Universidade Federal do Rio de Janeiro
Informática DCC/IM

Arquitetura de Computadores II

Clusters

Gabriel P. Silva

Clusters

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

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.

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.

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.

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.

Tipos de Clusters

- Assim que uma máquina falhar, as outras máquinas tentam assumir o trabalho.
- 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.

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.

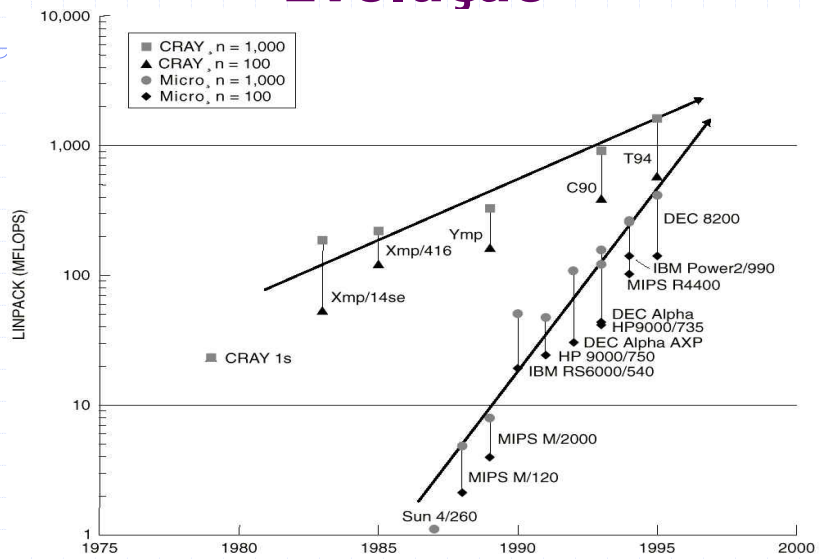
Clusters

- 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



Exemplo

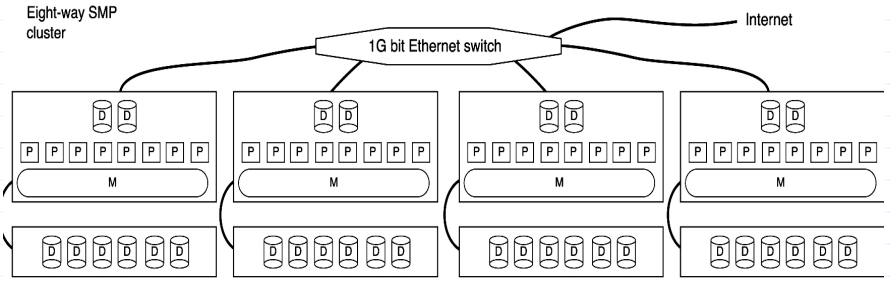
Hardware

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

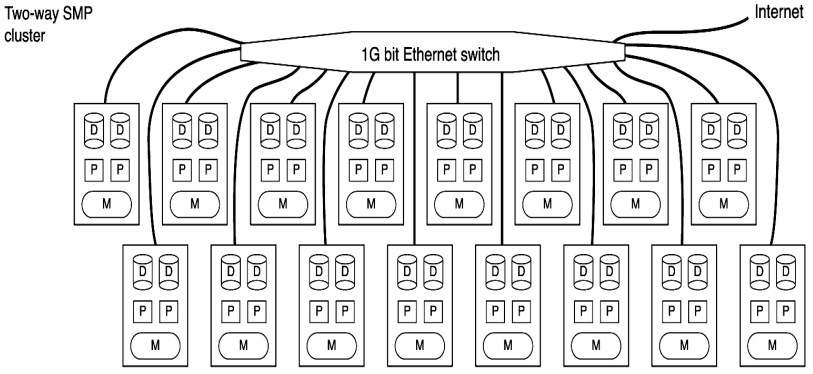
Hardware

- **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

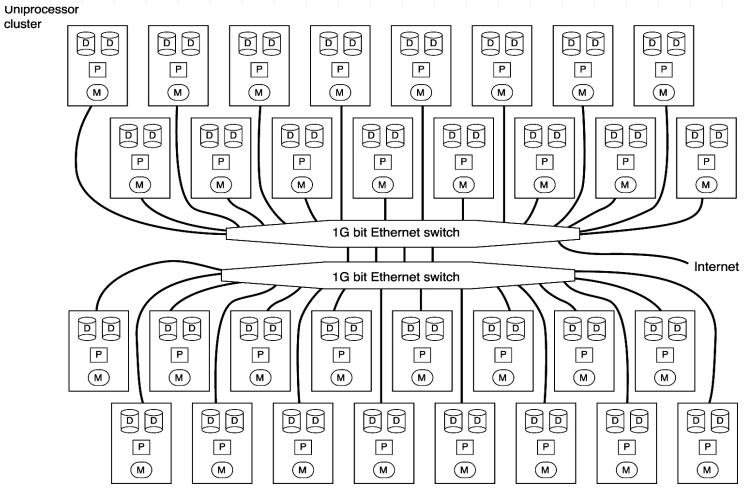
Cluster – SMP c/ 8 processadores



Cluster – SMP c/ 2 processadores



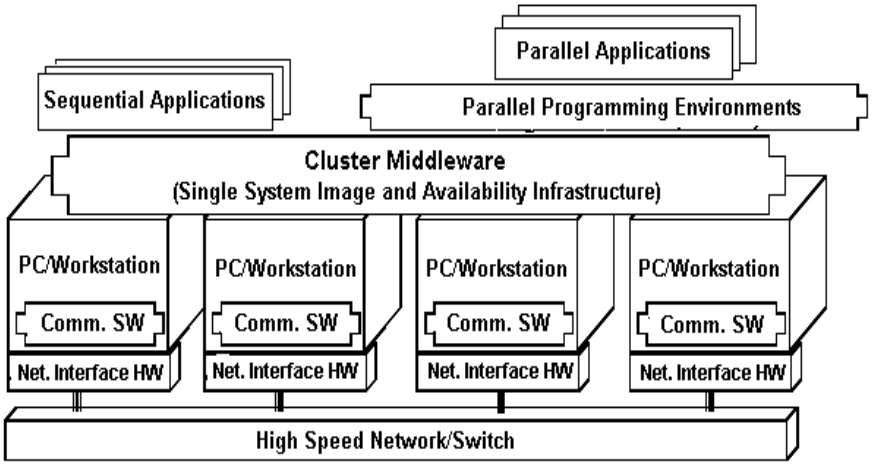
Cluster baseado em Monoprocessadores



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)

Arquitetura de um Cluster



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)**

Maiores Desafios

- **Balanceamento de Carga (CPU, Rede, Memória, Discos)**
- **Segurança e Encriptação (clusters de clusters)**
- **Gerenciabilidade (admin. e controle)**
- **Programabilidade (API simples)**
- **Aplicabilidade (aplicações voltadas para o cluster)**

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.

Cluster Middleware

- Sistemas Operacionais:
  - Solaris MC
  - Unixware
  - MOSIX
  - Rocks
- Sistemas em Execução
  - Sistemas em Execução (software DSM, PFS, etc.)
  - Gerenciamento de Recursos e Despacho (RMS):
    - CODINE, CONDOR, LSF, PBS, NQS, etc.

Ambientes de Programação

- Threads (PCs, SMPs, NOW..)
  - POSIX Threads
  - Java Threads
- MPI
  - http://www-unix.mcs.anl.gov/mpi/
- PVM
  - http://www.epm.ornl.gov/pvm
- Software DSMs (SHMEM da Cray/SGI)

Modelo de Programação SHMEM™

- Usando a biblioteca SHMEM um processador pode ler e escrever na memória de outro processador sem que necessite da cooperação deste outro processador.
- Este método se chama mensagem ativa.
- O uso de mensagem ativa contrasta com a troca de mensagem via MPI ou PVM, na qual uma mensagem é enviada em processo com dois passos: o processador origem faz uma chamada para enviar uma mensagem e o processador remoto faz uma chamada para recebe-la.

Modelo de Programação SHMEM™

- A cooperação é necessária entre os dois processadores porque o processador destino deve fazer uma chamada para uma biblioteca para aceitar os dados antes de utiliza-lo.
- Com o uso de mensagem ativa com SHMEM, o envio de dados envolve apenas uma CPU. O processador origem simplesmente coloca os dados na memória do processador destino.
- Do mesmo modo, um processador pode ler o conteúdo da memória de um outro processador sem interromper o processador remoto.

Modelo de Programação SHMEM™

- O processador remoto só toma conhecimento que sua memória foi modificada ou escrita se o programador implementar um mecanismo para isto.
- A biblioteca SHMEM inclui uma quantidade de rotinas altamente otimizadas para operações coletivas como reduções.
- Já que ela pode ser implementada com eficiência em sistemas com memória compartilhada endereçável globalmente, o uso desta biblioteca melhora a latência de comunicação uma ordem de grandeza acima das implementações com MPI.

Ferramentas

- Compiladores
  - C/C++/Java/
- Depuradores
- Ferramentas de Análise de Desempenho
- Ferramentas de Visualização

Aplicações

- Podem ser seqüenciais, quando se beneficiam do balanceamento de carga.
- Podem ser paralelas /distribuídas, quando se utilizam dos ambientes de programação existentes.
- Data-mining
- Servidores Web

Aplicações

- Grandes Desafios:
  - Previsão de tempo
  - Química Quântica
  - Modelagem de Biologia Molecular
  - Análise de Engenharia (CAD/CAM)
  - Modelagem de Oceanos

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.

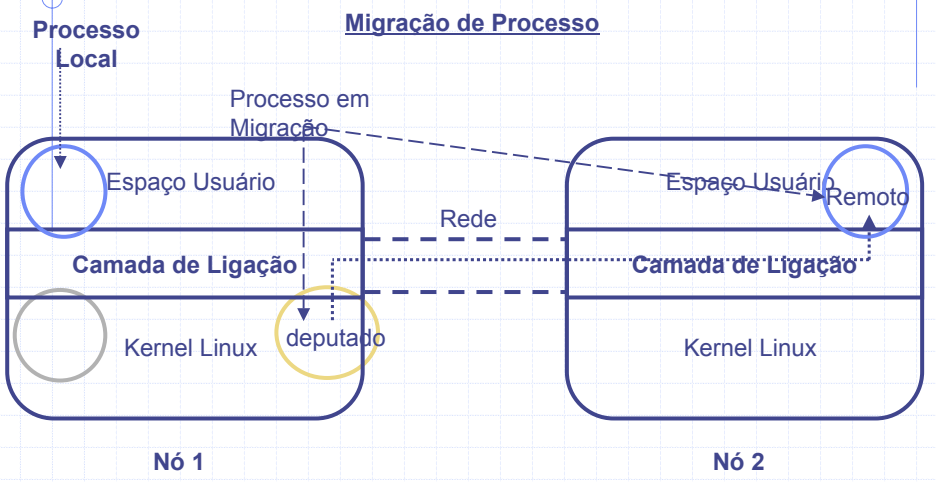
Tecnologia OpenMosix

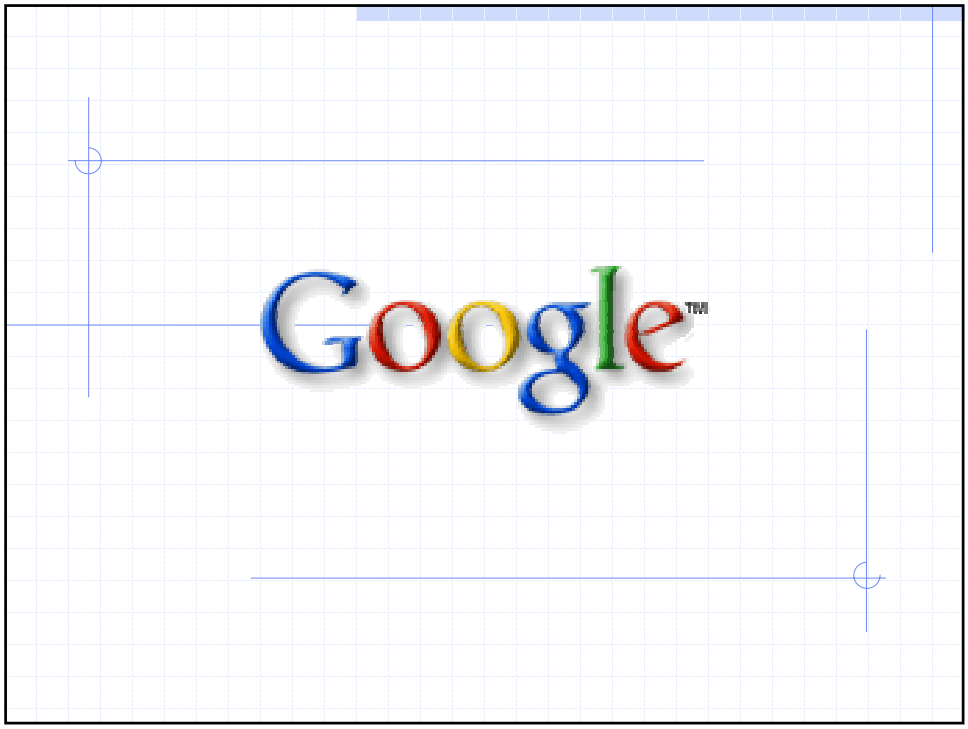
- Se um processo migra de uma máquina para outra, isto força a todas operações de E/S a serem feitas pela rede.
- É necessário também que as permissões de acesso e os usuários sejam consistentes ao longo de todas as máquinas do cluster.
- O NFS tradicional tem diversos problemas com falta de consistência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

Tecnologia OpenMosix

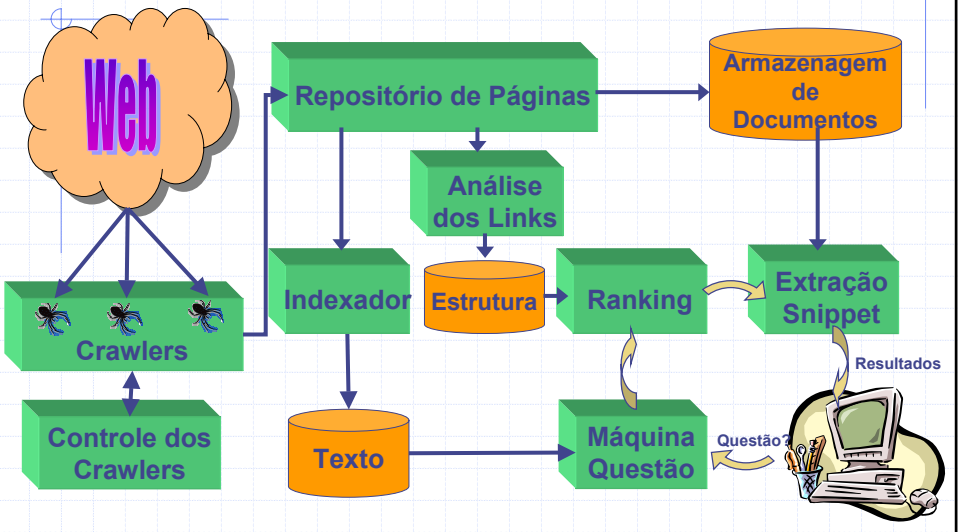
- No OpenMosix existem facilidades asseguram que independente do número de processos remotos escrevendo para um mesmo arquivo ao mesmo tempo, a integridade dos dados e as informação do arquivo são preservadas.
- Existe também uma opção que determina se é mais eficiente fazer uma escrita remota ou migrar 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?

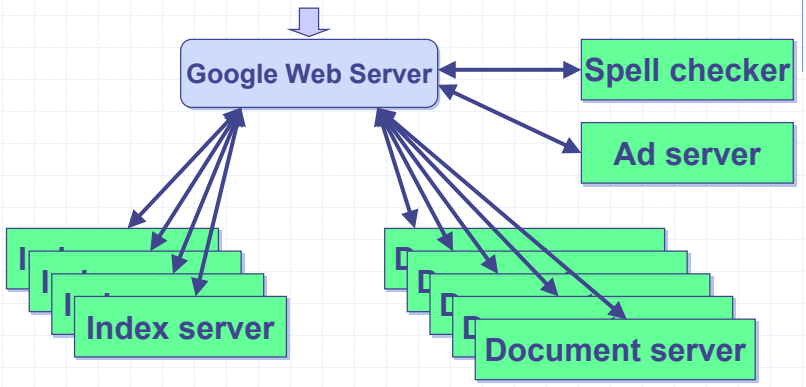




Arquitetura Máquina de Busca

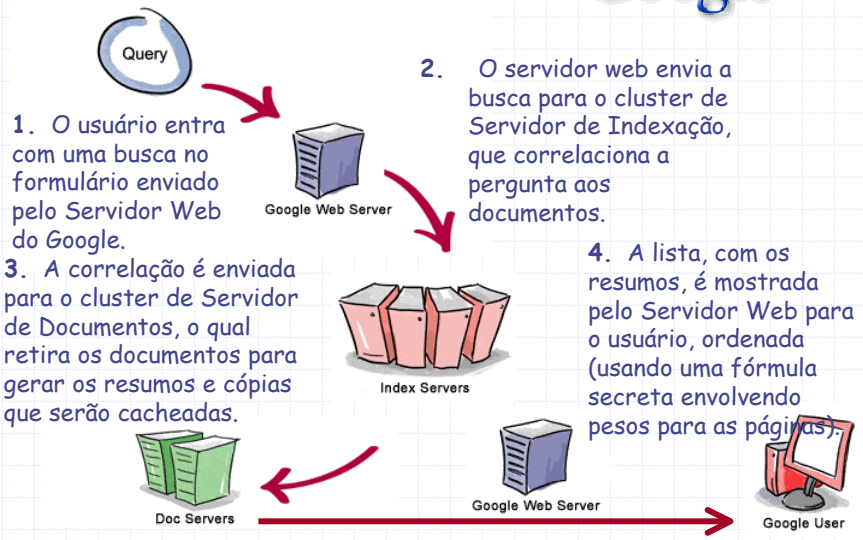


Arquitetura Servidor Google



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

Vida de uma Busca no Google



Projeto 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.**

Projeto 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.

Projeto Google

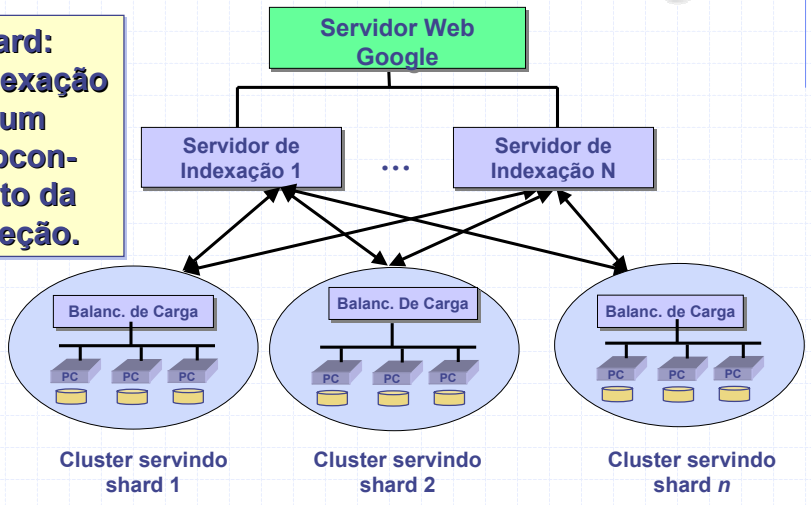
- Uma execução de uma resposta se dá em duas fases:
  - Os servidores de índice consultam uma tabela invertida que mapeia cada palavra da pergunta para uma lista de documentos correspondentes.
  - Os servidores de índice determinam um conjunto de documentos relevantes pela interseção das listas individuais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

Projeto 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".

Servidores de Indexação Google

Shard: indexação de um subcon-junto da coleção.



Projeto 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.

Projeto 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.

Projeto Google

- 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.

Page Rank

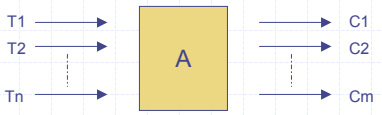
- O "PageRank" é baseado na natureza democrática da Web, utilizando a sua vasta estrutura de links como um indicador do valor individual de uma página.
- Em essência, o google interpreta um link de uma página A para uma página B como um voto da página A para a página B.
- O google faz mais do que apenas 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 eleas não estaõ 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 – Bring order to the 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.***

Page Rank

- **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.**

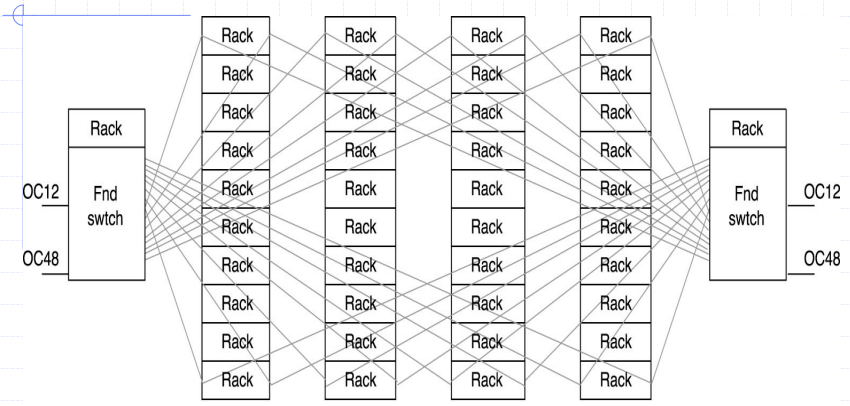
Princípos de Projeto Google

- **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 Google

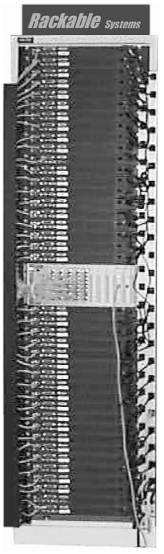
- Cada rack consiste de 40 a 80 servidores X86 montados em ambos os lados de um rack personalizado.
- Em dez/2002 havia diversas gerações de processadores em serviço, desde Celerons de 500 Mhz até servidores duais com Pentium III de 1.4 Ghz.
- Cada servidor contém um ou mais discos IDE de 80 Gb e 2 GB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

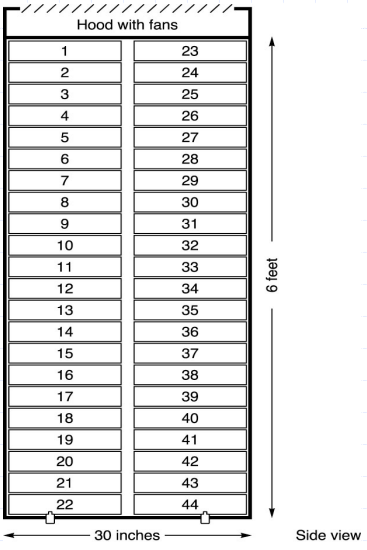


© 2003 Elsevier Science (USA). All rights reserved.

Arquitetura Google



Arquitetura Google



Configuração dos Racks Google

- 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 Google

- 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 Google

- **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.**

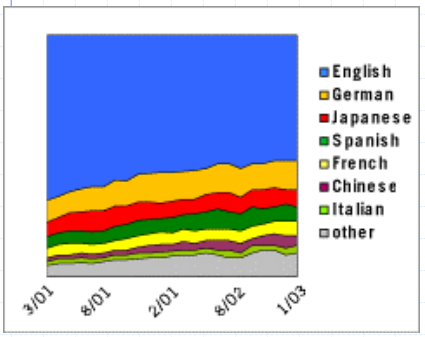
Curiosidades Google

- 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.

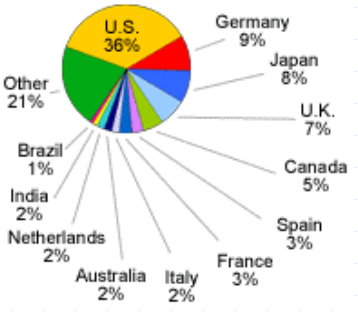
Curiosidades Google

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

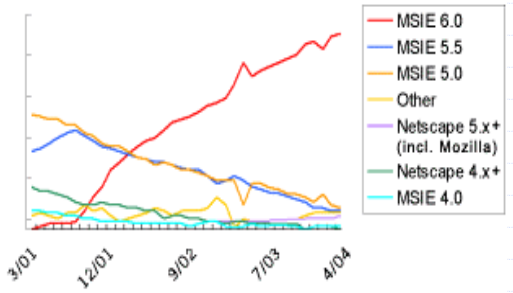
Línguas utilizadas no Google
(Março 2001 – Janeiro 2003)

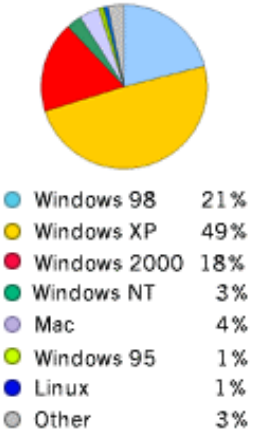


Países de Origem (Outubro 2001)



Curiosidades Google (abril/2004)





Curiosidades Google (abril/2005)

1. receita federal
2. hello kitty
3. amor
4. beijo
5. avril lavigne
6. indios
7. slipknot
8. britney spears
9. xuxa
10. ragnarok

Bibliografia Google

• Interpreting the Data: Parallel Analysis with Sawzall (Draft) [PDF]
Rob Pike, Sean Dorward, Robert Griesemer, and Sean Quinlan

• MapReduce: Simplified Data Processing on Large Clusters [PDF]
Jeffrey Dean and Sanjay Ghemawat

• The Google File System [PDF]
Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung

• Web Search for a Planet: The Google Cluster Architecture [PDF]
Luiz Barroso, Jeffrey Dean, and Urs Hoelzle

Bibliografia Google

•Query-Free News Search [PostScript]
Monika Henzinger, Bay-Wei Chang, Brian Milch, and Sergey Brin

•Extracting knowledge from the World Wide Web [PDF]
Monika Henzinger and Steve Lawrence

•Searching the Web by Voice [PDF]
Alex Franz and Brian Milch

•Who Links to Whom: Mining Linkage between Web Sites [PDF]
Krishna Bharat, Bay-Wei Chang, Monika Henzinger, and Matthias Ruhl

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.**