Universidade Federal do Rio de Janeiro
Informática DCC/IM

Arquitetura de Computadores II

Clusters

Gabriel P. Silva

O que é um Cluster?

- Um cluster é um tipo de sistema distribuído ou paralelo que consiste de uma coleção de computadores interconectados usados como um único e integrado recurso de computação.
- Um cluster é um tipo de sistema de processamento paralelo que consiste de uma coleção de computadores independentes interconectados através de uma rede, trabalhando cooperativamente como um único e integrado recurso computacional.

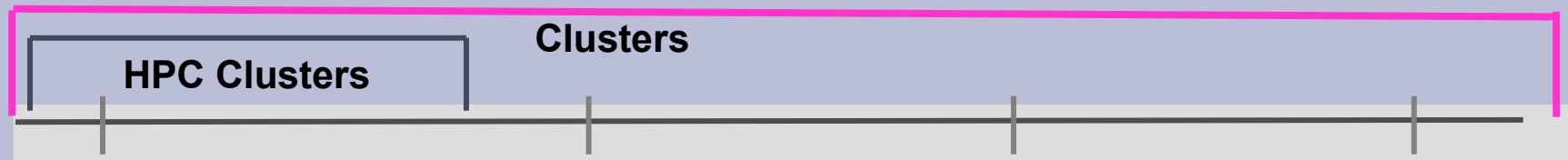
Clusters

- 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 seus computadores individualmente e quanto mais rápida for a rede de interconexão que os conecta.

Porquê o uso de Clusters

- **Permite levar a computação de alto desempenho para um domínio mais amplo de problemas.**
- **Uma ordem de magnitude em termos de preço/desempenho sobre computadores convencionais.**
- **Rápida resposta para as mudanças tecnológicas.**
- **Uso de sistemas operacionais, aplicações e ferramentas de software livre.**

Aplicabilidade do Cluster



Tightly-coupled Clusters

- Interconnected nodes
- Extensive inter-node communication
- Ultra-high speed links
- Operations performed on one node depend on output of operations on one or more other nodes in the cluster – parallelized applications
- Crash of single nodes impacts operation of entire cluster
- Local storage and NAS/SAN required
- SPOC for hardware/software management and software/application distribution and maintenance

Loosely-coupled Clusters

- Interconnected nodes
- Minimal to no inter-node communication
- High speed links
- Operations performed on one node are independent of operations on any other node – users or tasks distributed across nodes
- Crash of one node may impact performance of cluster, but operation of cluster continues
- Local storage and NAS/SAN required
- SPOC for hardware/software management and software/application distribution and maintenance

"Simple" Failover Clusters

- Two interconnected nodes
- Node-to-node communication limited to "heart-beat" only
- Only simple cable interconnect required
- Each node normally handles half the workload
- If one node crashes, the other node takes over its operations; significant performance impact
- Local storage and/or NAS/ SAN
- SPOC for hardware management; may support SPOC for software/application management and distribution

Single Purpose Server Farms

- Interconnected nodes
- No inter-node communication
- No special interconnect required
- Nodes operate independently, but all nodes dedicated to the same application/workload
- Crash of a node has no impact on operation of any other component in the farm
- Local storage and/or NAS/ SAN
- SPOC for hardware management only; may support SPOC for software/application management

Aplicabilidade do Cluster

Clusters

Multi-purpose Server Farms

- Interconnected nodes
- No inter-node communication
- No special interconnect required
- Nodes operate independently, and nodes run two or more applications/workloads
- Crash of a node has no impact on operation of any other component in the farm
- Local storage and/or NAS/SAN
- SPOC for hardware management only; may support SPOC for software/ application management

Standalone Networked Servers

- Interconnected nodes
- No inter-node communication
- Nodes operate independently
- Crash of one node has no effect on operation of any other node
- Local storage or NAS
- May or may not have SPOC for hardware management

Multiple Standalone Servers

- No connection between nodes
- Nodes operate independently
- Crash of one node has no effect on operation of any other node
- Local storage or NAS
- No SPOC

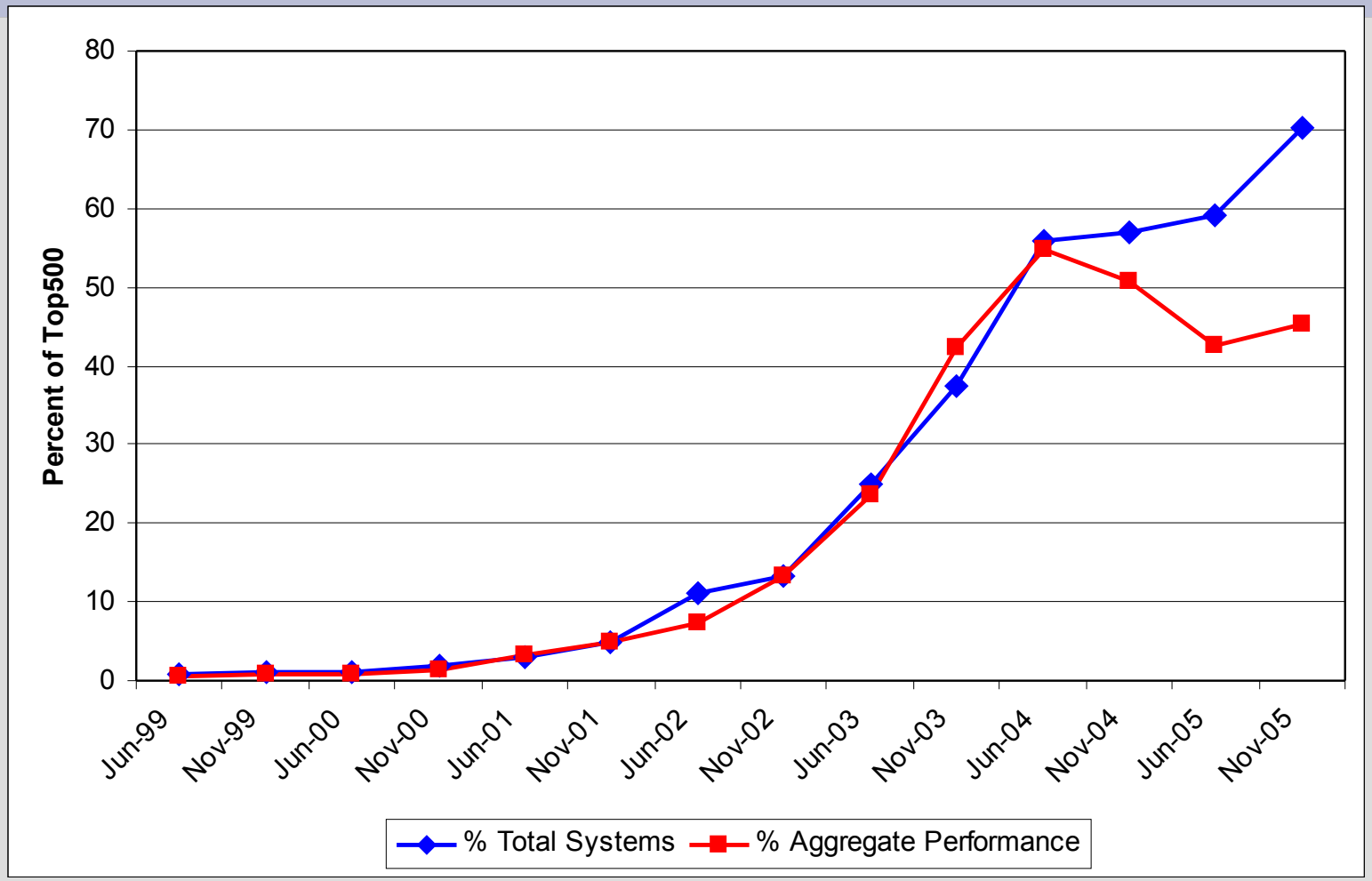
O que é o TOP500?

- O projeto TOP500 foi iniciado em 1993 para prover uma base confiável para registro e detecção de tendências em computação de alto desempenho.
- Duas vezes por ano é montada e divulgada uma lista dos sites indicando os 500 computadores mais rápidos do mundo.
- O melhor desempenho do programa de avaliação LINPACK é utilizado para resolver um sistema de equações lineares densas.

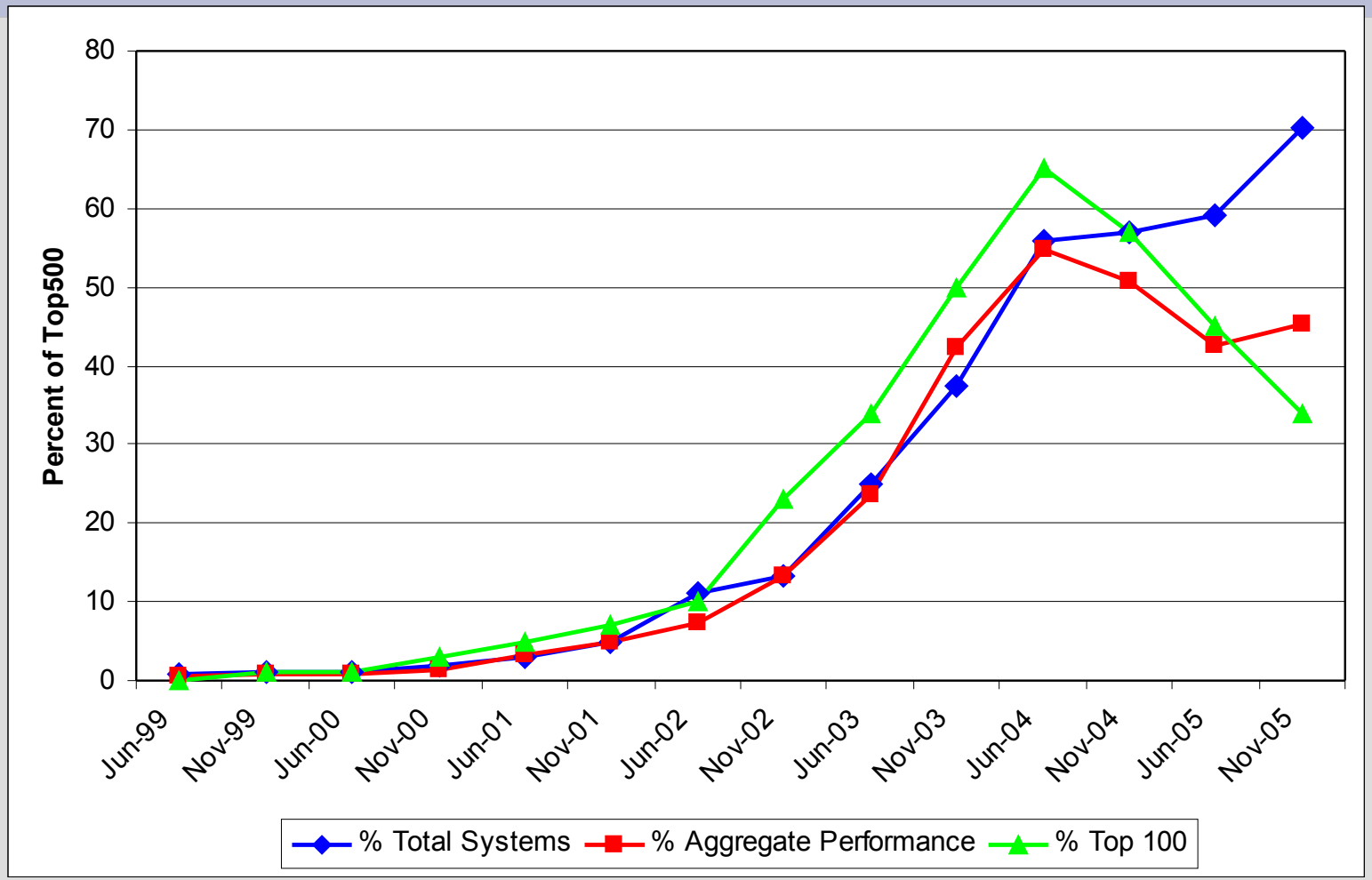
O que é o TOP500?

- O que é o LINPACK?
  - Os resultados obtidos com o programa de avaliação LINPACK não refletem o desempenho global do desempenho de um dado sistema, mas apenas o desempenho do sistema totalmente dedicado para resolver um sistema de equações lineares denso.
  -
- Porquê o LINPACK?
  - O LINPACK foi escolhido porquê é utilizado amplamente e os números de desempenho são disponíveis para a maioria dos sistemas.

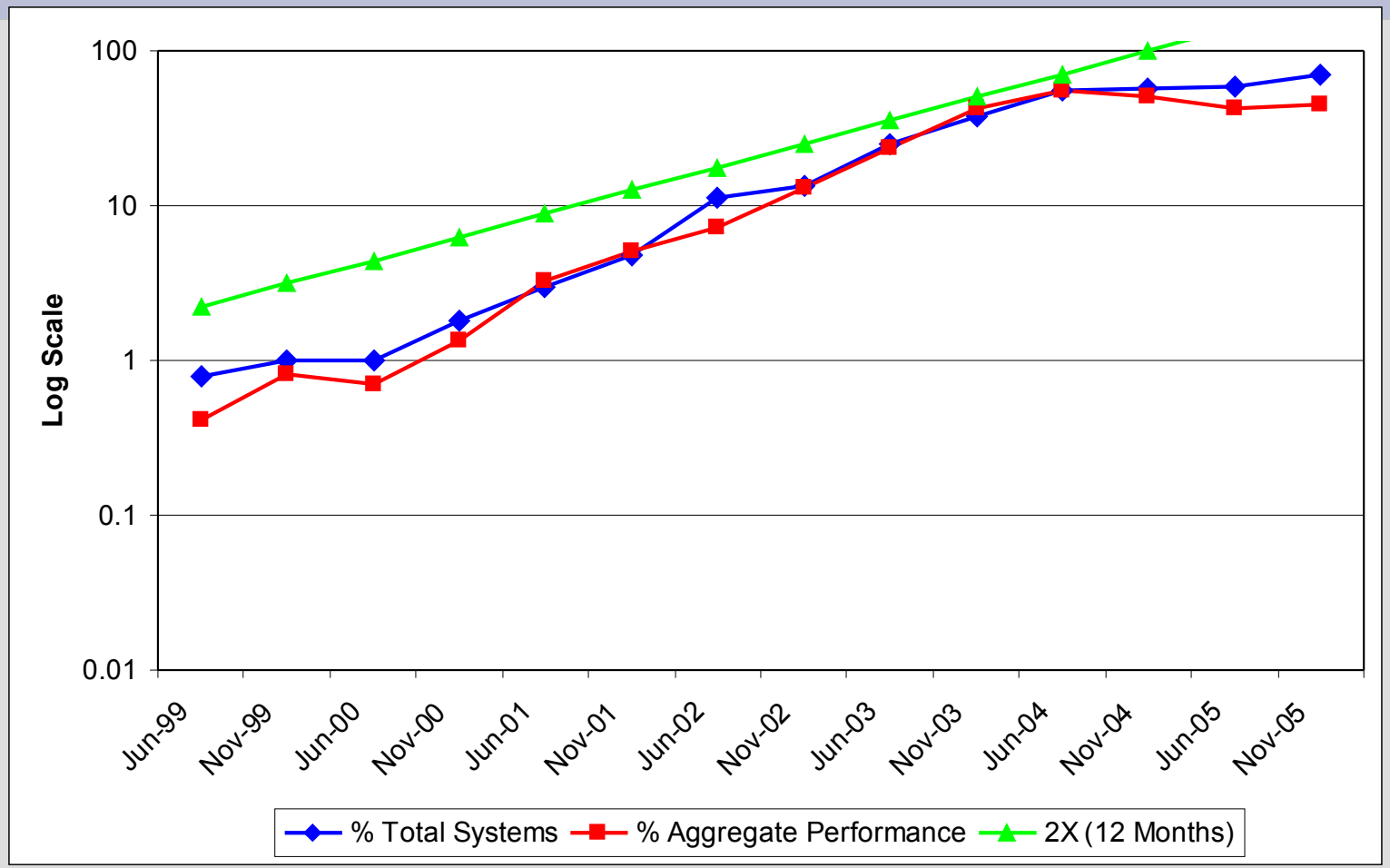
Clusters com Linux no Top500



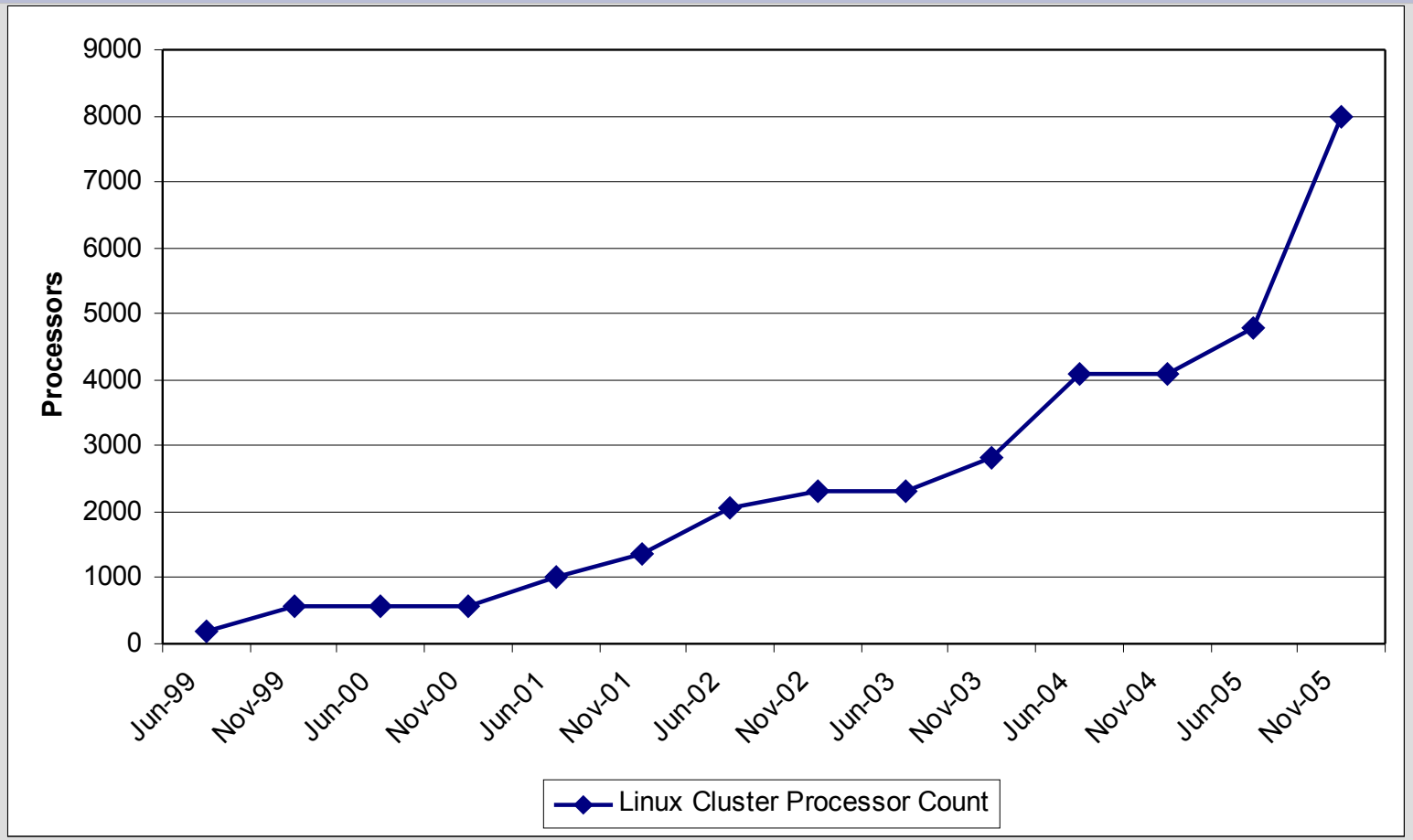
Clusters com Linux no Top100



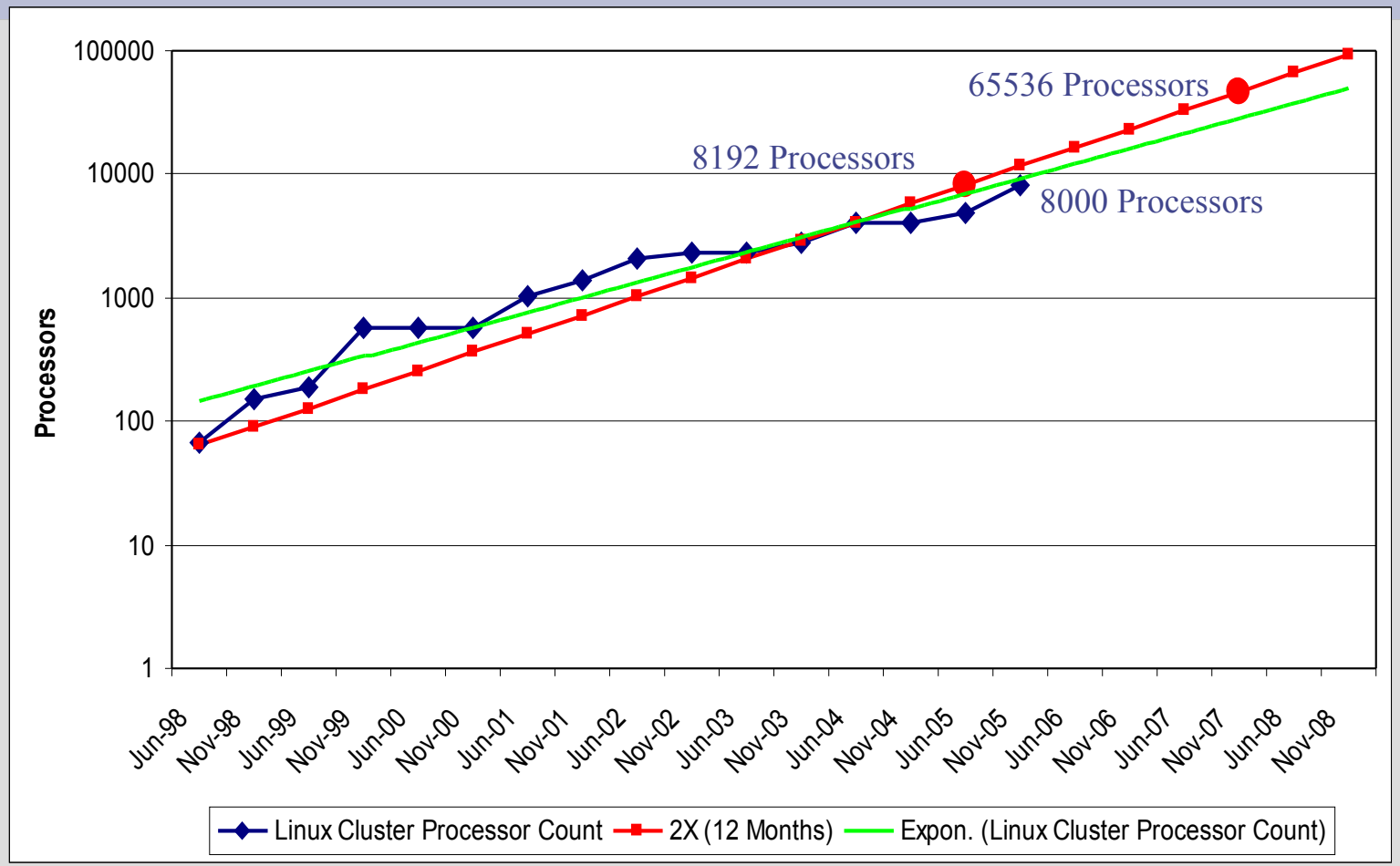
Crescimento do Clusters com Linux no Top500



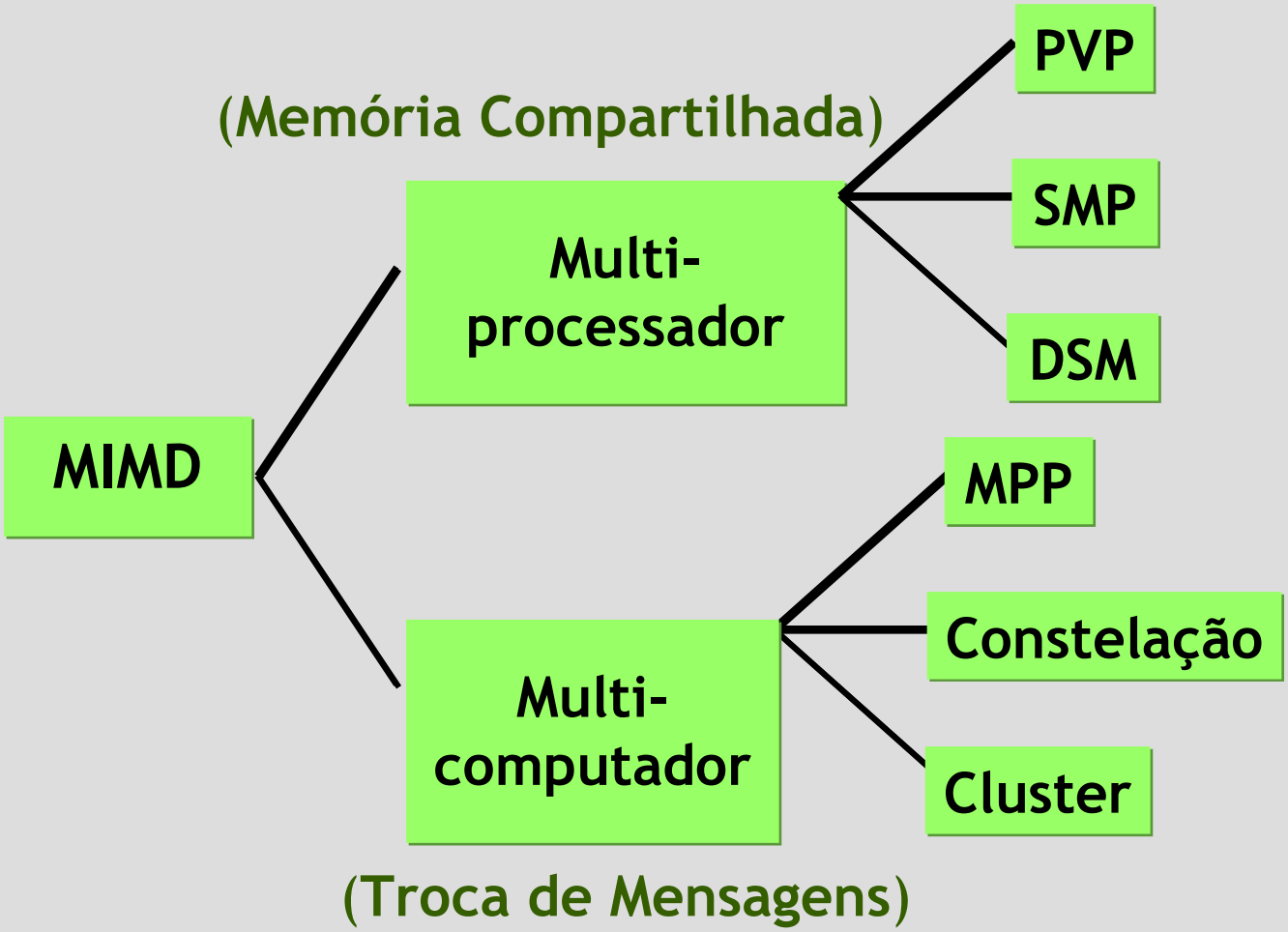
Tamanho dos Cluster com Linux no Top500



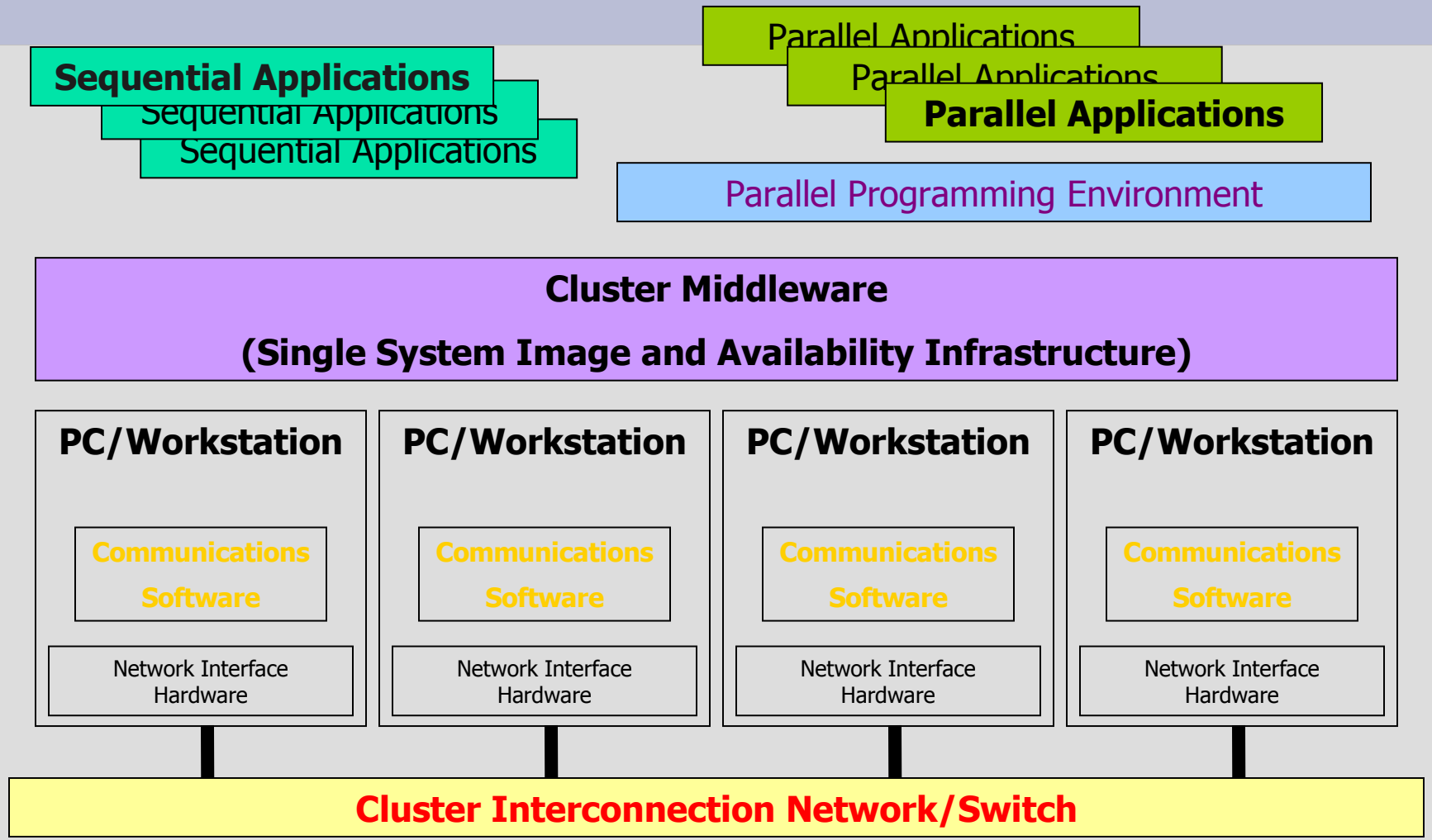
Projeções


100000
10000
1000
100
10
1
Processors
65536 Processors
8192 Processors
8000 Processors
Jun-98
Nov-98
Jun-99
Nov-99
Jun-00
Nov-00
Jun-01
Nov-01
Jun-02
Nov-02
Jun-03
Nov-03
Jun-04
Nov-04
Jun-05
Nov-05
Jun-06
Nov-06
Jun-07
Nov-07
Jun-08
Nov-08
Linux Cluster Processor Count
2X (12 Months)
Expon. (Linux Cluster Processor Count)

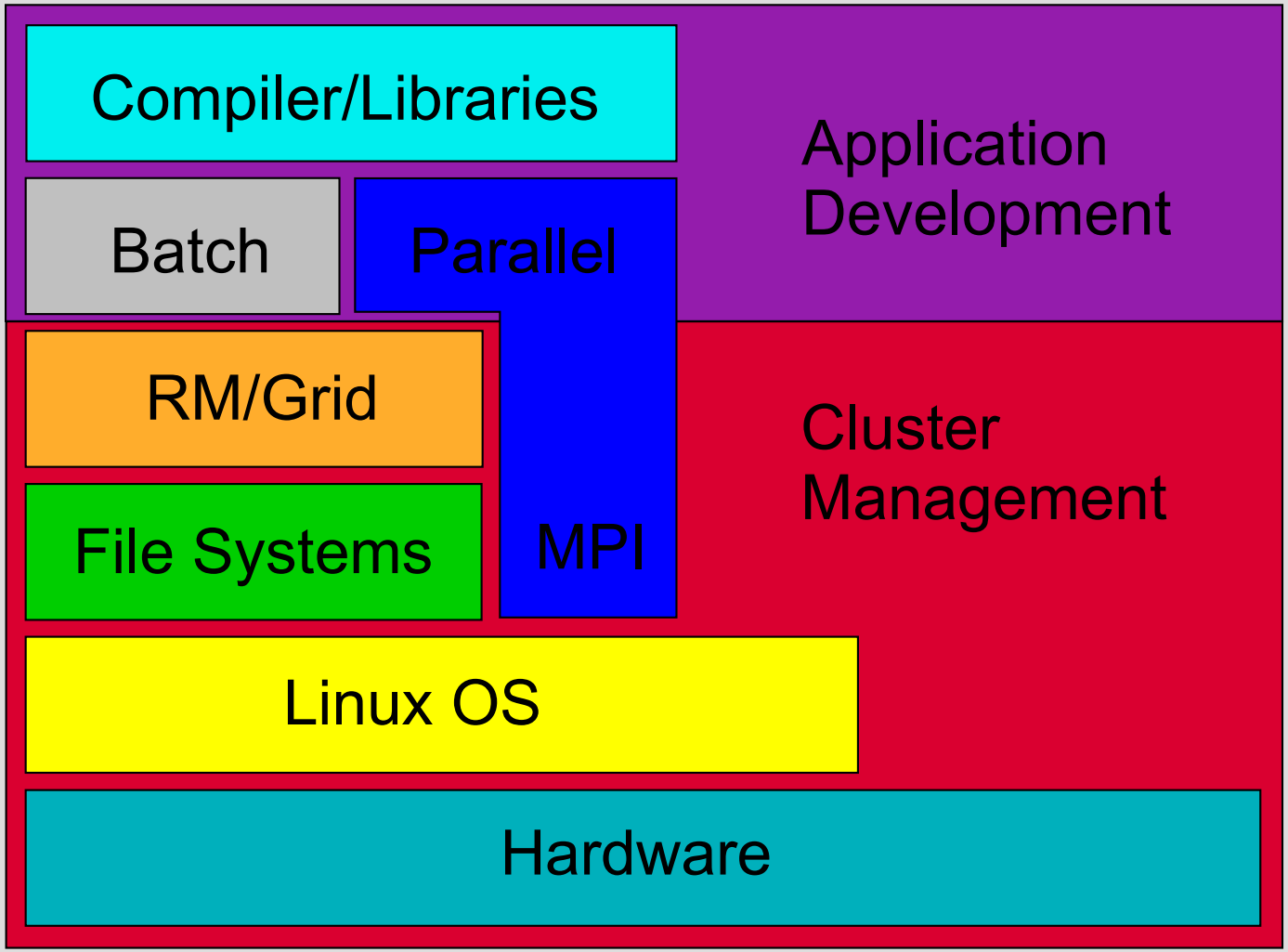
Arquitetura de Computadores Paralelos



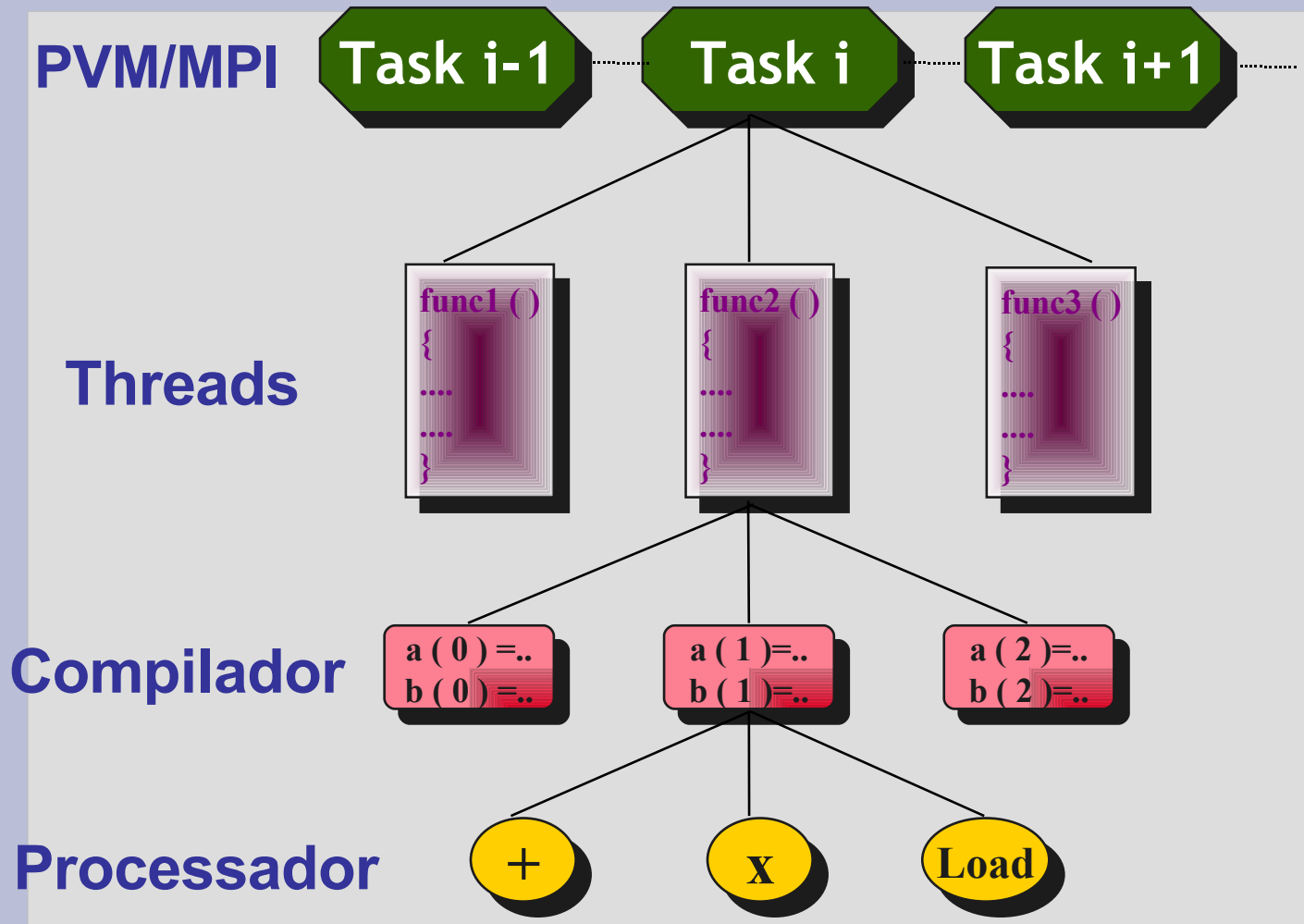
Arquitetura de um Cluster



Solução Completa



Níveis de Paralelismo



Ambientes de Programação

- Threads
  - Em sistemas multiprocessadores
    - Usado para utilizar simultaneamente todos os processadores disponíveis.
  - Em sistemas uniprocessadores
    - Usado para utilizar os recursos do sistema eficientemente.
  - Aplicações multithreaded oferecem resposta mais rápida ao usuário e executam mais rápido.
  - As aplicações são portáteis se escritos com interfaces padrões como pthreads e OpenMP.
  - Usadas extensivamente em desenvolver tanto aplicações como software de sistema.

Ambientes de Programação

- **Ambientes de Troca de Mensagem (MPI e PVM)**
  - **Permite que programas paralelos eficientes sejam escritos para sistemas de memória distribuída.**
  - **Os dois ambientes mais populares de troca de mensagem – PVM & MPI**
  - **PVM**
    - **Tanto um amiente quanto uma biblioteca de troca de mensagens.**
  - **MPI**
    - **Uma especificação de troca de mensagens padronizada.**
    - **Estabelece um padrão prático, portátil, eficiente e flexível para troca de mensagens.**
    - **Geralmente os desenvolvedores preferem MPI pois se tornou um padrão de fato para a troca de mensagens.**

Redes de Interconexão

- Fast Ethernet (100 Mbps)
- Gigabit Ethernet (1 Gbps)
- 10-Gigabit Ethernet (10 Gbps)
- Infiniband (10 Gbps (SDR) e 20 Gbps (DDR))
- Infiniband QDR (40 Gbps)
- Myrinet (2 Gbps)
- Quadrics QSNET I e II

Tipos de Clusters

- Basicamente 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 (heart-beat) instalado entre os dois.
- Assim que uma máquina falhar, as outras máquinas tentam assumir o trabalho.

Tipos de Clusters

- *Clusters 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.

Tipos de 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.

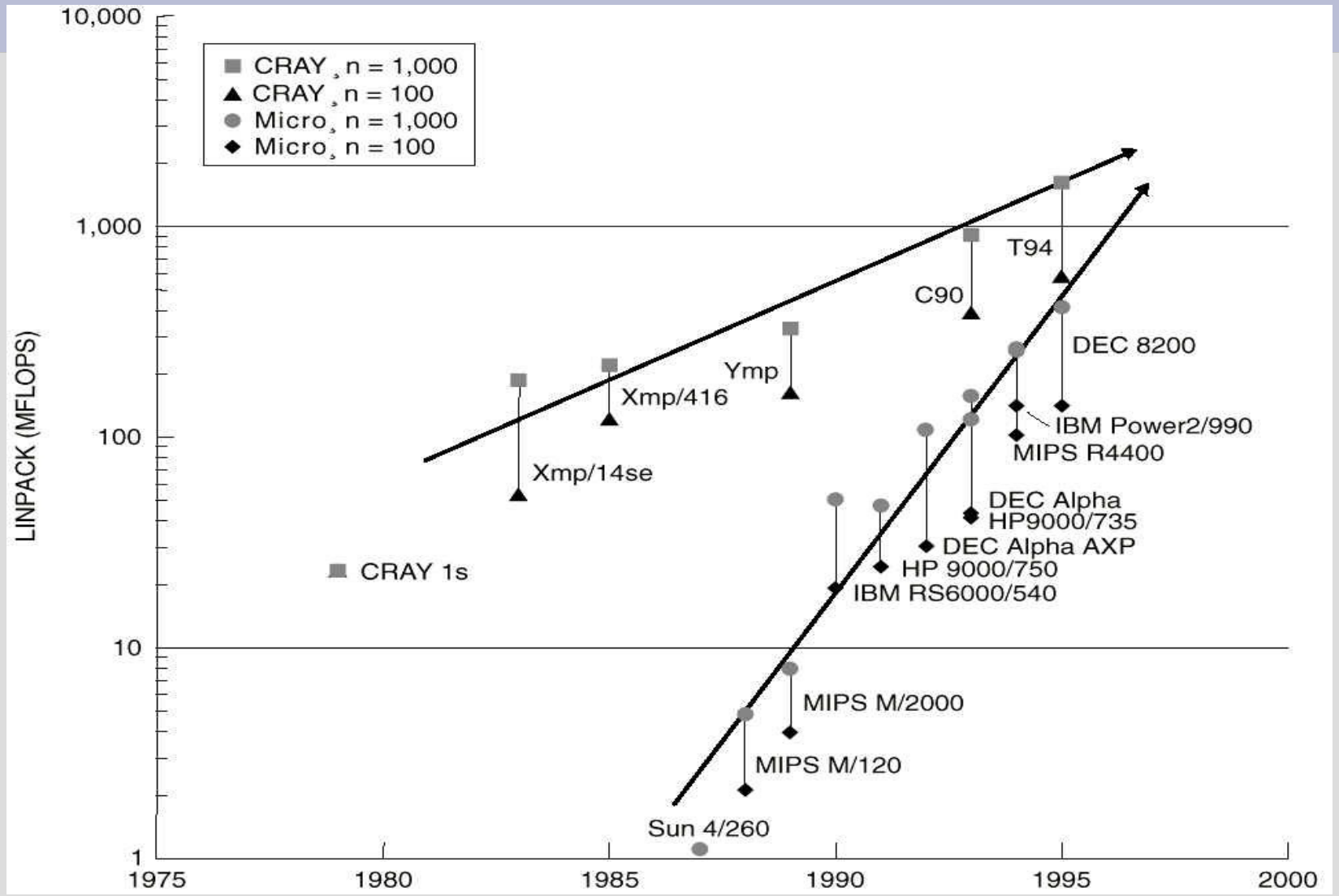
Clusters

- 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.

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.

Evolução

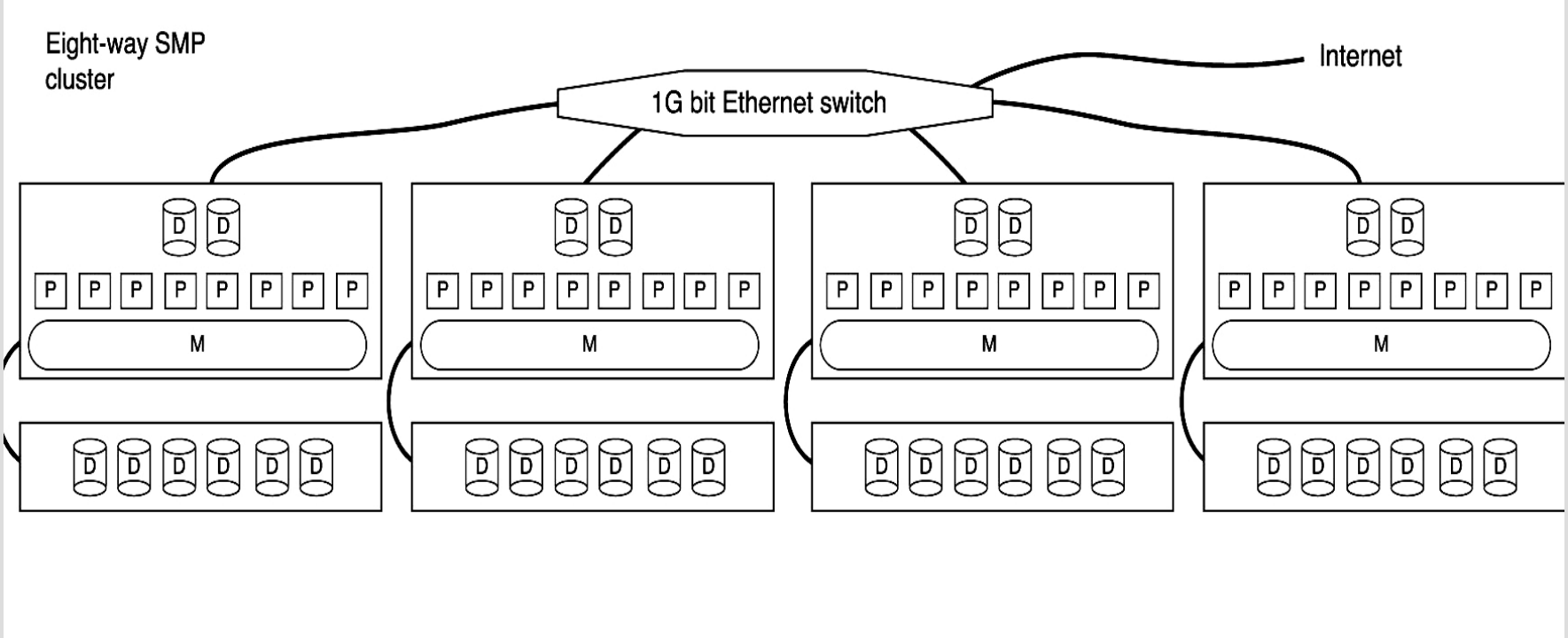

10,000
1,000
100
10
1
LINPACK (MFLOPS)
CRAY , n = 1,000
CRAY , n = 100
Micro, n = 1,000
Micro, n = 100
T94
C90
DEC 8200
Ymp
Xmp/416
IBM Power2/990
MIPS R4400
Xmp/14se
DEC Alpha
HP9000/735
DEC Alpha AXP
HP 9000/750
CRAY 1s
IBM RS6000/540
MIPS M/2000
MIPS M/120
Sun 4/260
1975
1980
1985
1990
1995
2000

Exemplo

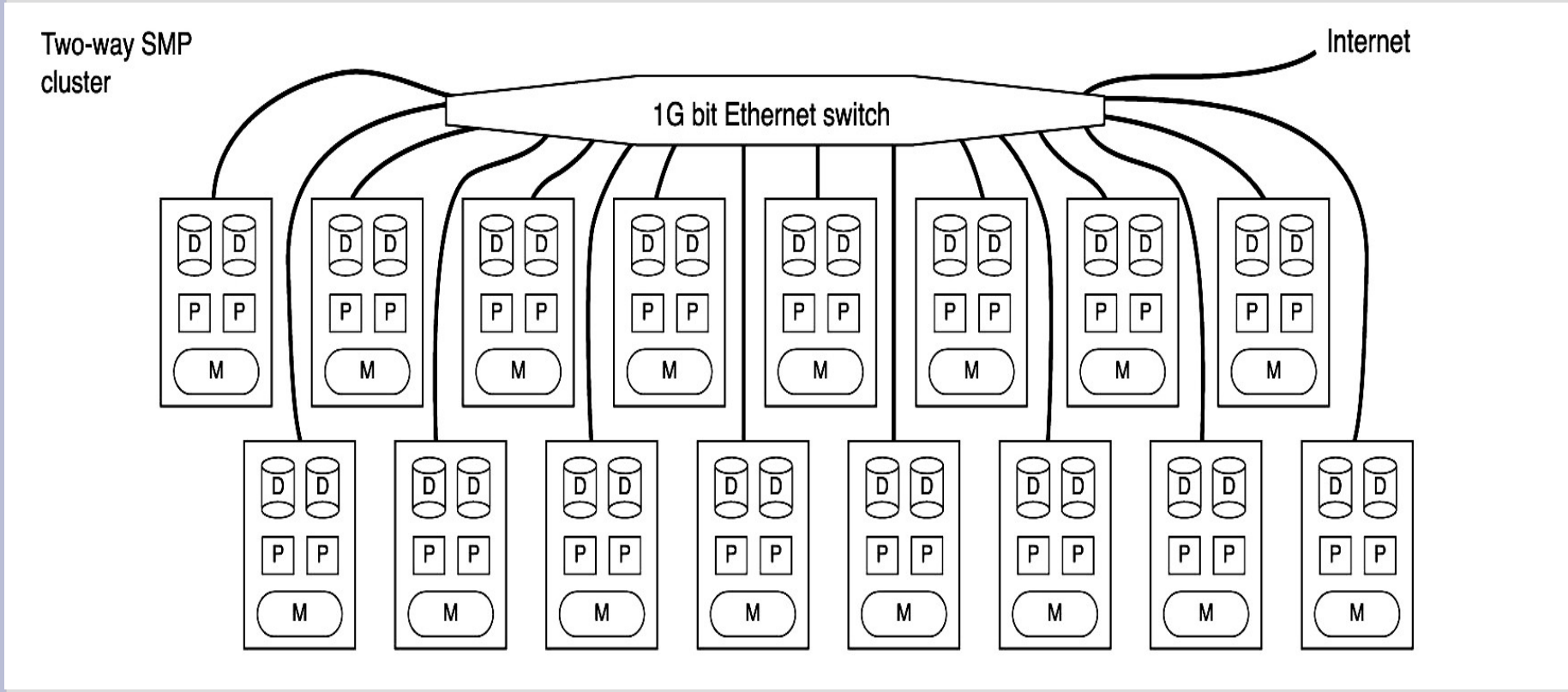
Hardware

- **Plataformas**
  - **PCs (Intel x86):**
    - **Desktop**
    - **Servidores**
      - **Singlecore**
      - **Multicore**
  - **Estações de Trabalho:**
    - **Alpha**
    - **IBM Power**
  - **Clusters de Clusters (Grids)**

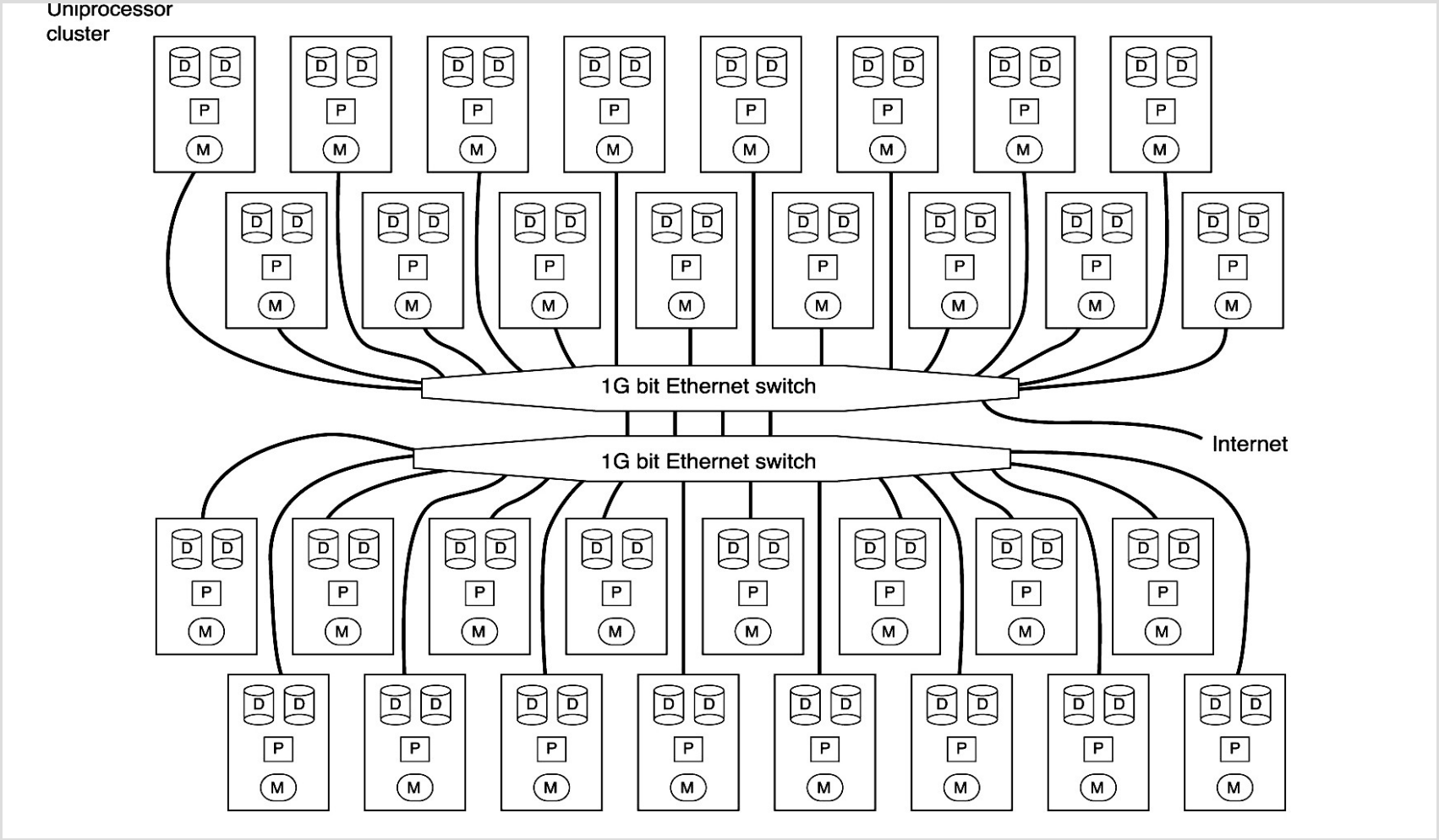
Cluster – SMP c/ 8 processadores



Cluster – SMP c/ 2 processadores



Cluster baseado em Monoprocessadores



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)

Maiores Desafios

- Escalabilidade (física e de aplicação)
- Disponibilidade (gerenciamento de falhas)
- Imagem Única do Sistema (parece ao usuário como um único sistema)
- Comunicação Rápida (redes e protocolos de comunicação)
- Balanceamento de Carga (CPU, Rede, Memória, Discos)

Maiores Desafios

- Segurança e Encriptação (clusters de clusters)
- Gerenciabilidade (admin. e controle)
- Programabilidade (API simples)
- Aplicabilidade (aplicações voltadas para o cluster)

Single System Image
SSI

Single System Image (SSI)

- Um cluster consiste de uma coleção de computadores convencionais interconectados entre si que podem atuar com um único e integrado recurso computacional.
- Cada nó de um cluster é um sistema completo tendo o seus próprios recursos de hardware e software.
- Contudo, eles oferecem ao usuário uma visão de um único sistema através de mecanismos de hardware e software comumente conhecidas como SSI.
- SSI é ilusão de um único sistema montado com recursos distribuídos.

Sistemas Operacionais

- **Existem duas variações de SSI:**
  - **Administração de Sistemas / Escalonamento de Tarefas – um "middleware" que habilita cada nó para prover os serviços requisitados.**
  - **Núcleo do S.O. – uso transparente de dispositivos remotos ou usar um sistema de armazenamento que é visto pelo usuários como um único sistema de arquivos.**

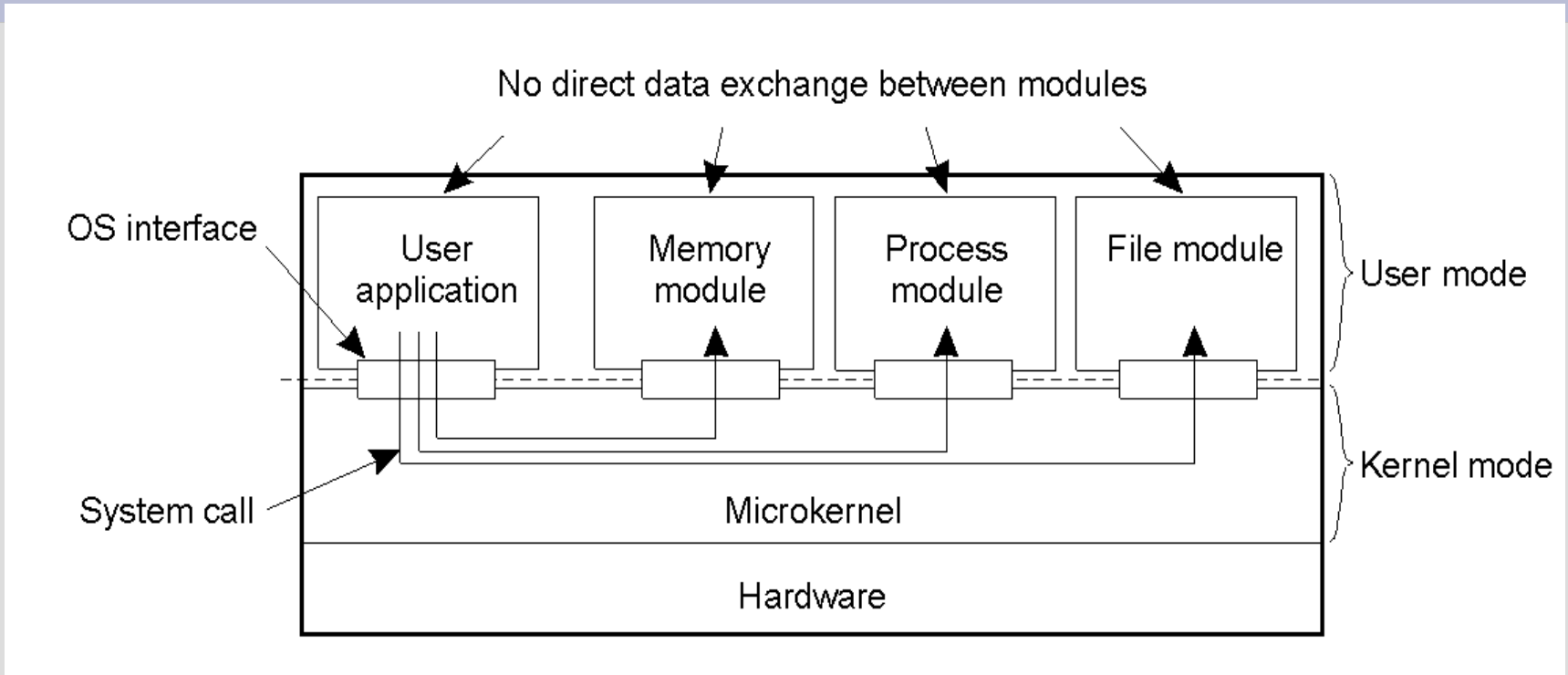
Exemplos – Solaris MC

- **Uma versão multicomputador do sistema operacional chamada de Solaris MC.**
  - **Incorpora alguns avanços, incluindo uma metodologia orientada a objeto e o uso do CORBA no núcleo.**
  - **Consiste de um pequeno conjunto de extensões ao núcleo e uma biblioteca de middleware – provê SSI ao nível dos dispositivos:**
    - **Processos executando em um nó podem fazer acesso a dispositivos remotos como se fossem locais, também fornece um sistema de arquivos globais e espaço de processo também global.**

Exemplos – Micro-kernels

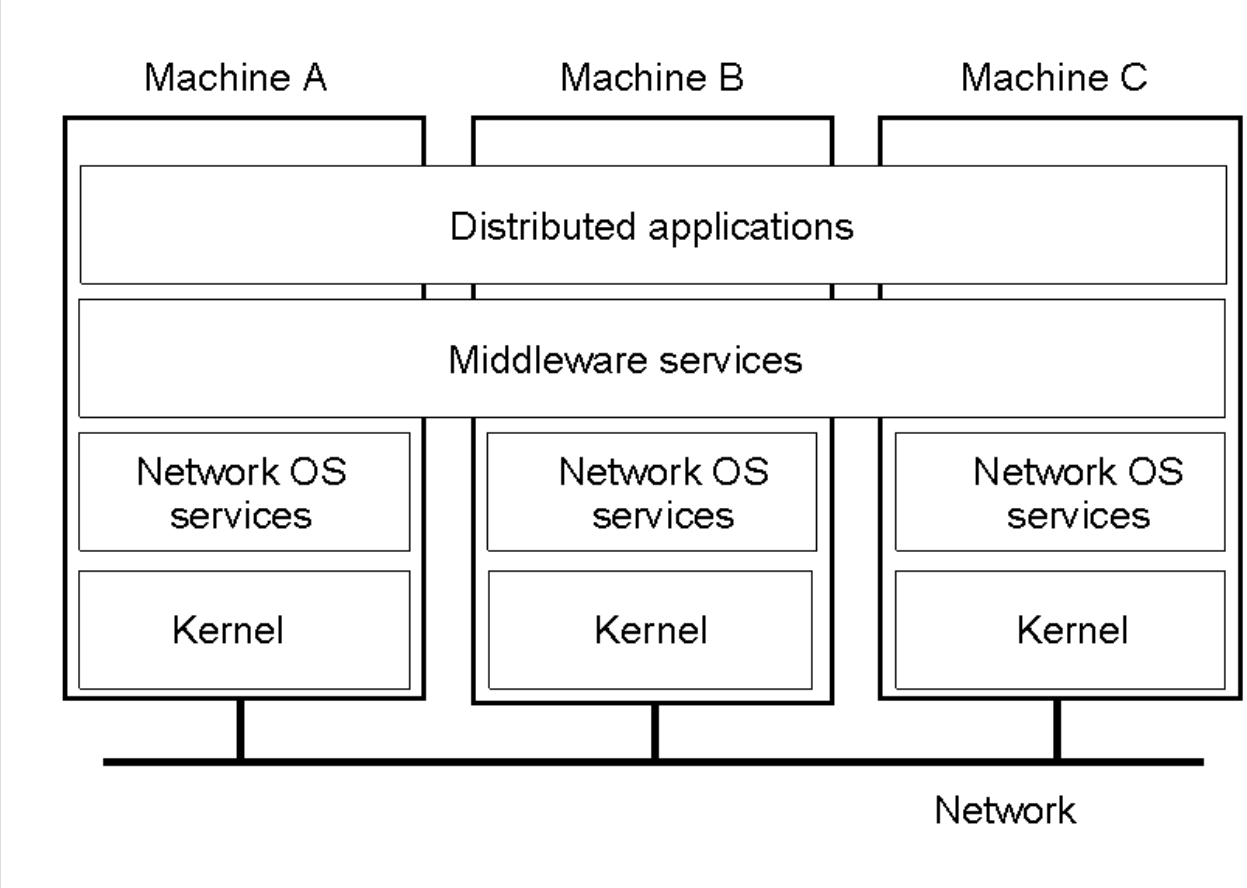
- Uma outra aproximação é minimalista é o uso de micro-kernels.
- Com essa abordagem, apenas uma quantidade mínima de funcionalidade é construída no núcleo dos sistemas operacionais; os serviços são carregados sob demanda.
  - Isso maximiza a memória física disponível pela remoção de funcionalidade indesejada;
  - O usuário pode alterar as características do serviço, por exemplo, um escalonador específico para uma aplicação do cluster pode ser carrregado para ajudar que ela seja executada mais eficientemente.

Sistemas Operacionais Microkernel



- **Separação das aplicações do código do sistema operacional com um microkernel.**

Middleware


Machine A
Machine B
Machine C
Distributed applications
Middleware services
Network OS services
Network OS services
Network OS services
Kernel
Kernel
Kernel
Network

Cluster Middleware

- Reside ente o S.O. e aplicações e oferece infra-estrutura para suportar:
  - Imagem Única do Sistema (SSI)
  - Disponibilidade do Sistema (SA)
- O SSI faz uma coleção de máquinas parecer como um recurso único (visão globalizadas dos recursos do sistema).
- O SA são pontos de verificação e migração de processos.

OpenMosix

OpenMosix

- Pacote de software que transforma computadores em rede rodando Linux em um cluster.
- *Tipo*: Cluster de Alto Desempenho
- Facilidades:
  - Não há necessidade de recompilação ou integração com outras bibliotecas.
  - Um novo nó pode ser adicionado enquanto o cluster está funcionando.
- Cria uma plataforma confiável, rápida e de baixo custo – usada como um supercomputador.

OpenMosix

- Extensão ao núcleo (kernel) do Linux.
- Cluster com Imagem Única do Sistema (SSI)
  - Algoritmo adaptativo de balanceamento de carga.
  - Migração dinâmica de processo para balanceamento de carga.
  - Sistema de Arquivos em Cluster.
  - Totalmente transparente para usuários e aplicações.
- Licença de Pública Geral (GPL)

O que é OpenMosix

- O OpenMosix é um pacote de software que transforma computadores ligados em rede rodando Linux/GNU em um cluster.
- Ele balanceia automaticamente a carga entre diferentes nós do cluster e nós podem entrar ou deixar o cluster sem interrupção do serviço.
- A carga é espalhada entre nós diferentes do cluster de acordo com sua velocidade de processamento e de interconexão.
- Como o OpenMosix é uma parte do núcleo e mantém total compatibilidade com o Linux, um programa de usuário irá funcionar como antes, sem nenhuma modificação.

O que é OpenMosix

- O usuário mais distraído não vai perceber a diferença entre o Linux e o sistema OpenMosix.
- Para ele o todo o cluster irá funcionar com um único (e rápido) sistema Linux.
- O OpenMosix é um remendo para o núcleo do linux que provê total compatibilidade com as plataformas Linux para arquitetura Intel 32 bits.
- O algoritmo interno de balanceamento de carga transparentemente migra os processos para os outros nós do cluster.

O que é OpenMosix

- A vantagem é um melhor balanceamento de carga entre os nós.
- Esta facilidade de migração transparente de processos faz o cluster parecer com um GRANDE sistema SMP com tanto processadores quanto forem os nós disponíveis no cluster.
- O OpenMosix também provê um sistema de arquivos otimizado (oMFS) que, ao contrário do NFS, provê consistência de cache, link e tempo.

Cluster SSI

- Mesma escalabilidade e “overhead” para 2 e para 200 nós.
- Usuários não enxergam os nós individualmente.
- Programas não precisam ser modificados para obter vantagem do cluster (ao contrário do PVM, MPI, etc.)
- Sempre com balanceamento de carga automático.
- Fácil de gerenciar.

Tecnologia OpenMosix

- Migração preemptiva de processos (PPM) transparente
- Processos podem migrar enquanto estão executando:
  - Contexto de Usuário (remoto)
  - Contexto de Sistema (deputado)

Tecnologia OpenMosix

- **Compartilhamento Adaptativo de Recursos (balanceamento de carga)**
  - **Migração rápida, apenas a pilha do processador, registradores e apontador de instruções são efetivamente migrados.**
  - **Paginação sob demanda, apenas as páginas que sofrem falha são enviadas através da rede.**

Tecnologia OpenMosix

- **Memory ushering**, migra processos de um nó que está prestes a ficar sem memória para prevenir o swap das páginas.
- **Parallel File I/O**, traz o processo para o servidor, faz o *direct file I/O* dos processos migrados.
  - Acesso aos Arquivos
    - Direct File System Access (DFSA)
  - Não há nenhuma relação master/slave.

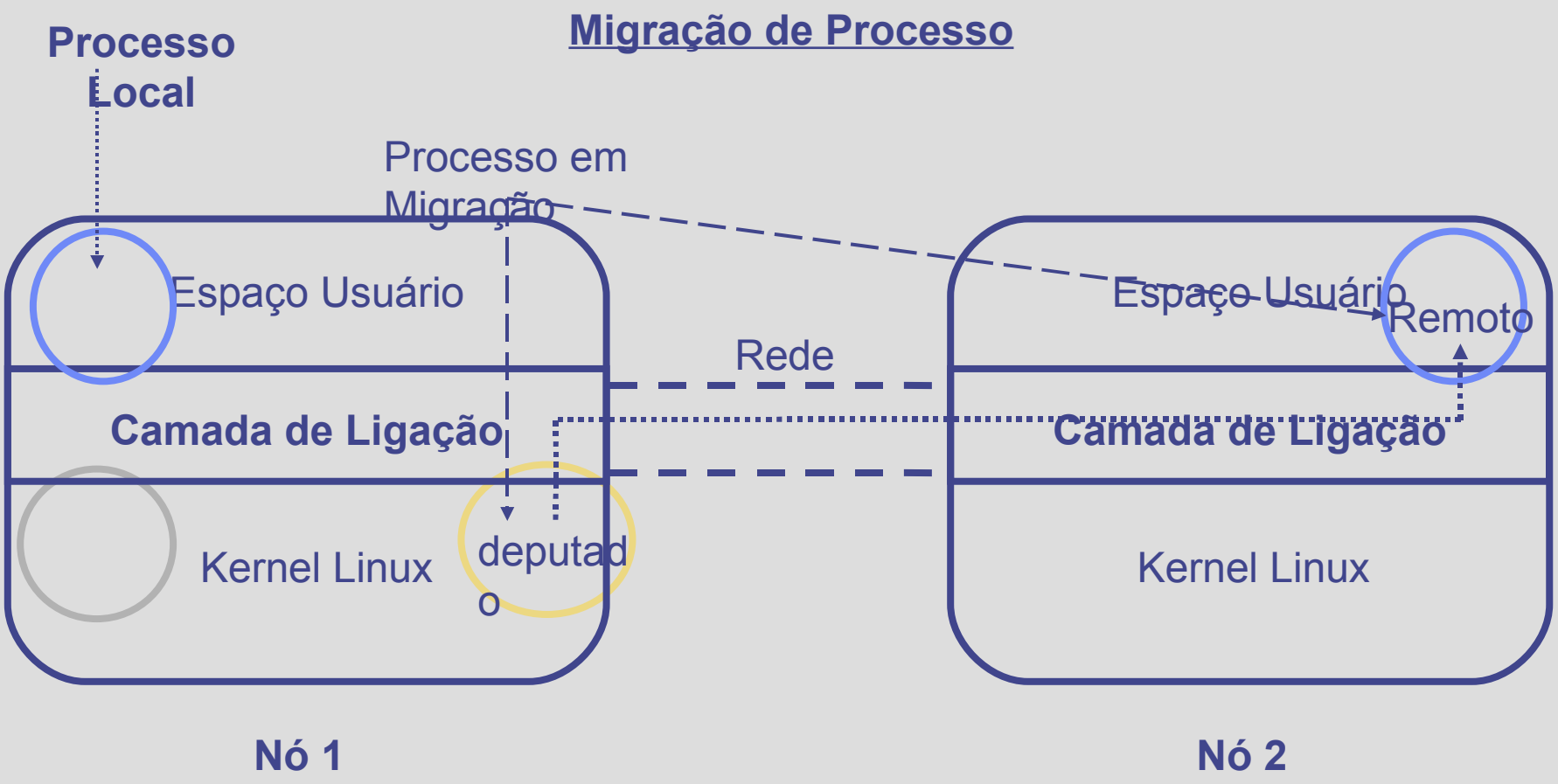
Tecnologia OpenMosix

- Se um processo migra de uma máquina para outra, isto força a todas operações de E/S a serem feitas pela rede.
- É necessário também que as permissões de acesso e os usuários sejam consistentes ao longo de todas as máquinas do cluster.
- O NFS tradicional tem diversos problemas com falta de consistência entre as caches: se dois processos em máquinas distintas estão escrevendo no mesmo arquivo, o arquivo final no disco não será consistente.

Tecnologia OpenMosix

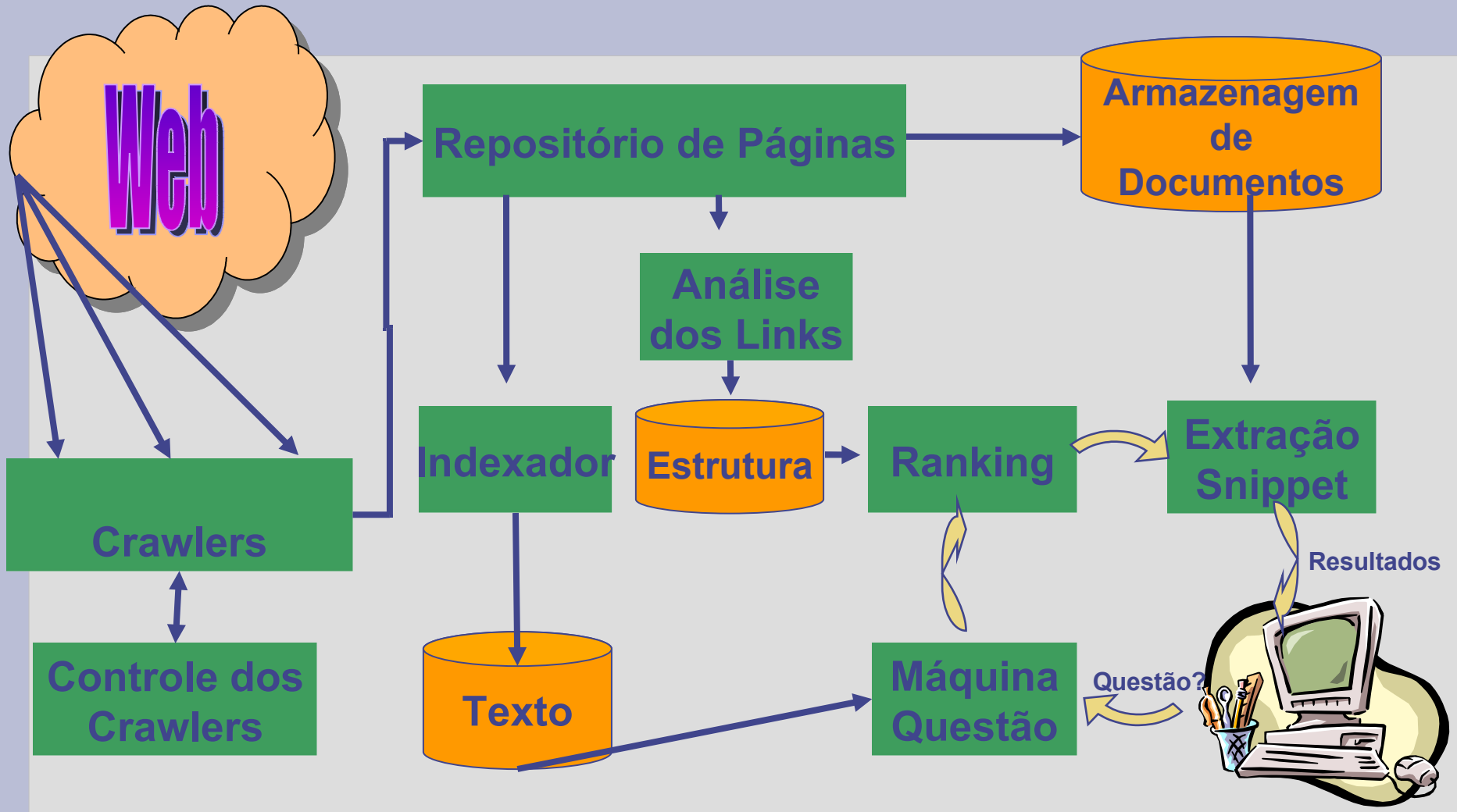
- No OpenMosix existem facilidades asseguram que independente do número de processos remotos escrevendo para um mesmo arquivo ao mesmo tempo, a integridade dos dados e as informação do arquivo são preservadas.
- Existe também uma opção que determina se é mais eficiente fazer uma escrita remota ou migrar a aplicação de volta para o nó que conte, dos dados.
- Em outras palavras, mover o processo para os dados e não contrário.

Como o OpenMosix funciona?

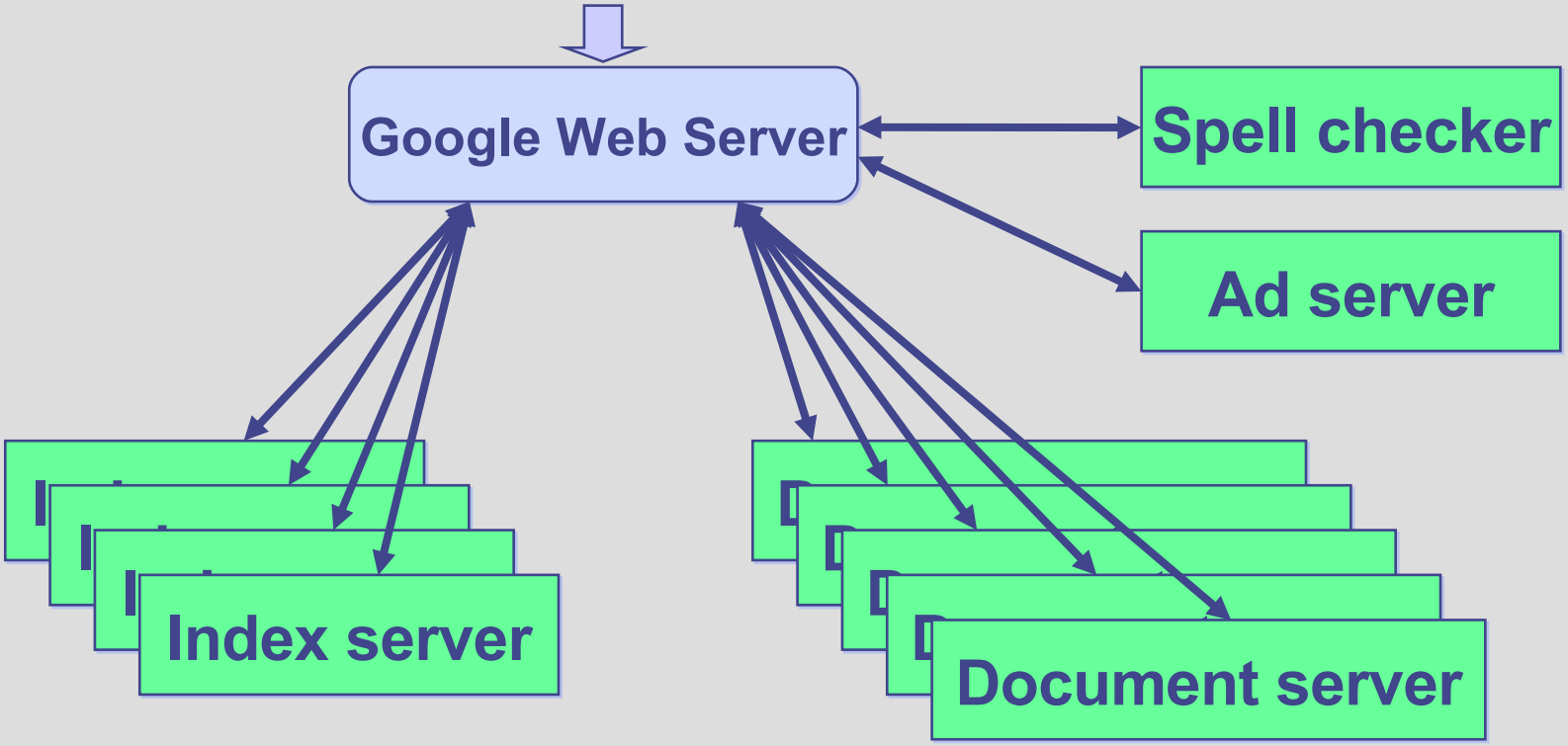


Google™

Arquitetura Máquina de Busca



Arquitetura do Servidor



- **Mais de 15,000 PCs comerciais**
- **Barroso, Dean, Hölzle, "Web Search For A Planet: The Google Cluster Architecture", IEEE Micro, April-March 2003**

Ciclo de Vida de uma Busca

1. O usuário entra com uma busca no formulário enviado pelo Servidor Web do Google.

- O servidor web envia a busca para o cluster de Servidor de Indexação, que correlaciona a pergunta aos documentos.

3. A correlação é enviada para o cluster de Servidor de Documentos, o qual retira os documentos para gerar os resumos e cópias que serão cacheadas.

4. A lista, com os resumos, é mostrada pelo Servidor Web para o usuário, ordenada (usando uma fórmula secreta envolvendo pesos para as páginas).

Projeto

- 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.

Projeto

- 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.

Projeto

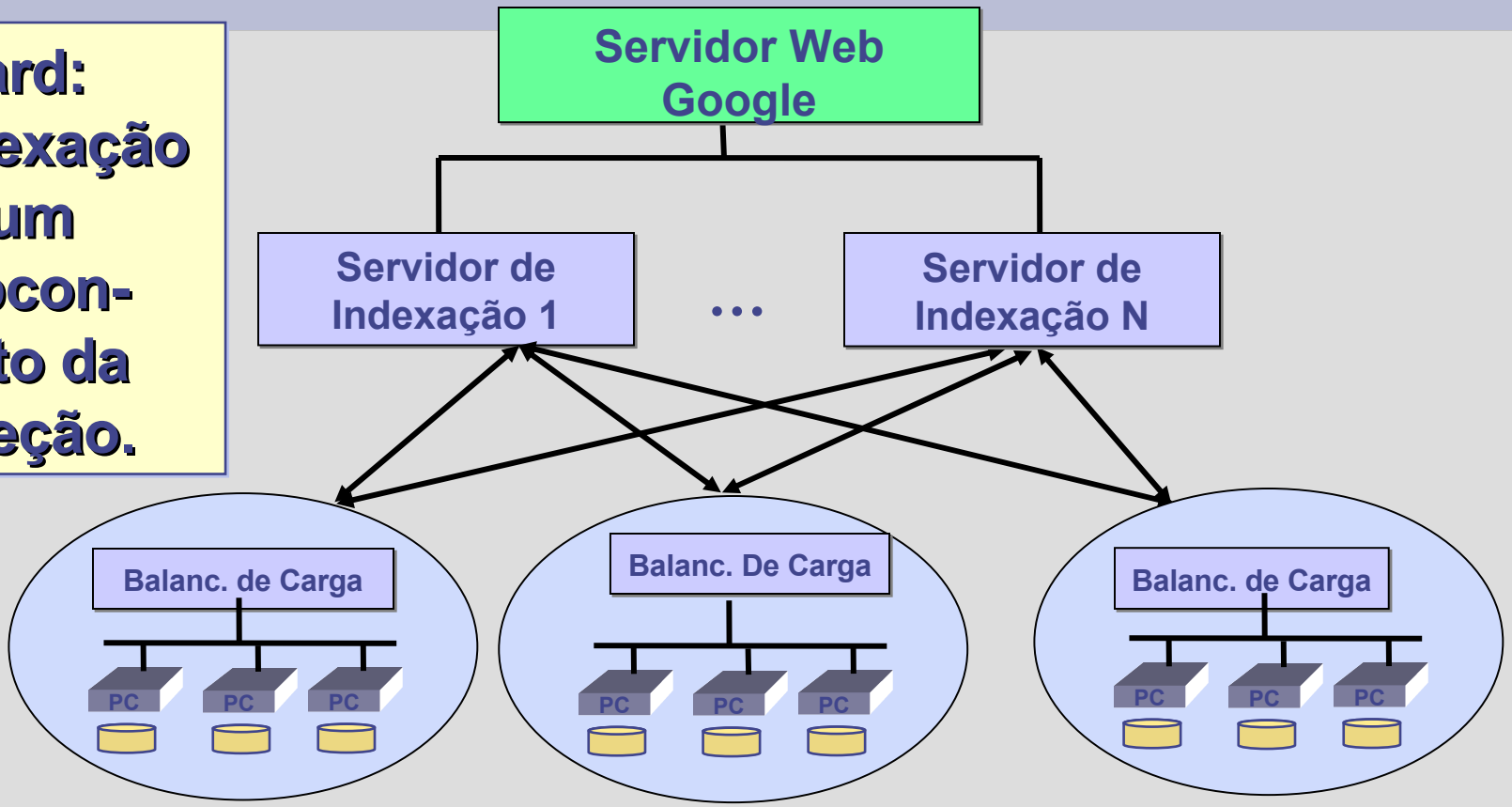
- Uma execução de uma resposta se dá em duas fases:
  - Os servidores de índice consultam uma tabela invertida que mapeia cada palavra da pergunta para uma lista de documentos correspondentes.
  - Os servidores de índice determinam um conjunto de documentos relevantes pela interseção das listas individuais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

Projeto

- 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”.

Servidores de Indexação

Shard: indexação de um subconjunto da coleção.



Cluster servindo shard 1

Cluster servindo shard 2

Cluster servindo shard *n*

Projeto

- 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.

Projeto

- 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.

Projeto

- Em realidade existem diversas cópias do conteúdo da Web nos servidores Google por uma questão de desempenho e disponibilidade.
- Em todo o processo o máximo de paralelismo é explorado pela subdivisão das tarefas através de diversos servidores do cluster.
- No final do processo, o servidor GWS monta a página HTML que é visualizada pelo usuário.

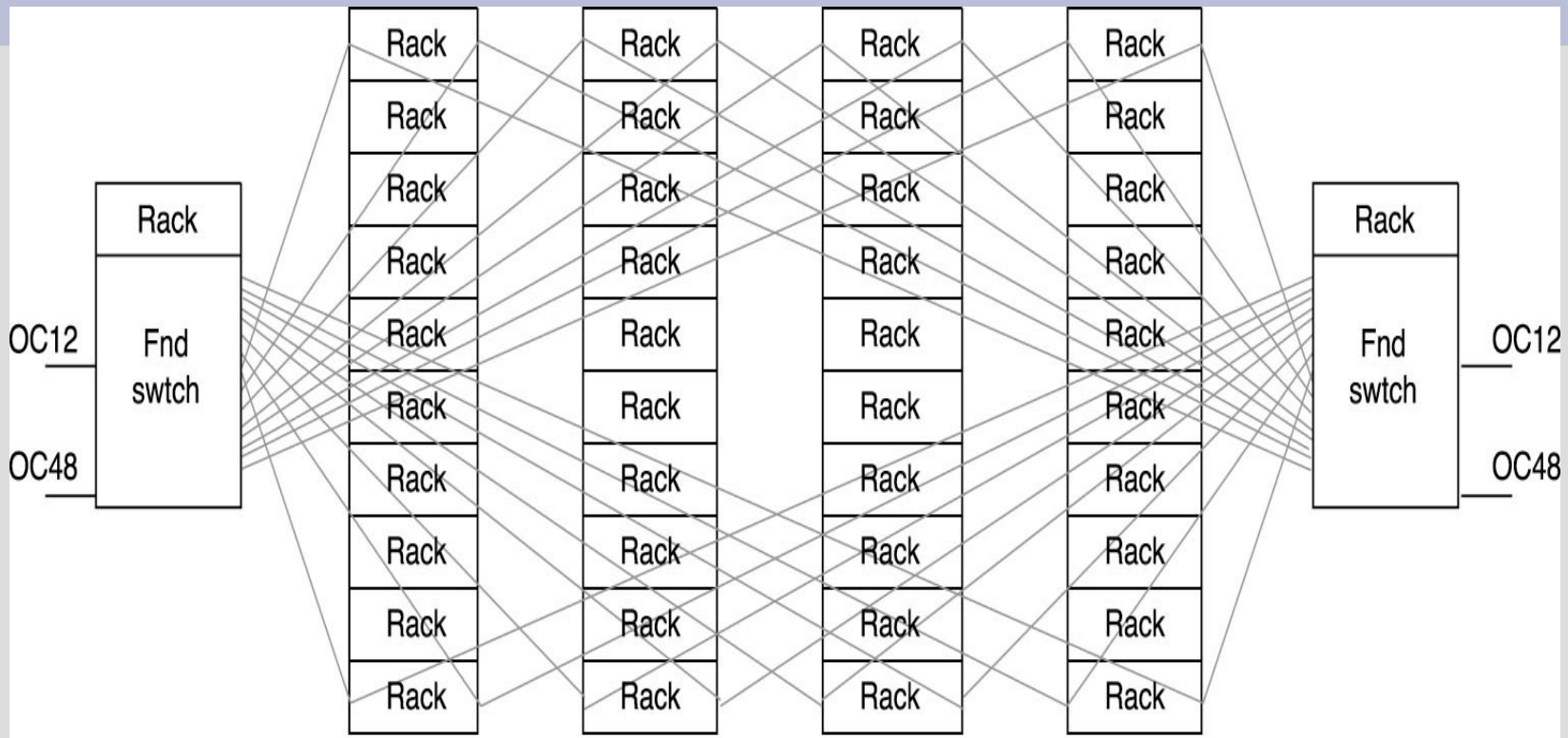
Princípos de Projeto

- 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.

Configuração dos Racks

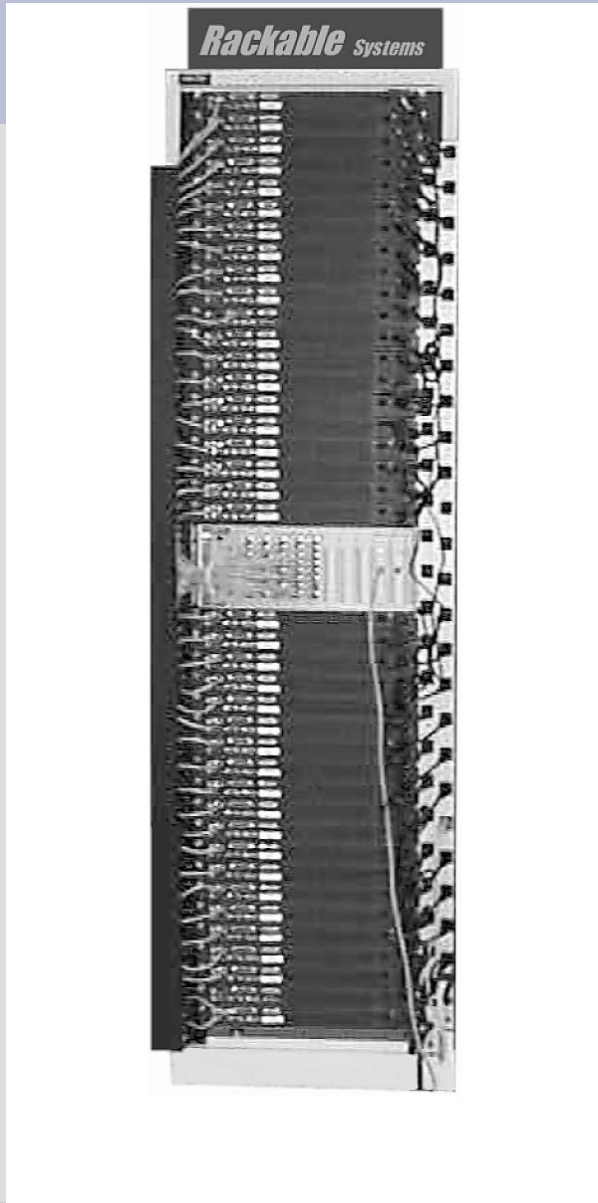
- Cada rack consiste de 40 a 80 servidores X86 montados em ambos os lados de um rack personalizado.
- Em dez/2002 havia diversas gerações de processadores em serviço, desde Celerons de 500 Mhz até servidores duais com Pentium III de 1.4 Ghz.
- Cada servidor contém um ou mais discos IDE de 80 Gb e 2 GB 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.

Arquitetura

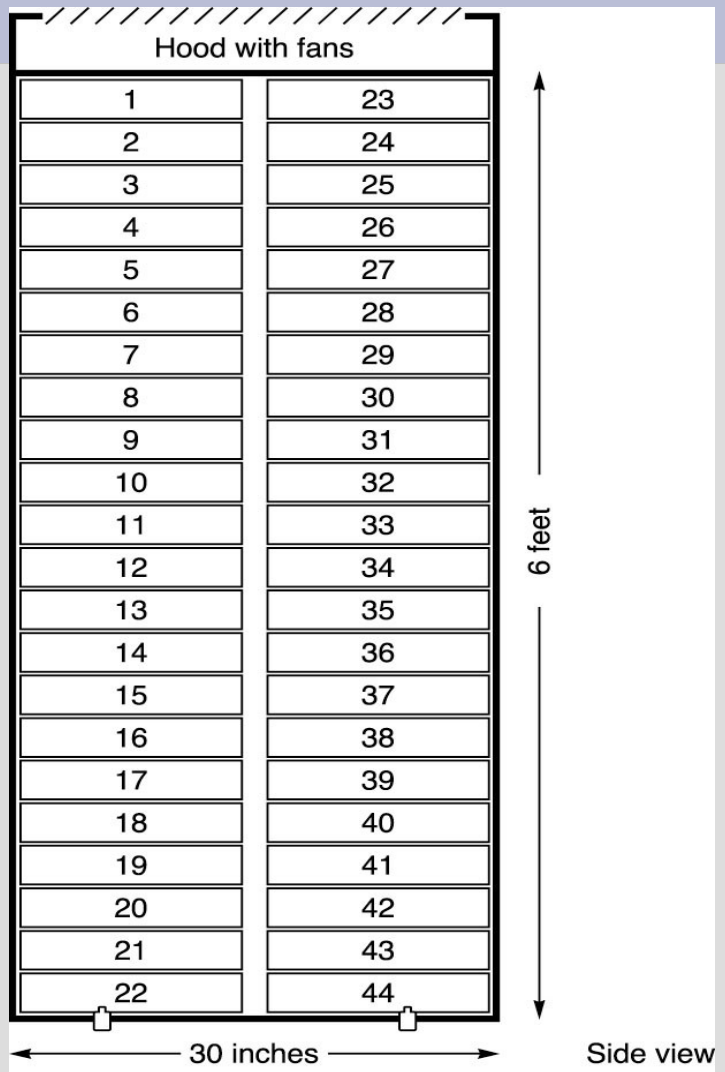


© 2003 Elsevier Science (USA). All rights reserved.

Arquitetura



Arquitetura



Configuração dos Racks

- O critério final para a seleção é o custo por pedido, expressado pela soma de capital dispendido mais os custos de operação dividido pelo desempenho.
- O custo do equipamento deve ser amortizado em dois ou três, pois ao final deste período ele já estará obsoleto.
- Por exemplo, o custo total de um rack era de U$ 280.000,00 em dez/2002. Isto de traduz em custo mensal de capital de U$ 7.700 por rack ao longo de três anos.
- Por conta disto, o uso de placas mães com 4 processadores foi descartado, assim com discos SCSI, pois este custo se elevaria demasiadamente.

Consumo dos Racks

- **Um rack com 80 servidores consome cerca de 10 KW, ou 120 W por servidor.**
- **Considerando que um rack ocupa 2,3 m^2, resulta em uma densidade de potência de 4,3 KW/m^2. Com o uso de servidores de alto desempenho este valor pode subir para cerca de 7,6 KW/m^2.**
- **Mas o custo de energia é relativamente barato, cerca de U$ 1500,00/mês, bem menor do que o custo de depreciação de U$ 7.700,00 /mês.**

Dados sobre o Google

- 3 bilhões de páginas da Web
- 22 milhões de arquivos PDF
- 700 milhões de mensagens de grupos
- 425 milhões de imagens indexadas
- Serve + de 150 milhões pesquisas/dia.
- http://labs.google.com/os

Curiosidades

- No ano de 2000 o Google serviu 1000 pedidos por segundo.
- O Google busca a web inteira uma vez por mês.
- Em dezembro de 2000 (quatro anos atrás) Google usava 6000 processadores e 12000 discos totalizando 1 Petabyte de dados, distribuídos em 3 centros de serviços nos EUA.
- As buscas têm crescido a uma taxa de 90% a cada ano no Google.
- Estima-se hoje que o Google tenha cerca de 100.000 servidores distribuídos por uma dúzia de centros em todo o mundo.

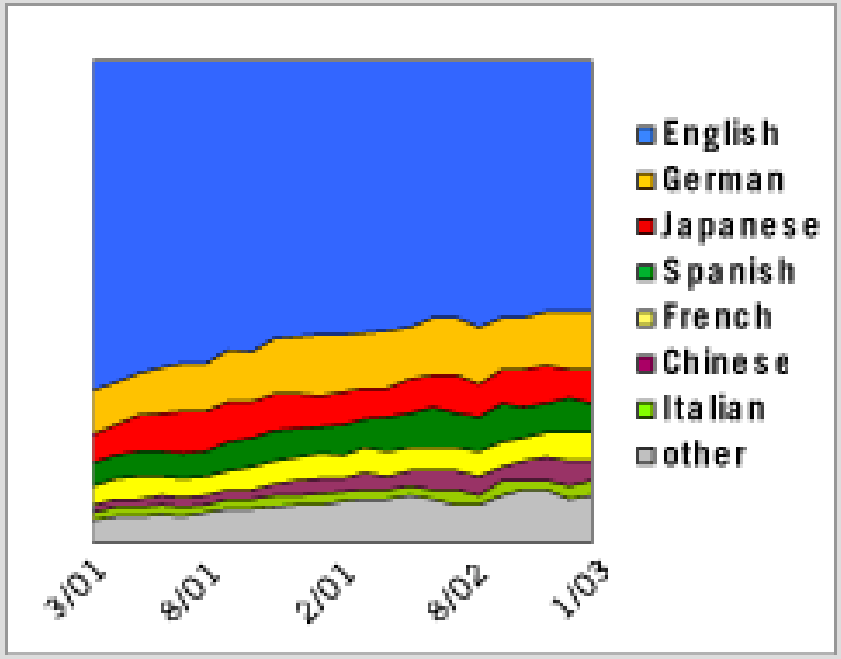
Curiosidades

- Software é o elo fraco – a maior parte das fontes de falha são de software.
- Cerca de 20 máquinas devem ser reiniciadas por dia.
- Cerca de 80 máquinas quebram por dia.
- A reiniciação deve ser feita manualmente
- 2-3% dos PCs devem ser substituídos todo ano.
- Discos e Memória respondem por 95% das falhas.

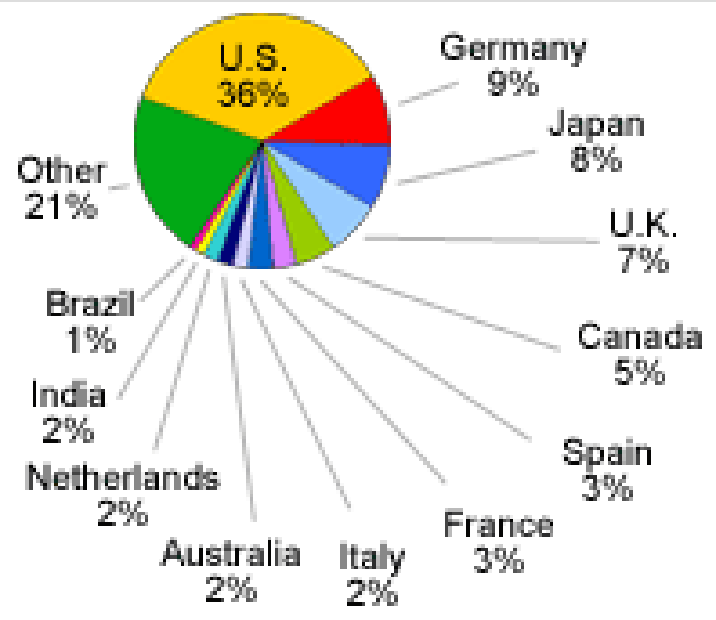
Curiosidades

- http://www.google.com/press/zeitgeist.html

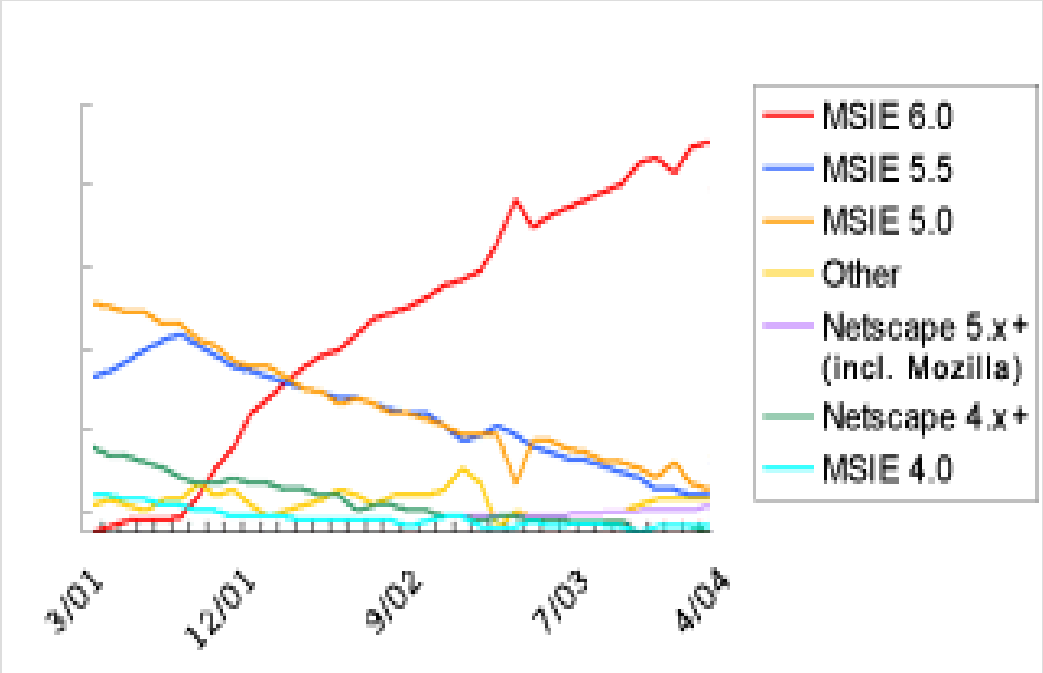
Línguas utilizadas no Google
(Março 2001 – Janeiro 2003)

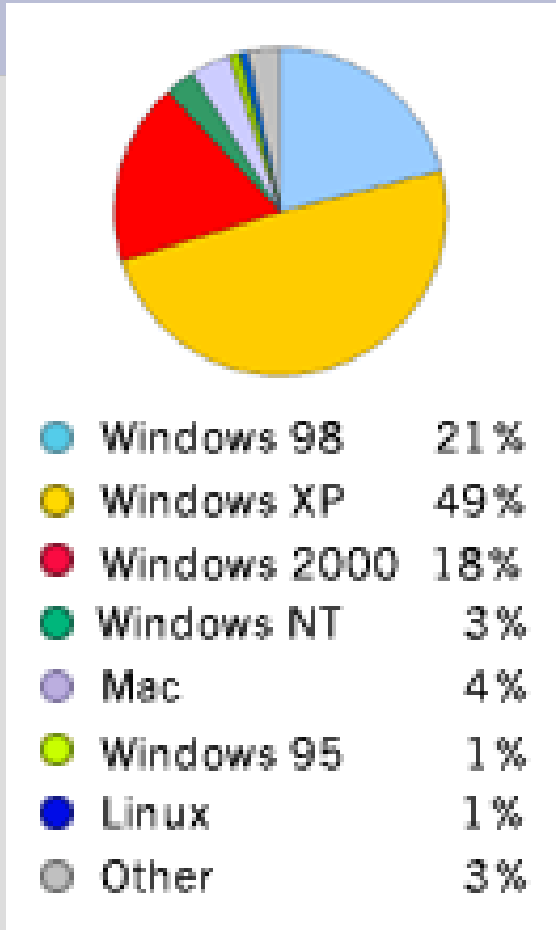


Países de Origem (Outubro 2001)



Curiosidades
(abril/2004)





Curiosidades
(abril/2005)

- receita federal
- hello kitty
- amor
- beijo
- avril lavigne
- indios
- slipknot
- britney spears
- xuxa
- ragnarok

Page Rank

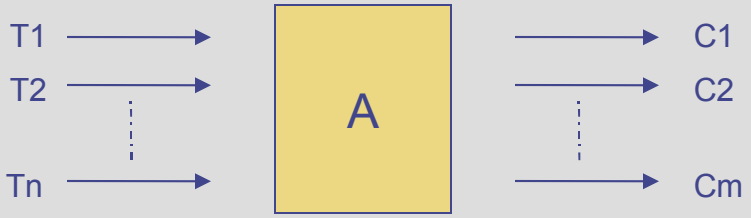
- O “PageRank” é baseado na natureza democrática da Web, utilizando a sua vasta estrutura de links como um indicador do valor individual de uma página.
- Em essência, o google interpreta um link de uma página A para uma página B como um voto da página A para a página B.
- O google faz mais do que apenas olhar o volume de votos, ou links, que uma página recebe.
- Votos dados por páginas que são consideradas “importantes” valem mais e ajudam a fazer outras páginas a serem “importantes” também.

Page Rank

- Páginas de alta qualidade, consideradas "importantes", recebem um PageRank maior, que o google se lembra cada vez que faz uma busca.
- Claro, páginas importantes não significam nada para você se elas não estão de acordo com seu critério de busca.
- Então o Google combina o PageRank com técnicas sofisticadas de "text-matching" para encontrar páginas que são tão importantes quanto relevantes para sua busca.

Page Rank

- PageRank – Trazendo ordem à WEB



- **PR(A) = (1-d) + d (PR(T1)/C(T1) + ….. + PR (Tn)/C(Tn))**
  - ***PR(T1) is the PageRank of the page that links to our (A) page.***
  - ***C(T1) is the number of links going out of page T1.***
  - ***d is a damping factor, usually set to 0.85.***
  - ***The sum of all web pages' PageRanks will be one.***

Curiosidades

- O Google vai além do número de vezes que um termo aparece em uma página e examina todos os aspectos do conteúdo da página (e o conteúdo das páginas que apontam para ela) para determinar se é uma boa escolha para a sua busca.

- O seu grande desempenho é explicado por um conjunto engenhoso de “hardware” e “software” trabalhando harmoniosamente para fornecer um serviço de qualidade para os seus usuários.

Conclusões

- Cluster hoje são uma realidade.
- Oferecem crescimento incremental e cabem no orçamento.
- Novas tendências tecnológicas em hardware e software permitirão aos “clusters” parecer cada vez mais com um único sistema.
- Supercomputadores baseados em “clusters” poderão ser uma solução computacional para países como o Brasil.