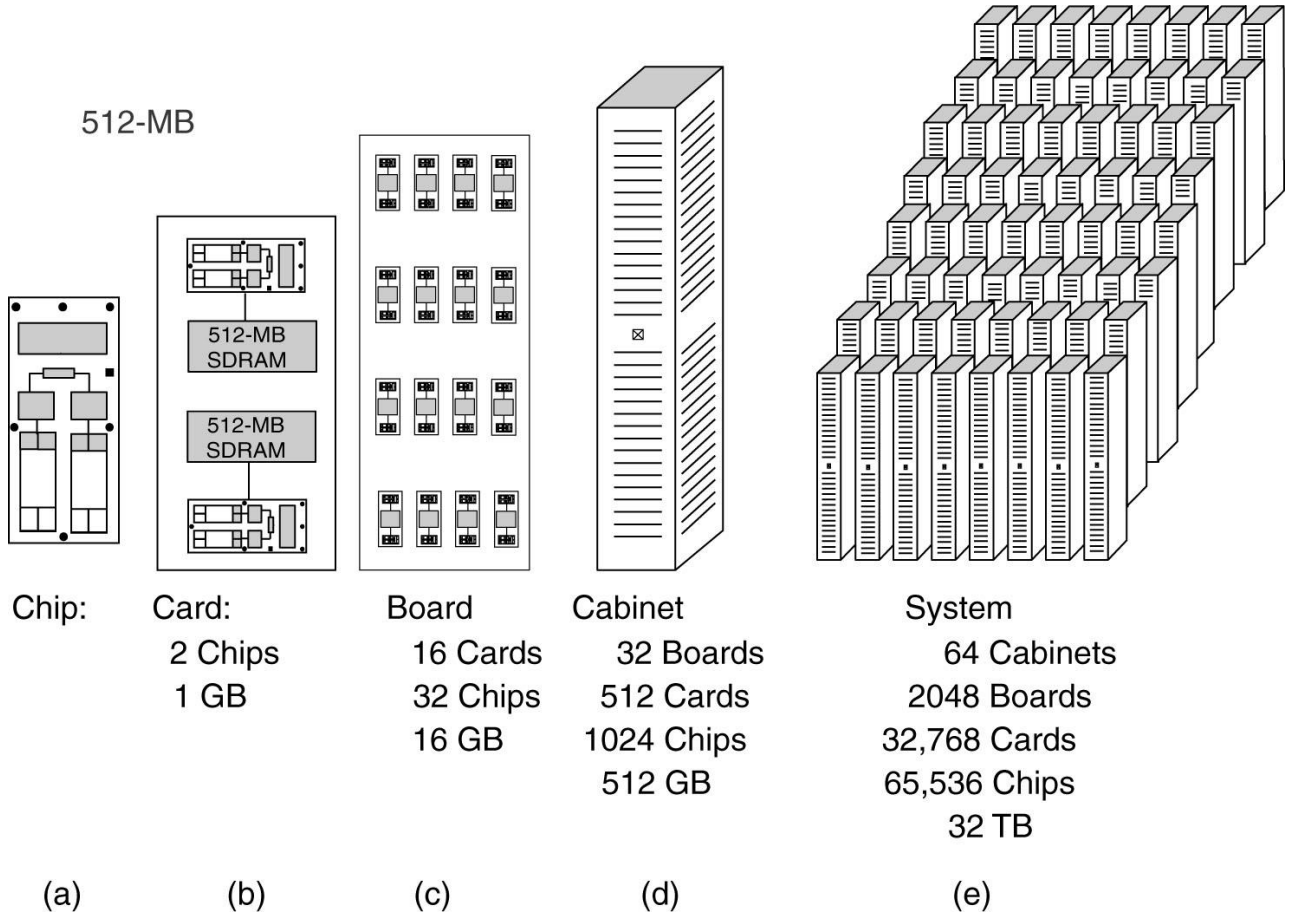
MPP – *Massively Parallel Processors*
BlueGene

- O “coração” do BlueGene é um chip que contém dois cores do processador PowerPC 440, a 700 MHz
  - PowerPC 440 é um processador pipeline com 2 unidades superescalares

MPP – *Massively Parallel Processors*
BlueGene



MPP – *Massively Parallel Processors*
BlueGene



The BlueGene/L. (a) Chip. (b) Card. (c) Board. (d) Cabinet. (e) System.

MIMD - Clusters

- Arquitetura alternativa às SMPs
- Um grupo de computadores conectados
  - trabalham juntos como um recurso unificado
  - fornecem a ilusão de ser uma máquina paralela
  - cada computador é chamado de nó
- Benefícios:
  - Alto desempenho
  - Escalabilidade incremental
  - Alta disponibilidade
  - Boa relação custo/desempenho

MIMD - Clusters

- A unidade básica do cluster é um único computador, também chamado de nó.
- Os cluster podem aumentar de tamanho pela adição de outras máquinas.
- O cluster como um todo será mais poderoso quanto mais rápidos forem os seu computadores individualmente e quanto mais rápida for a rede de interconexão que os conecta.

MIMD - Clusters

- Um cluster típico possui:
  - Rede mais rápida e próxima do que uma rede local;
  - Protocolos de comunicação de baixa latência;
  - Conexão mais frouxa que um SMP

MIMD - Clusters

- Se você tiver dois ou mais computadores, existe uma grande chance de que, em um determinado instante, pelo menos um deles não esteja fazendo nada.
- E infelizmente, quando você realmente precisar de capacidade de processamento, toda aquele que estiver disponível provavelmente não será suficiente.
- A idéia por detrás do uso de clusters é espalhar as cargas entre todos os computadores disponíveis, usando ao máximo os recursos que estão livres nas outras máquinas.

MIMD - Clusters

- Além disso, o sistema operacional de um cluster deve fazer o melhor uso do hardware disponível em resposta às mudanças de condições da computação.
- Isto será um grande desafio se o cluster for composto de diferentes tipos de computador (um cluster “heterogêneo”) , se um grande número de máquinas deixar e entrar no cluster aleatoriamente e se as cargas não puderem ser previstas com antecipação.

MIMD - Clusters

- Dois tipos de clusters podem ser considerados:
  - Centralizados
    - Cluster de estações de trabalho ou PCs, instalados em um *rack* em uma sala
    - Normalmente são máquinas homogêneas e não possuem outros periféricos que não placas de rede e discos
    - Chamados de COW (*Cluster of Workstation*)

MIMD - Clusters

- Tipos de Clusters (cont.)
  - Descentralizados
    - Estações de trabalho ou PCs espalhados por prédios, campus, cidade, etc.
    - Muitos ficam ociosos por boa parte do dia
    - Normalmente são conectados por LAN e são heterogêneos
    - Também conhecido como *grid* (grades)

MIMD - Clusters

- Existem 3 tipos de clusters:
  - Tolerante à falhas
  - Balanceamento de Carga
  - Computação de Alto Desempenho
- Clusters Tolerantes à Falhas consistem de dois ou mais computadores conectados em rede com um software de monitoração (*heartbeat*) instalado entre os dois
  - Assim que uma máquina falhar, as outras não mais receberação as respostas das mensagens de *heartbeat* e tentam assumir o trabalho

MIMD - Clusters

- Cluster com Balanceamento de Carga utilizam o conceito de, por exemplo, quando um pedido chega para um servidor Web, o cluster verifica qual a máquina menos carregada e envia o pedido para esta máquina.
- Na realidade na maioria das vezes um cluster com balanceamento de carga é também um cluster tolerante à falha com a funcionalidade extra de balanceamento de carga e um número maior de nós.

MIMD - Clusters

- A última variação de cluster é o de alto desempenho: as máquinas são configuradas especialmente para oferecer o maior desempenho possível.
- Estes tipos de clusters também tem algumas funcionalidades para balanceamento de carga, já que eles tentam espalhar os processos por máquinas diferentes para obter maior desempenho.
- Mas o que ocorre normalmente é que um processo é paralelizado e que as rotinas (ou threads) é que podem executar em paralelo em máquinas diferentes.

MIMD - Clusters

- Os supercomputadores tradicionais foram construídos por um pequeno número de fabricantes, com um alto orçamento destinado ao projeto.
- Muitas universidades não podem arcar com os custos de um supercomputador, então o uso de clusters se torna um alternativa interessante.
- Com o uso de hardware mais barato e disponível no mercado, sistemas com desempenho similar aos supercomputadores podem ser construídos.

MIMD - Clusters

- O desempenho dos componentes dos PCs e estações de trabalho é próximo do desempenho daqueles usados nos supercomputadores:
  - Microprocessadores
  - Redes de Interconexão
  - Sistemas Operacionais
  - Ambientes de Programação
  - Aplicações
- A taxa de melhoria de desempenho dos componentes ao longo do tempo é muito alta.

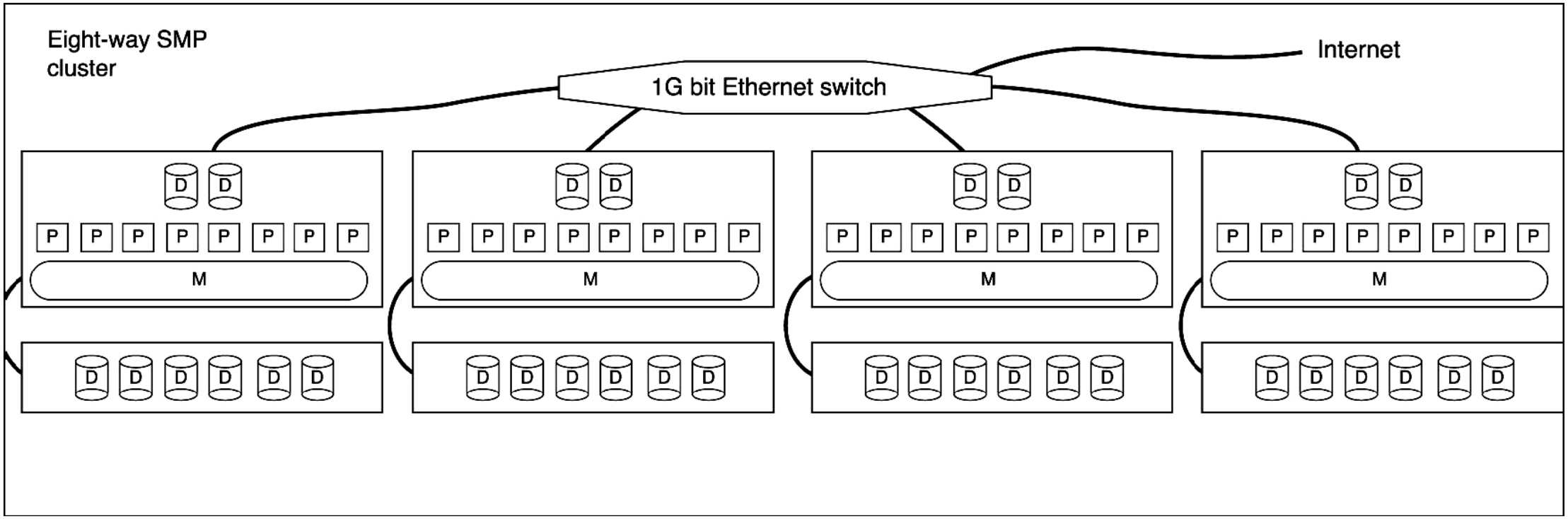
MIMD - Clusters

- Hardware
  - Plataformas
    - PCs (Intel x86):
      - Desktop
      - Servidores
    - Estações de Trabalho:
      - Alpha
      - IBM Power
    - SMPs
      - Xeon
    - Clusters de Clusters

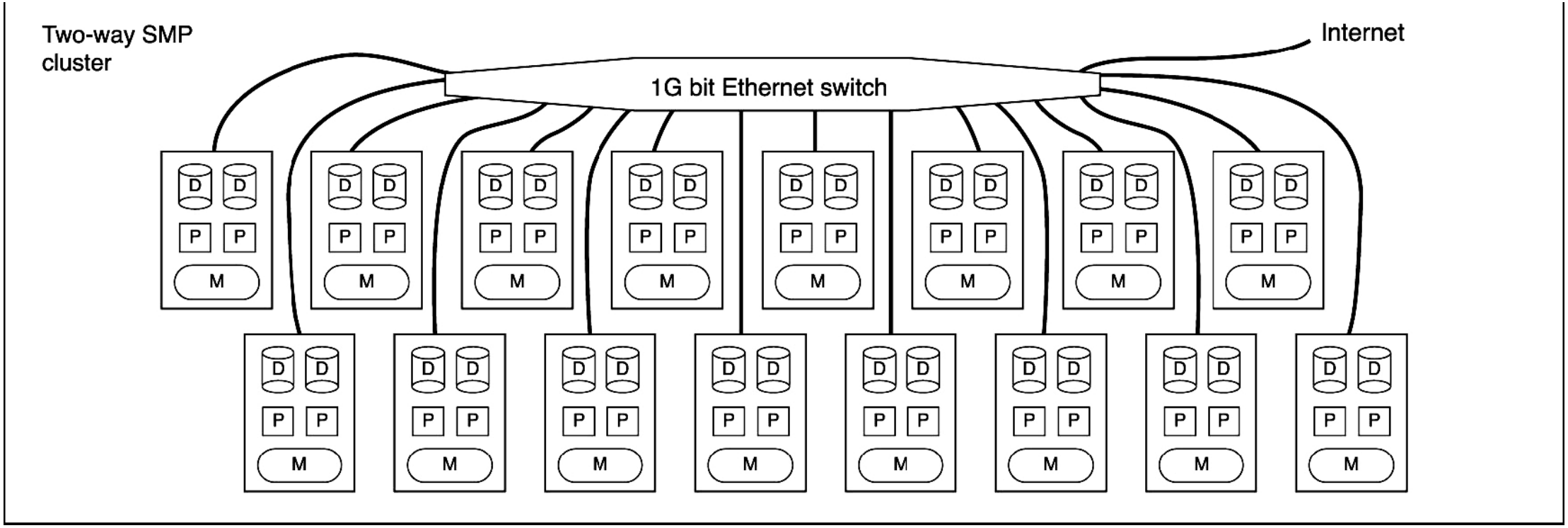
MIMD - Clusters

- Hardware
  - Redes de Interconexão
    - Fast Ethernet (100 Mbps)
    - Gigabit Ethernet (1 Gbps)
    - Myrinet (2 Gbps)
    - Ethernet 10 Gbps
    - Quadrics QSNet
    - Mellanox Infiniband (10 Gbps)
    - SCI (Dolphin – MPI – 12 µs latência)
    - ATM
    - Digital Memory Channel
    - FDDI

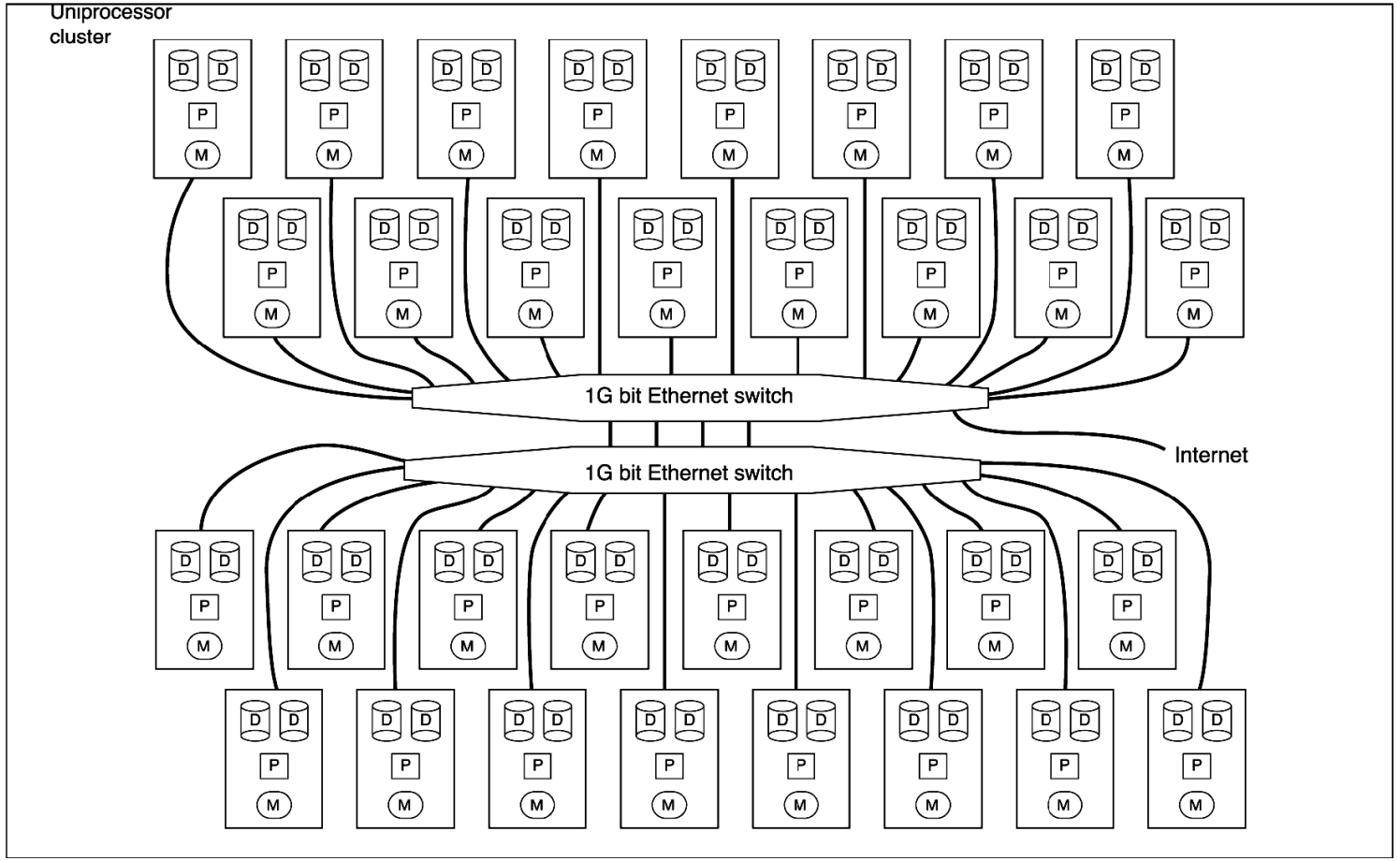
MIMD - Clusters



MIMD - Clusters



MIMD - Clusters

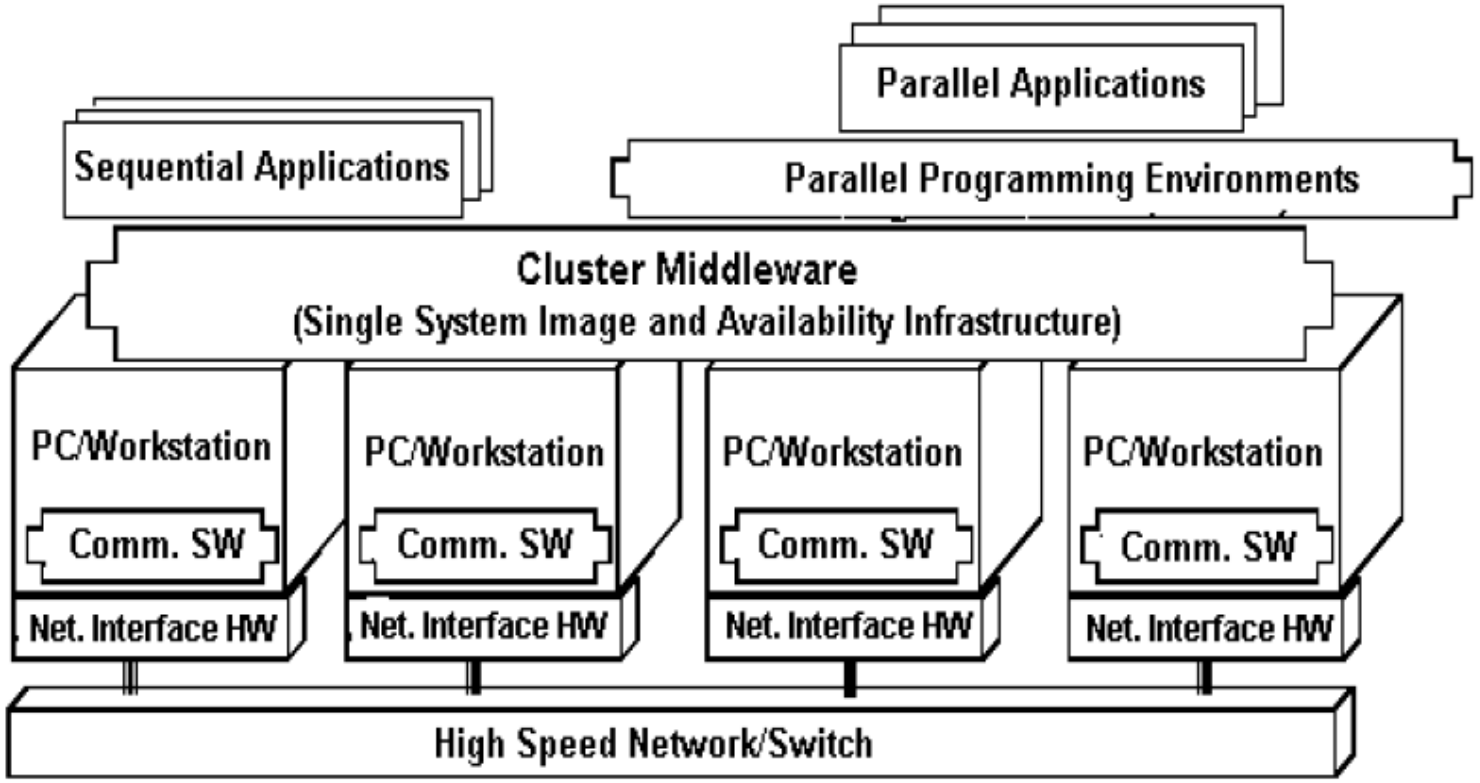


MIMD - Clusters

- Software de Comunicação
  - As facilidades tradicionais também são suportadas (mas são pesadas devido ao protocolo de processamento):
    - Soquetes (TCP/IP), Pipes, etc.
  - Protocolos mais leves são mais adequados (Comunicação no nível de usuário):
    - Active Messages (AM) (Berkeley)
    - Fast Messages (Illinois)
    - U-net (Cornell)
    - XTP (Virginia)
    - Virtual Interface Architecture (VIA)

MIMD - Clusters

- Arquitetura de um Cluster



MIMD - Clusters

- Maiores desafios:
  - Escalabilidade (física e da aplicação)
  - Disponibilidade (gerenciamento de falhas)
  - Imagem única do sistema (faz o usuário perceber que é um único sistema)
  - Comunicação rápida (redes e protocolos de comunicação)
  - Balanceamento de carga (CPU, Rede, Memória, Discos)
  - Segurança e Encriptação (clusters de clusters)
  - Programabilidade (administração e controle)
  - Aplicabilidade (aplicações voltadas para o cluster)

MIMD - Clusters

- Cluster Midleware
  - Reside entre o SO e aplicações e oferece infra-estrutura para suportar:
    - Imagem Única do Sistema (SSI – *Single System Image*)
    - Disponibilidade do Sistema (SA – *Sistem Avaliability*)
  - O SSI faz uma coleção de máquinas parecer como um recurso único (visão globalizada dos recursos do sistema)
  - O SA são pontos de verificação e migração de processos

MIMD - Clusters

- Sistemas Operacionais
  - Solaris MC
  - Unixware
  - MOSIX
  - Rocks
- Sistemas de execução
  - Sistemas em Execução
    - Software DSM (*Distributed Shared Memory*), PFS (*Parallel File System*), etc.
  - Gerenciamento de Recursos e Escalonamento (RMS)
    - Codine, Condor, LSF, PBS, NQS, etc.

MIMD - Clusters

- Ambientes de Programação
  - Threads (PCs, SMPs, NOW)
    - POSIX Threads
    - Java Threads
  - MPI
  - PVM
  - Software DSMs
    - SHMEM da Cray/SGI

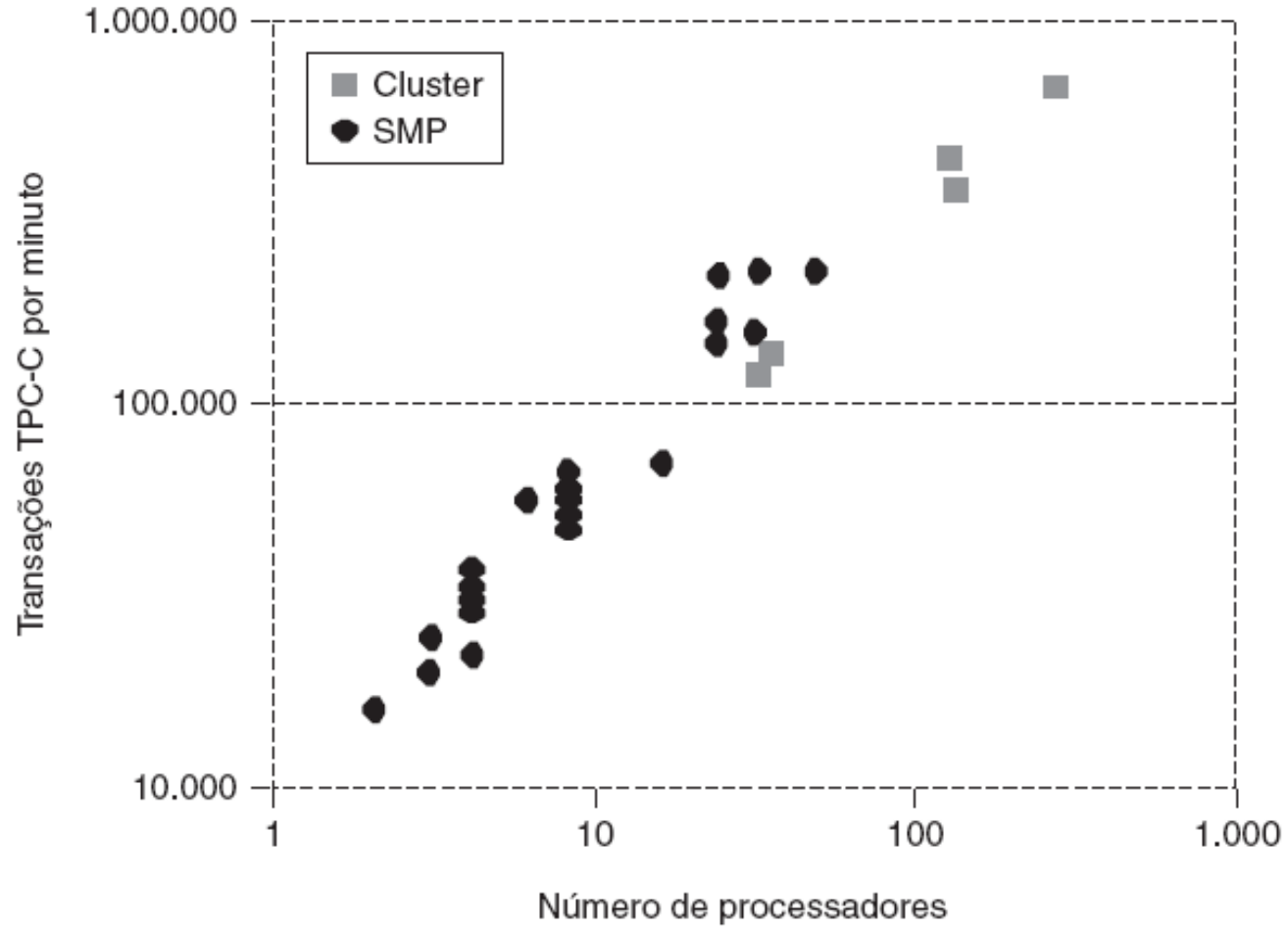
MIMD - Clusters

- Aplicações
  - Podem ser sequenciais quando se beneficiam do balanceamento de carga
  - Podem ser paralelas/distribuídas quando se utilizam dos ambientes de programação existentes
  - Servidores Web
  - Data-Mining

Cluster x SMP

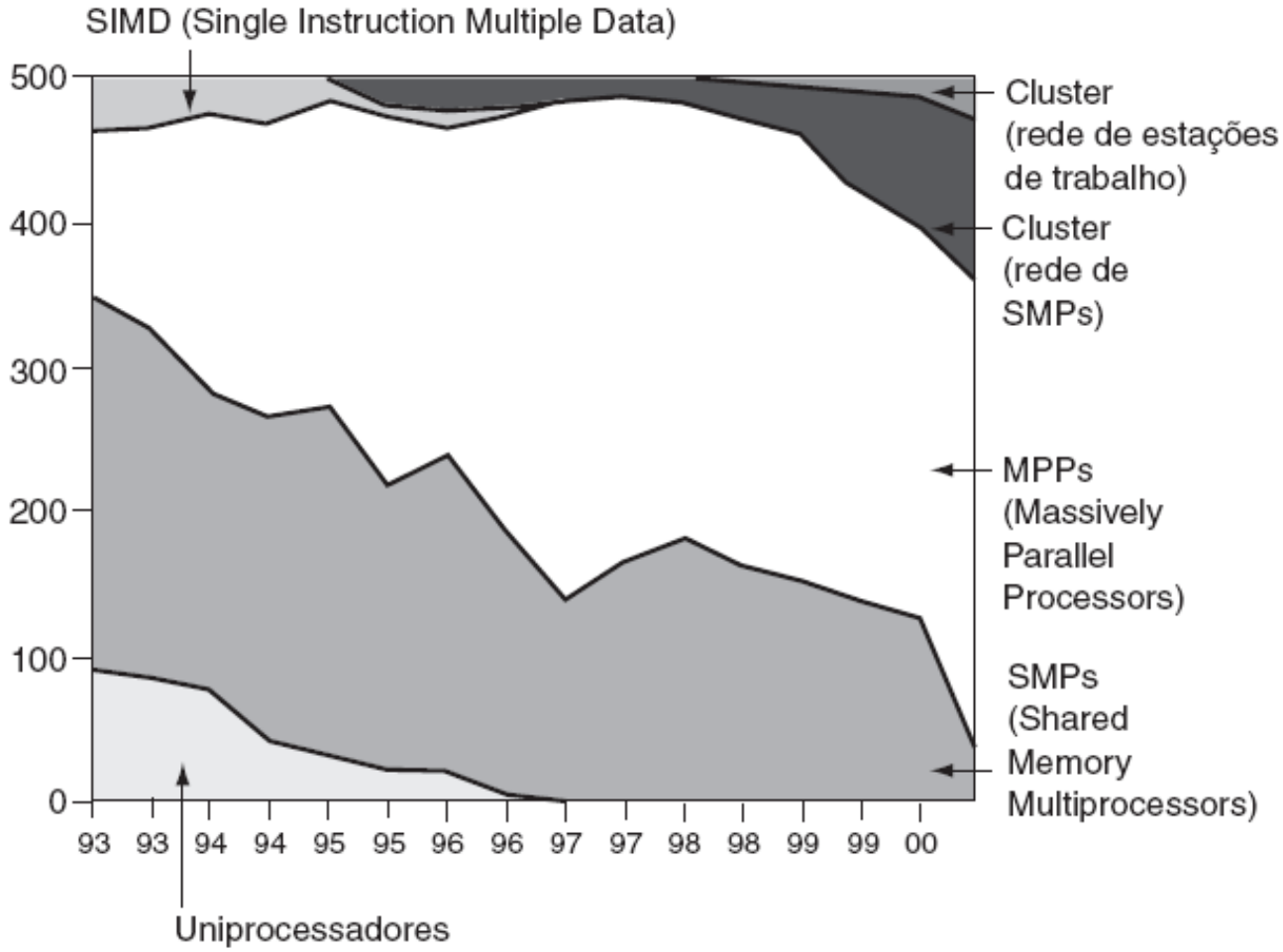
- Ambas fornecem suporte de multiprocessadores para aplicações com alta demanda
- Ambas estão disponíveis comercialmente
  - SMP é mais antiga
- SMP:
  - mais fácil para gerência e controle
  - mais próxima das arquiteturas monoprocessadas
    - escalonamento é a principal diferença
    - menos espaço físico e menor consumo de energia
- *Clusters* :
  - capacidade de crescimento superior
  - melhor disponibilidade
    - maior redundância

Comparação de desempenho
Cluster x SMP utilizando o benchmark TPC-C



Fonte: Organização e Projeto de Computadores (Patterson & Hennesy), 2003

Tipos de máquinas no Top500 por ano



Fonte: Organização e Projeto de Computadores (Patterson & Hennesy), 2003

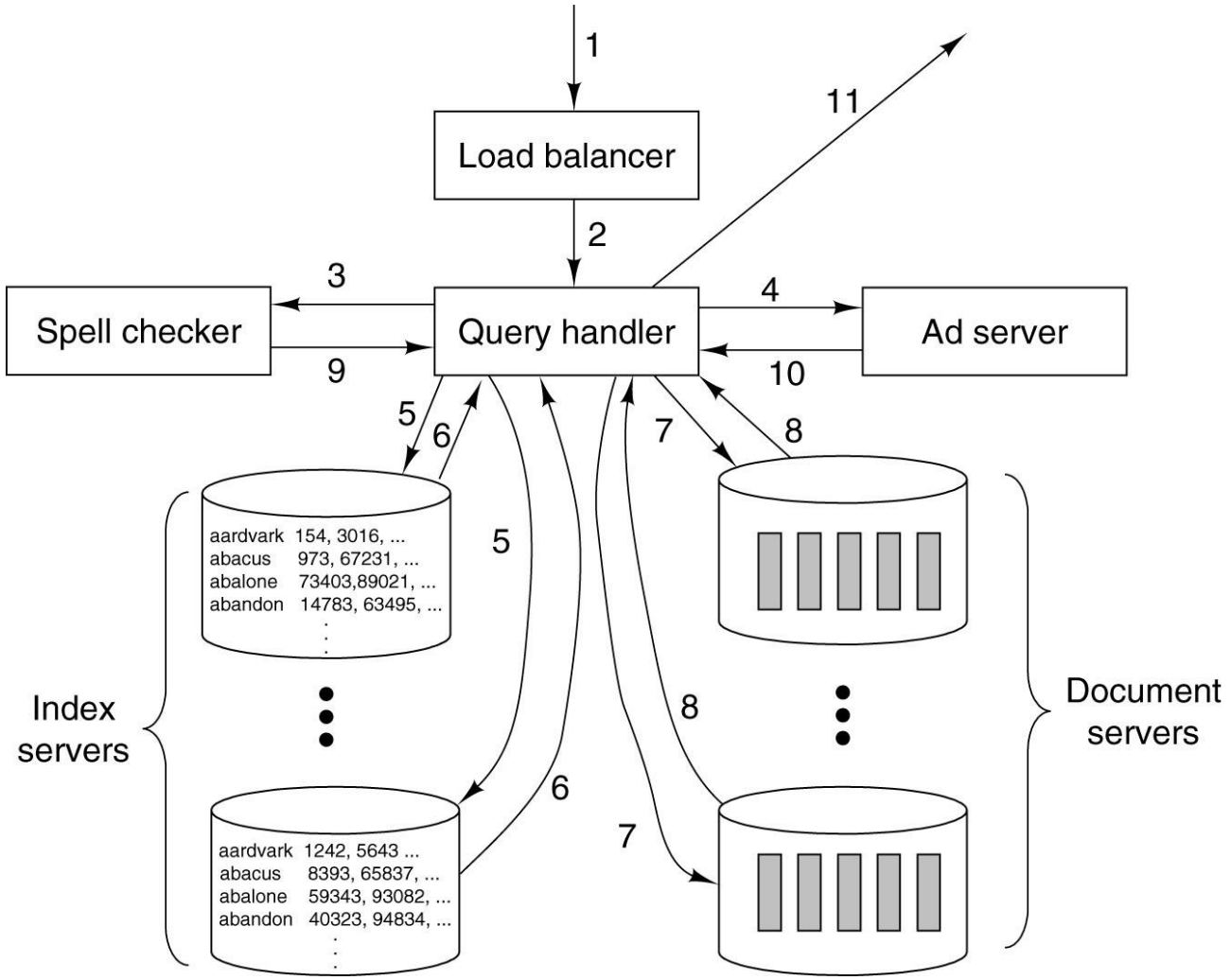
Google

- Mecanismo de busca de informação na Internet
- Muito utilizado devido à sua fácil interface e rápido tempo de resposta
- Para isso, o Google tem que:
  - procurar, indexar e armazenar toda a WWW
  - ser capaz de procurar tudo em menos de 0,5 segundos
  - manipular centenas de pesquisas/seg vindas de todas as partes do mundo
  - nunca falhar, nem mesmo com terremotos, falhas elétricas, de hardware e das operadoras telefônicas, e falhas de software

Google

- A maior preocupação no projeto da arquitetura do Google foi utilizar computadores com uma excelente relação custo/desempenho.
- Isto não significa, necessariamente, o computador com processador mais avançado para um dado momento.
- A confiabilidade é provida a nível de software e não no hardware.
- O projeto procurou paralelizar os pedidos individuais como forma de obter o melhor *throughput* agregado.

Google


1
Load balancer
2
11
3
Spell checker
9
Query handler
4
Ad server
10
5
6
7
8
aardvark 154, 3016, ...
abacus 973, 67231, ...
abalone 73403,89021, ...
abandon 14783, 63495, ...
5
8
6
7
aardvark 1242, 5643 ...
abacus 8393, 65837, ...
abalone 59343, 93082, ...
abandon 40323, 94834, ...
Index servers
Document servers

Google

- Ao fazer uma pergunta para o Google, o navegador do usuário deve primeiro fazer a conversão do DNS para um endereço IP em particular.
- Para fazer frente à quantidade de tráfico, o serviço Google consiste de diversos clusters espalhados geograficamente.
- Um sistema de balanceamento escolhe um cluster levando em conta a sua proximidade geográfica do usuário com cada cluster.
- Um balanceador de carga em cada cluster monitora a disponibilidade do servidores e realiza balaceamento local de carga.

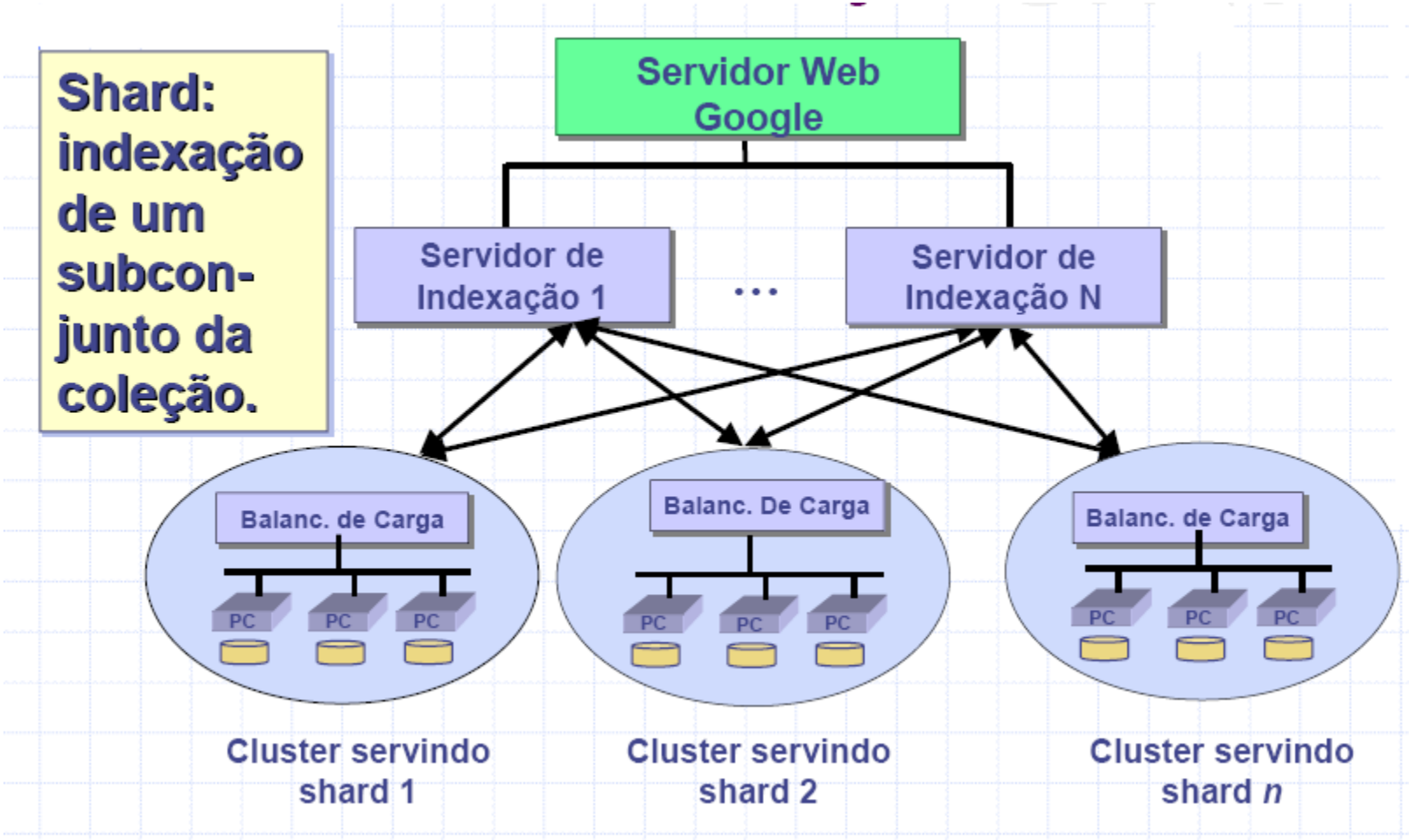
Google

- Uma execução de uma resposta se dá em duas fases:
  - Os servidores de índice consultam uma tabela invertida que mapeia cada palavra da pergunta para uma lista de documentos correspondentes.
  - Os servidores de índice determinam um conjunto de documentos relevantes pela interseção das listas individuais de cada palavra da pergunta e computam um índice de relevância para cada documento.

Google

- A busca dos índices é paralelizada dividindo-o em partes chamadas “index shards”, cada uma contendo um subconjunto de documentos do índice completo.
- Existem várias cópias de cada “shard” espalhadas pelo cluster, com um conjunto específico de máquinas servindo a cada uma delas.
- Cada pedido escolhe uma máquina dentro de um conjunto usando um balanceador de carga intermediário.
- Em outras palavras, cada pedido vai para uma máquina (ou um subconjunto) atribuído a cada “shard”.

Google


Shard: indexação de um subcon-junto da coleção.
Servidor Web Google
Servidor de Indexação 1
...
Servidor de Indexação N
Balanc. de Carga
PC
PC
PC
Balanc. De Carga
PC
PC
PC
Balanc. de Carga
PC
PC
PC
Cluster servindo shard 1
Cluster servindo shard 2
Cluster servindo shard n

Google

- O resultado final da primeira fase de busca é uma lista ordenada de identificadores de documentos (docids).
- A segunda fase da computação envolve pegar a lista de docids e computar a URL e o título real de cada um desses documentos.
- Os servidores de documentos (docservers) realizam esta fase da computação.
- A estratégia utilizada também é a de dividir o processamento em diversas etapas.

Google

- Distribuindo aleatoriamente os documentos em “shards” menores.
- Tendo multiplas cópias de servidores responsáveis para cada “shard”.
- Roteando pedidos através de um balanceador de carga.
- O cluster de servidor de documentos deve ter acesso “on-line” e de baixa latência a uma cópia com o conteúdo de toda a Web.

Google

- Utiliza mais de 6.000 processadores e 12.000 discos, totalizando 1 petabyte de armazenamento
- Ao invés de RAID, o Google se baseia em sites redundantes, cada um com milhares de discos e processadores:
  - Dois sites no Vale do Silício
  - Dois sites na Virgínia

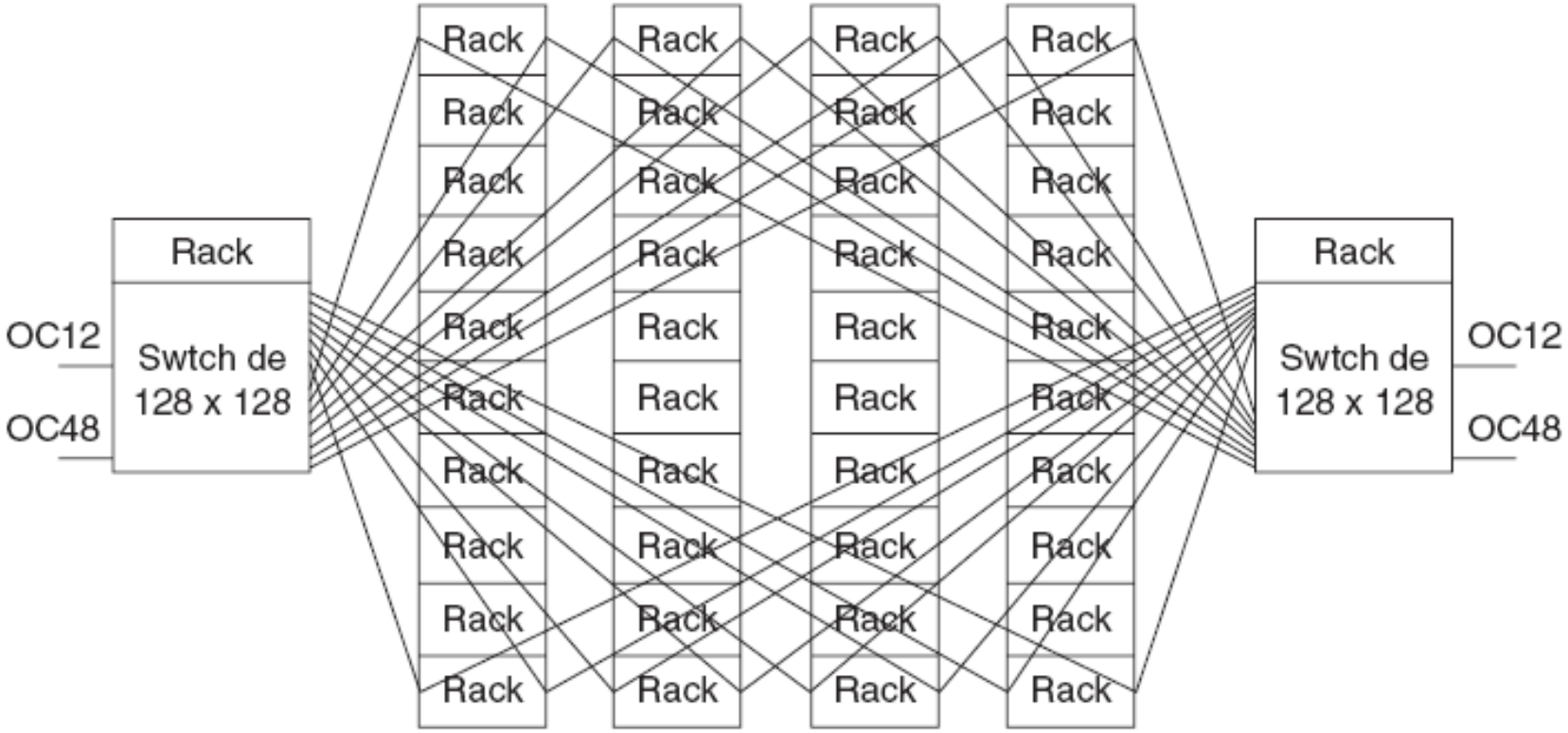
Google – Princípios de Projeto

- Confiabilidade por software
  - Não é feito o uso de fontes de alimentação redundantes, nem de RAIDs, nem de componentes de alta qualidade.
- Uso de replicação para melhor throughput e disponibilidade
  - Cada um dos serviços é replicado em muitas máquinas
- Preço/desempenho acima do desempenho de pico
  - São compradas gerações de CPU que no momento oferecem o melhor desempenho por unidade de preço, ao invés do maior desempenho absoluto.
- PC´s de mercado reduzem o custo da computação
  - Como resultado podem ser utilizados mais recursos computacionais para cada pedido.

Google – Princípios de Projeto

- Cada rack consiste de 40 a 80 servidores X86 montados em ambos os lados de um rack personalizado.
- Em dez/2002 havia diversas gerações de processadores em serviço, desde Celerons de 500 Mhz até servidores duais com Pentium III de 1.4 Ghz.
- Cada servidor contém um ou mais discos IDE de 80 Gb e 2 GB de memória.
- Os servidores em ambos os lados do rack se interconectam via um switch ethernet de 100 Mbits, que se conecta via um ou dois links a um switch gigabit que interconecta todos os “racks” entre si.

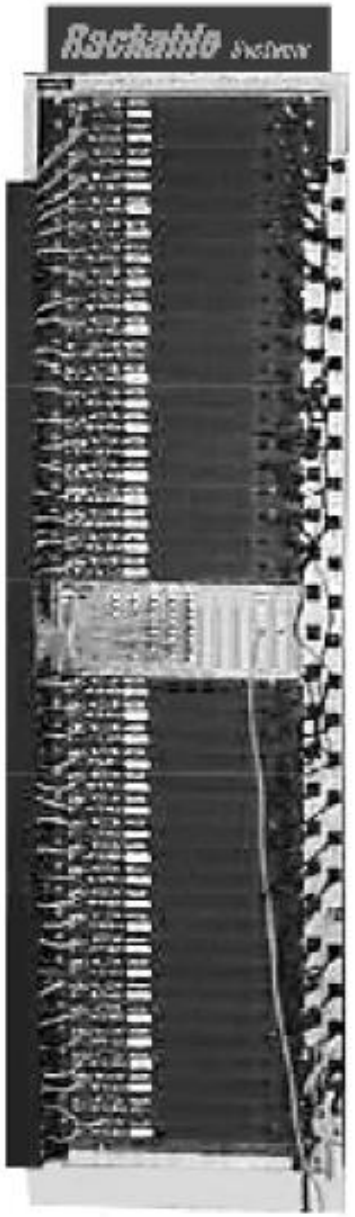
Google – Princípios de Projeto



Cada switch pode se conectar a 128 racks a 1 Gbit/seg
Racks de PCs, cada um com 4 interfaces Ethernet de 1 Gbit/seg, são conectados aos 2 switches Ethernet de 128 x 128

Ligação OC48 (2488 Mbits/s)

Google
Princípios de Projeto



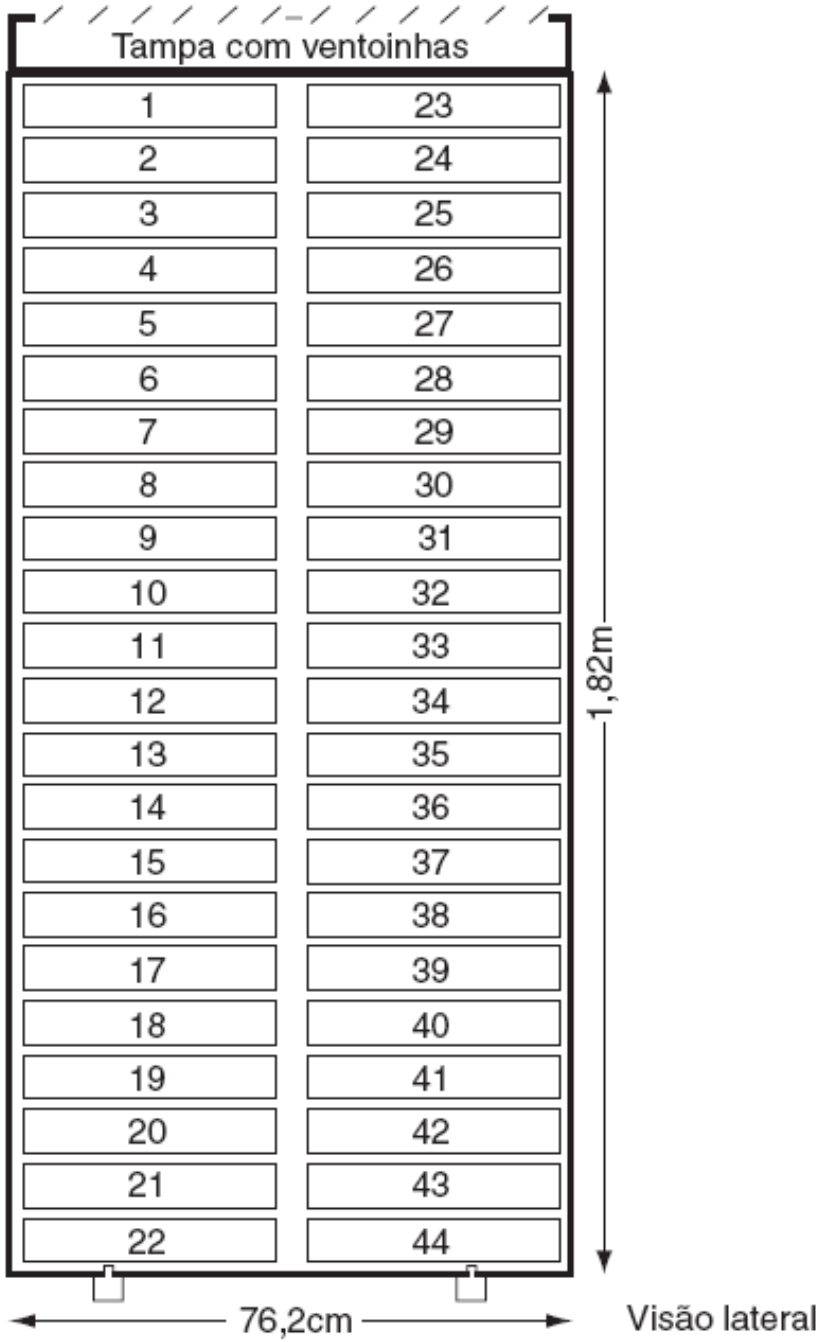
Visão frontal (também visão de trás)

48,26cm

Google – Princípios de Projeto

Visão em close dos PCs de 1 RU

Google
Princípios de Projeto



Material de consulta

- Artigo:
  - Web Search for a Planet: The Google Cluster Architecture

    Luiz Andre Barroso, Google
    Jeffrey Dean, Google
    Urs Holzle, Google

    IEEE Micro

    Vol 23, No. 2

Google Search hoje...

- Algumas ferramentas importantes:
  - GFS (Google File System)
  - MapReduce
  - BigTable

- Vocês estudarão isso na disciplina de Sistemas Distribuídos...

THE END
IS NEAR
HERE

thank you
danke
謝謝
ngiyabonga
teşekkür ederim
gracias
merci
obrigado
bedankt
dziękuję
go raibh maith agat
mochchakkeram
sukriya
kop khun krap
terima kasih
감사합니다
спасибо
tapadh leat
dank je
grazie
arigatô
dakujem
takk
taiku
sagolun
mauruuru
hvala
ありがとう
мерси
ευχαριστώ
Баярлалаа
рахмат
mersi
barka
tack
faafetai lava
vinaka
blagodaram
misaotra
matondo
paldies
grazzi
kiitos
dankie
dhanyavad
enkosi
sobodi
dekuji
mesi
najis tuke
chnorakaloutioun
gratias ago
gracies
sulpay
tanemirt
rahmet
diolch
dhanyavadagalu
shukriya
trugarez
mamnun
xвала
asante
manana
obrigada
djiere dieuf
дякую
xiexie
kam sah hamnida