



Disciplina: Sistemas Distribuídos

---

Sistemas de Informação  
8º Período

**Prof. Cristiano Vieira**  
**2020**



# Sistemas Distribuídos

---

## **Introdução aos Sistemas Distribuídos**

AULA 1

# Definição

---

- **Sloman, 1987**

“Um sistema de processamento distribuído é tal que, vários processadores e dispositivos de armazenamento de dados, comportando processos e/ou bases de dados, interagem cooperativamente para alcançar um objetivo comum. Os processos coordenam suas atividades e trocam informações por passagem de mensagens através de uma rede de comunicação”

# Definição

---

- **Andrew Tanenbaum**

“Coleção de computadores independentes que se apresenta ao usuário como um sistema único e consistente (coerente)”

- **Coulouris**

“Coleção de computadores autônomos interligados através de uma rede de computadores e equipados com software que permita o compartilhamento dos recursos do sistema: hardware, software e dados”

# Principais implicações das definições

- ❖ Concorrência: Execução de programas (**processos**) de forma concorrente com possível acesso a recursos compartilhados.
- ❖ Ausência de relógio global: Os processos cooperam por meio de troca de mensagens e assim necessitam de uma noção compartilhada de tempo, embora não seja possível utilizar um relógio único. A depender do ambiente, relógios podem ser sincronizados entre si, mas há limites de precisão (**distorção entre relógios**) obtida pela sincronização.
- ❖ Falhas independentes: Qualquer componente do sistema pode falhar. As falhas podem ocorrer de forma independente (**sem correlação entre si**), se a infraestrutura subjacente (e.g. nós processadores, sistema de alimentação, rede etc.) for independente. Falhas não são imediatamente percebidas pelos demais componentes.

# Motivação

---

- Crescente dependência por parte dos usuários
- Demanda maior que avanços combinados de *hardware* e *software* centralizados
- Características inexistentes em sistemas centralizados como tolerância a falhas (*fault tolerance*)

# O que é um Sistema Distribuído?

---

Um sistema distribuído é uma coleção de ***hosts*** **autônomos**, conectados através de uma **rede** de computadores. Cada *host* executa componentes e opera um **middleware** de distribuição, o qual habilita os componentes a **coordenarem** suas atividades de tal forma que usuários percebam o sistema como um **ambiente computacional único e integrado**.

# Características de Sistemas Centralizados

---

- Um componente, com partes não autônomas
- Componentes são compartilhados por todos os usuários durante todo o tempo
- Todos os recursos acessíveis (tipicamente)
- Software 'roda' em um único processo
- Ponto de controle único
- Ponto de falha único

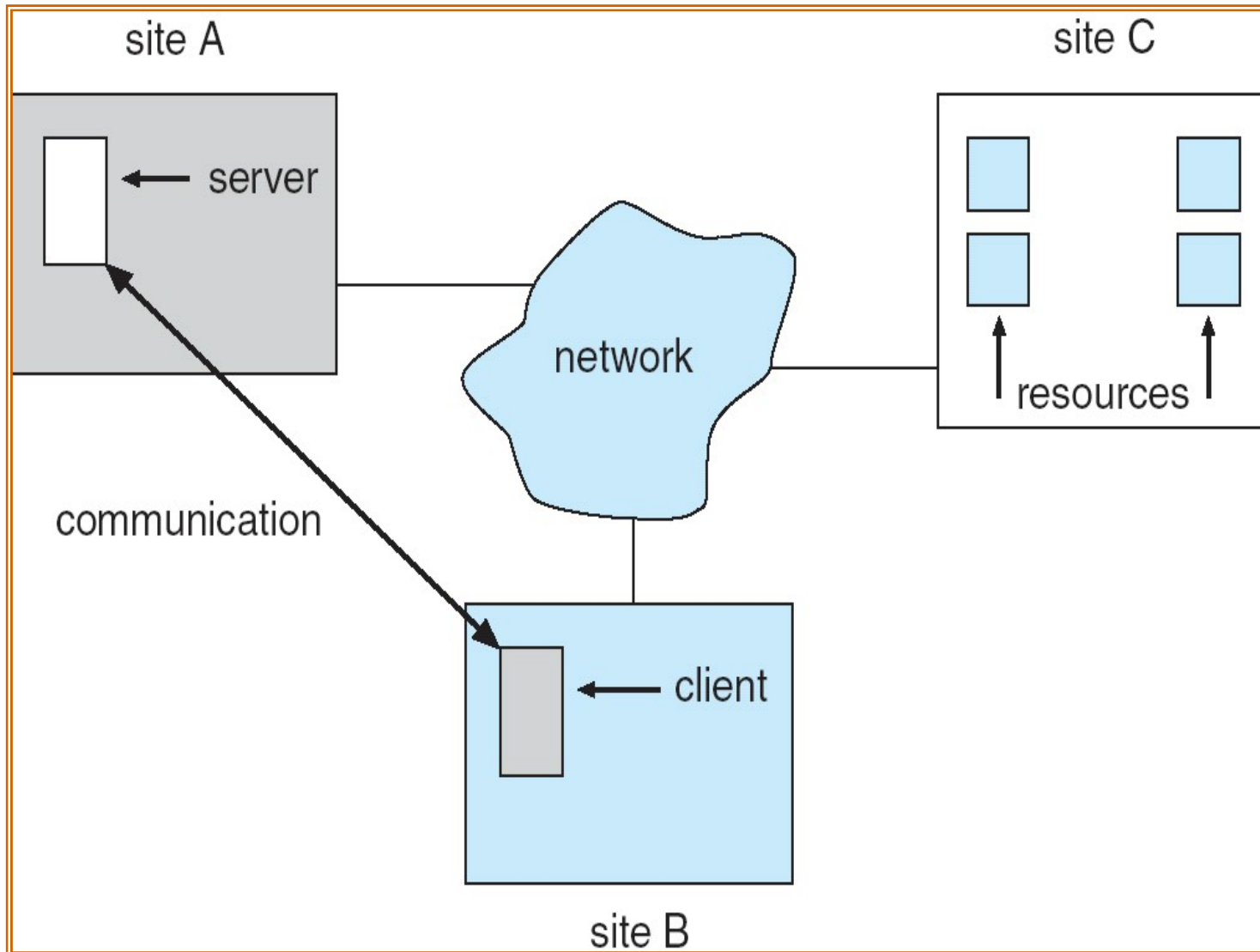


# Características de Sistemas Distribuídos

---

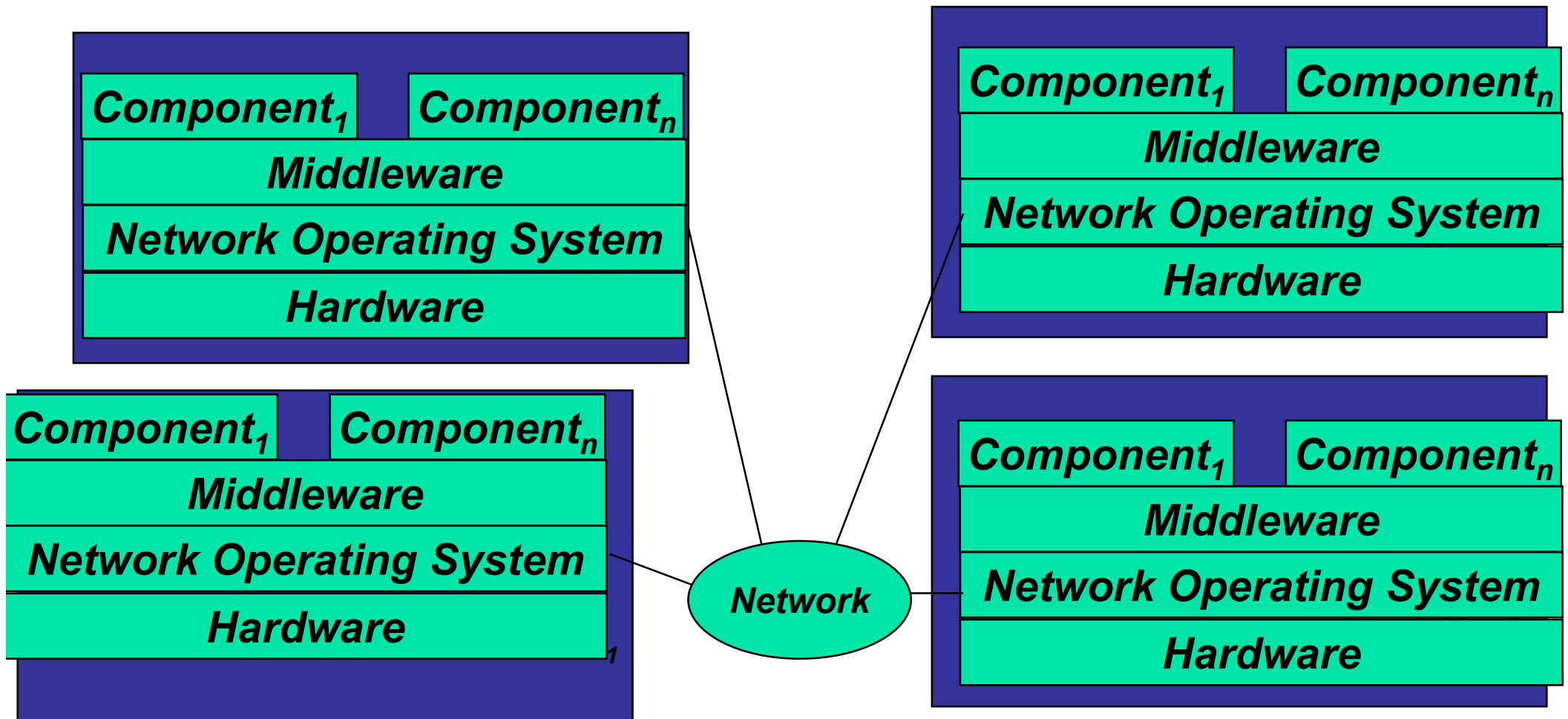
- Múltiplos componentes autônomos
- Componentes não são compartilhados por todos os usuários
- Recursos podem não ser acessíveis
- Software 'roda' em processos concorrentes e em processadores distintos
- Múltiplos pontos de controle
- Múltiplos pontos de falha (!!!)

# Um sistema distribuído

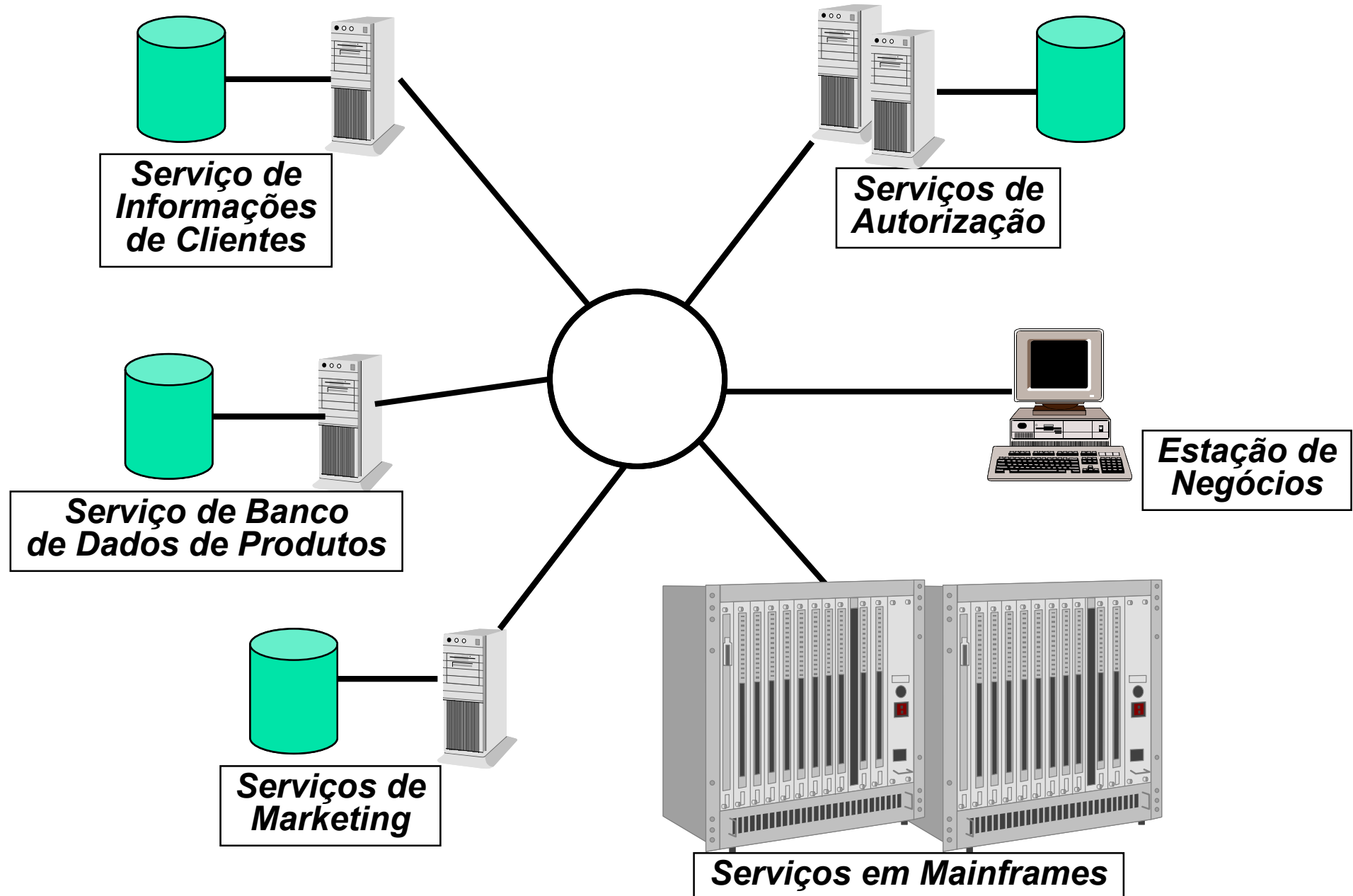


# O que é um Sistema Distribuído?

---



# Infraestrutura do Setor Bancário



# Requisitos

---

- Tempo para se chegar ao mercado
  - Desenvolvimento de novas aplicações com tecnologia recente
  - Integração de novas aplicações tem se tornado cada vez mais difícil
- Escalabilidade
  - Administração de milhões de contas e clientes
  - Milhares de usuários concorrentes
- Confiabilidade

## Requisitos (cont.)

---

- Heterogeneidade de Hardware
  - Mainframes (Unisys, IBM, etc.)
  - Servidores SUN SPARC
  - PCs
- Heterogeneidade de Sistema Operacional
  - VMs, UNIX, Linux, Windows, OSX
- Heterogeneidade de Linguagem de Programação
  - PHP, C/C++, Visual Basic, Java

# Tecnologia de Objetos Distribuídos

---

- Componentes de prateleira encapsulam a funcionalidade da aplicação
- Resolvendo o problema de distribuição em um nível mais elevado de abstração
- Resolvendo o problema da heterogeneidade
- Escalabilidade da solução

# Por que Tecnologia de Objetos Distribuídos

---

- Permite uma **visão uniforme** de todos os serviços da empresa e de como acessá-los
- Provê um nível apropriado de **abstração**
- Preserva o investimento **encapsulando aplicações legadas**
- Permite explorar as vantagens da tecnologia de objetos em novos projetos
- Uma forma natural de resolver:
  - distribuição
  - heterogeneidade



# Compartilhamento de Recursos

---

- Habilidade de usar qualquer hardware, software ou dados em qualquer lugar do sistema
- Gerenciador de recursos
  - Controla o acesso aos recursos
  - Provê um esquema de nomes para os recursos
  - Controla acessos concorrentes aos recursos

# Compartilhamento de Recursos (2)

- Modelo de compartilhamento
  - Cliente / Servidor
  - Baseado em objetos
- Define:
  - a forma pela qual recursos são providos
  - formas de uso dos recursos
  - como o provedor do recurso e os usuários interagem entre si e com o gerenciador

---

# Cases

# Case 1: Gerência de Configuração Boeing 777

---



# Problemas a serem resolvidos

---

## ■ Escala

- 3.000.000 de peças por aeronave
- A configuração de cada aeronave é diferente
- Regulamentos demandam que registros sejam mantidos para cada peça de uma aeronave
- Aeronave evolui durante manutenções
- Produção de 500 aeronaves por ano
- Banco de dados de configuração cresce 1,5 bilhão de partes a cada ano
- Tempo de vida de uma aeronave: 30 anos
- 45.000 engenheiros necessitam acesso on-line aos dados de configurações

## Problemas a serem resolvidos (cont.)

- Integração de componentes de prateleira
  - Infra-estrutura de TI se tornou inadequada
  - Mas a empresa não podia se dar ao luxo de re-construir toda a sua infra-estrutura de TI
  - Componentes foram comprados de diversos fabricantes especializados
    - Banco de dados relacional
    - Planejamento de recursos da empresa (ERP)
    - Planejamento de projetos auxiliado por computador
  - Componentes precisavam ser integrados

## Problemas a serem resolvidos (cont.)

- Heterogeneidade
  - 20 máquinas de banco de dados Sequent para gerenciar os dados de configuração de aviões
  - 200 servidores de aplicações UNIX
  - Estações de trabalho NT e UNIX para os engenheiros

## Case 2: Napster

---





# Outros Exemplos Similares

---

- Similares ao NAPSTER
  - EMULE
  - BITORRENT
  - SKYPE (VoIP)
  - MSN MESSENGER, YAHOO
- Outros
  - APLICAÇÕES DE MULTIMÍDIA
    - WEB conferência
    - Difusão de streams on-line (voz, vídeo)

## Case 3: Cluster (JAGUAR)

---



### JAGUAR

Cluster de máquinas do Oak Ridge National Laboratory (EUA)

Primeiro da lista dos TOP 500 SUPERCOMPUTERS

INSTALADO EM 2009

224.162 CORES 1.759.000 Gflops

FONTE <http://www.top500.org/list/2010/06/100>

**Solução: agrupamento físico de máquinas de menor porte = cluster:**

- Mesmo espaço geográfico
- Normalmente dentro da mesma organização, com software/hardware homogêneos
- Softwares: OpenMosix, Condor e Oscar

**Cluster atual:**

<https://tecnoblog.net/104684/ibm-sequoia-supercomputador/>

<http://www.top500.org/list>

# Case: Cluster

---

## CLUSTERS ou AGRUPAMENTOS

- Máquinas conectadas por rede de alta velocidade
- Necessidade de alto desempenho computacional
- Evitar custo de máquinas de alto desempenho
- Objetivo: compartilhar recursos computacionais
  - De processamento
  - De armazenamento
  - De memória
  - Outros

# Cluster

---

## Clusters são utilizados para...

- Aumentar a disponibilidade de serviço
  - se um nodo falha, outro assume
- Equilibrar carga de trabalho
  - um ou mais computadores do cluster atuam como distribuidores da carga entre os demais
- Alto desempenho
  - Para resolver tarefas complexas que podem ser decompostas em sub-tarefas, cada uma rodando num nodo do cluster.
  - Implementação mais comum: LINUX e software livre para implementar paralelismo = Beowulf cluster

# Desafios

## □ Escalabilidade –

- ▣ O sistema distribuído permanece eficiente quando há um aumento significativo no número de recursos e no número de usuários.

<i>Date</i>	<i>Computers</i>	<i>Web servers</i>
1979, Dec.	188	0
1989, July	130,000	0
1999, July	56,218,000	5,560,866
2003, Jan.	171,638,297	35,424,956

Computadores (com endereços IP registrados) na Internet.

# Desafios

- A heterogeneidade implica em diferentes mecanismos e formatos de dados:
  - ▣ Tipos de dados podem ser armazenados de formas diferentes, o que requer negociar formatos padronizados para a comunicação.
  - ▣ Linguagens de programação diferentes também utilizam diferentes formatos para armazenar estruturas de dados.
  - ▣ Chamadas para comunicação podem ser diferentes em sistemas operacionais distintos, como Windows e Linux, o que pode ser um desafio em ambientes multiplataforma.



# Desafios

- Abordagens para heterogeneidade:
  - ▣ **Middleware**: camada de software que provêm uma abstração determinando um **modo de programação** que pode ser utilizado para mascarar a heterogeneidade de plataformas
    - Diferentes middlewares serão abordados no curso.
  - ▣ Em ambientes com migração de código (**código móvel**, que pode ser enviado de um computador para outro, não necessariamente de mesma plataforma - *applets*), uma máquina virtual pode ser necessária para prover o ambiente de execução do código móvel, provendo inclusive isolamento do ambiente local.

# Desafios

## ❑ Sistemas abertos

- ▣ Pode ser estendido e reimplementado de várias maneiras;
- ▣ Definido na especificação e documentação – principais interfaces são publicadas;
- ▣ Entender a complexidade dos sistemas distribuídos, formados por muitos componentes;
- ▣ Padronização de protocolos
  - ▣ Exemplo para a Internet – RFCs;
- ▣ Sistemas distribuídos abertos – projetados a partir de padrões públicos; podem ser construídos a partir de software e hardware heterogêneos.



# Desafios



- Aspectos de segurança (*security*) devem ser considerados:
  - ▣ Confidencialidade – proteção contra exposição para pessoas não autorizadas;
  - ▣ Integridade – proteção contra alteração ou dano;
  - ▣ Disponibilidade – proteção contra interferência com os meios de acesso aos recursos;
  - ▣ Autenticidade – garantia de identidade de quem acessa.

# Desafios

- Tratamento de falhas –
  - ▣ Componentes (software e hardware) de um sistema distribuído podem falhar, podendo gerar erros nos resultados do sistema.
  - ▣ Técnicas:
    - Detecção de falhas: algumas falhas podem ser detectadas (ex. paridade); gerenciar falhas que não podem ser detectadas mas que podem ser suspeitas (sistemas assíncronos);
    - Mascaramento de falhas: ocultar falhas ( ex. retransmitir mensagens perdidas/ replicar dados); nem sempre é possível;

# Desafios



- Concorrência –
  - ▣ Solicitações concorrentes a recursos compartilhados devem ser atendidas de forma concorrente, sem gerar conflitos ou resultados inconsistentes;
  - ▣ Utilizar diferentes fluxos de execução (*threads*) nos servidores para atender a cada solicitação;
  - ▣ Sincronização da execução concorrente: uso de técnicas como semáforos ou monitores.



# A evolução dos sistemas distribuídos

---

principais eventos

# Breve Histórico

---

- Apareceu no fim da década de 60 dentro do contexto de Sistemas Operacionais.
- A motivação foi a criação de unidades de *hardware* denominadas canais ou dispositivos de controle.
- Estes dispositivos funcionavam independente de um processador de controle e podiam fazer operações de E/S concorrentemente com a execução de um programa.

## Breve Histórico (2)

---

- Um canal que comunicava-se com o processador central através de uma interrupção.
- Com a introdução dos canais, partes de um programa poderiam funcionar de forma imprevisível.
- Logo após o aparecimento dos canais, foram desenvolvidos as máquinas multiprocessadas.
- Estas máquinas permitiam que aplicações diferentes pudessem ser executadas em processadores diferentes ao mesmo tempo.

## Breve Histórico (3)

---

- Permitiam também que uma aplicação pudesse ser executada mais rapidamente se fosse reescrita de forma a utilizar múltiplos processadores.

# ARPANet

---

- Advanced Research Projects Agency Network - 1969
- Departamento de Defesa dos EUA
  - bases militares e departamentos de pesquisa do governo
  - universidades e outras instituições que faziam trabalhos envolvidos à defesa
- primeira rede à base de comutação de pacotes
  - backbone subterrâneo
  - mainframes
  - NCP (Network Control Protocol)
  - compartilhamento de arquivos, acesso remoto, correio eletrônico
- precursora da Internet
  - ARPANet (academia) e MILNET (militar)
  - TCP/IP ( Transfer Control Protocol/Internet Protocol) – 1983



# Ethernet

---

- Protocolo para rede local de computadores
- Camadas física (1) e de enlace (3) da rede
- Xerox PARC, 1973, Robert Metcalfe
- inspirado na ALOHAnet: rede por rádio, Hawaii, anos 60
- padrão de fato: DEC, Intel e Xerox, 1980
- produção em escala: baixo custo
- computadores pessoais

# Internet

---

- DNS (Domain Name System) – 1984
- FTP (File Transfer Protocol) – 1985
- SNMP (Simple Mail Transfer Protocol)
- POP3 (Post Office Protocol)
- IMAP (Internet Message Access Protocol)
- SSH (Secure Shell)
- VNC (Virtual Network Computing)
- computadores pessoais

# World Wide Web – WWW

---

- CERN (Suíça)
- Tim Berners-Lee
- 1980: ENQUIRE
  - compartilhamento de informações sobre pesquisa
  - sistema de hipertexto (origem em 1960: Projeto Xanadu)
- 1990: World Wide Web
  - união com a Internet
  - sistema cliente-servidor
  - HTTP (HyperText Transfer Protocol)
  - HTML (HyperText Markup Language)
  - URI (Uniform Resource Identifier)
- 1993:
  - CERN libera a WWW
  - Navegador Mosaic, precursor do Netscape

# Motores de Busca

---

- Information Retrieval
  - início: 1970 (aprox.)
  - formalização do conceito de hipertexto: 1972
  - consolidação: 1978 (aprox.)
    - modelos probabilísticos
- Yahoo!
- Altavista
- Google:
  - alta disponibilidade
  - desempenho
  - escalabilidade
  - baixo custo operacional

# Colaboração na Internet

---

- Mensagem instantânea (síncronos)
- Chats (assíncronos)
- Vídeoconferência (streams)
- Redes sociais
- Wiki

# Transmissão de Mídia

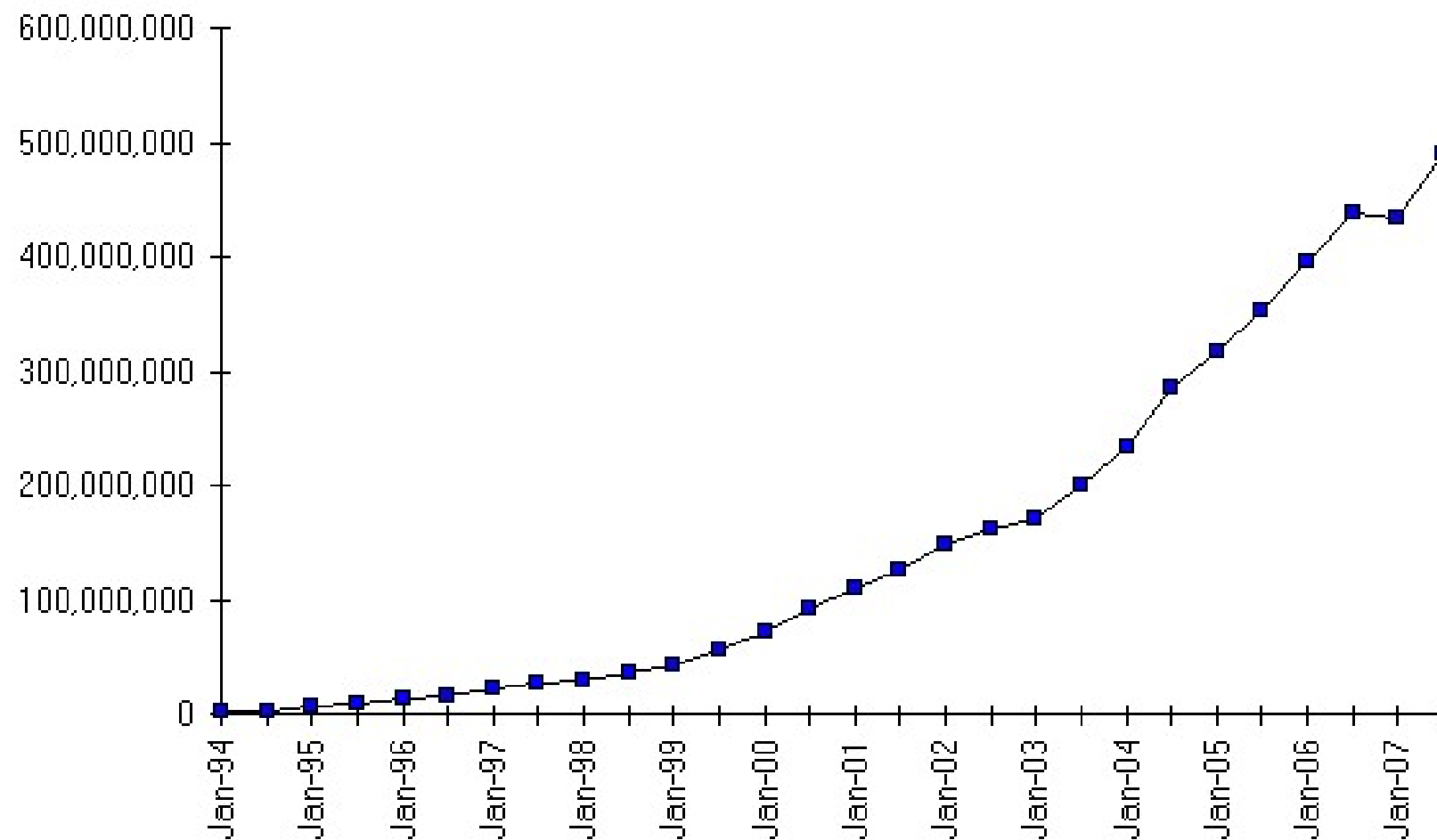
---

- canais de rádio e TV
- podcast
- voz/telefonia sobre IP (redes convergentes)
- Napster (1999 – 2002)
- BitTorrent – 2003 (Peer-to-Peer)
- eDonkey – 2000
- KaZaA – 2002

# Internet – Número de Hosts

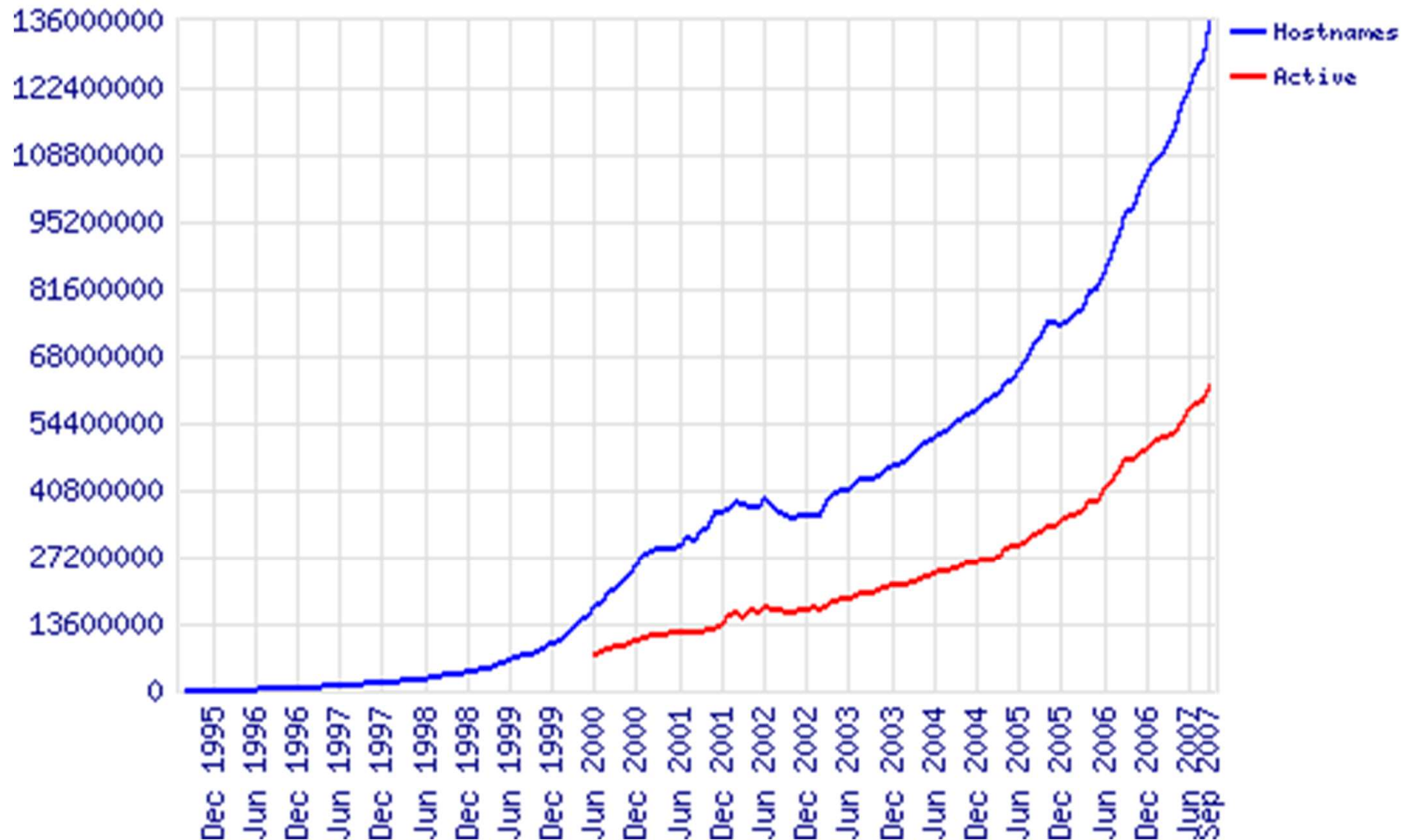
---

Internet Domain Survey Host Count



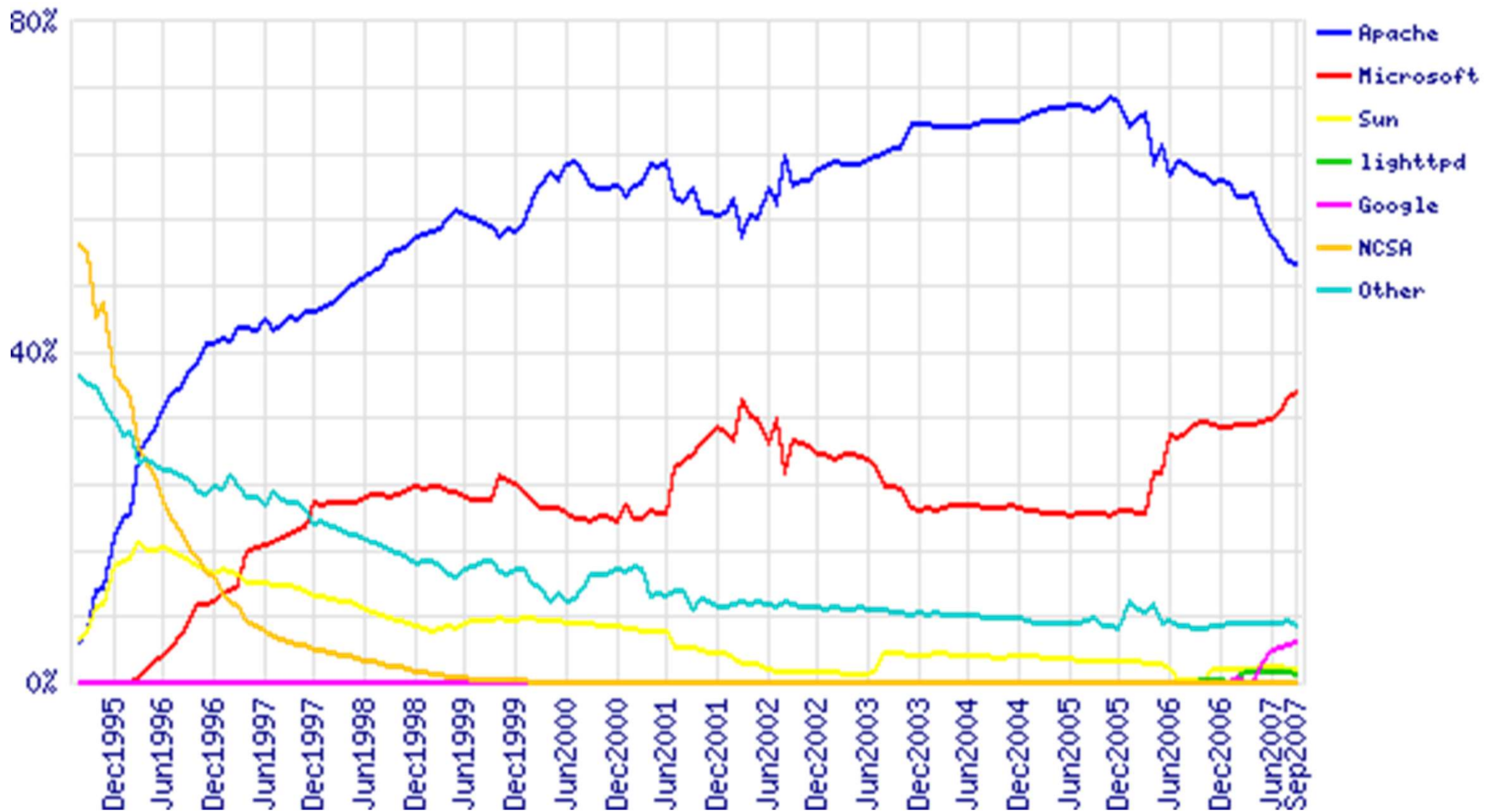
Source: Internet Systems Consortium ([www.isc.org](http://www.isc.org))

# Internet – Número de Servidores Web





# Internet – Mercado de Servidores Web





# Os benefícios de sistemas distribuídos

---

# Benefícios

---

- Comunicação de dados
- Compartilhamento de recursos
- Integração e cooperação
- Confiabilidade e disponibilidade
- Crescimento gradativo do poder computacional
- Relação custo/benefício
- Implantação de aplicações



# Os problemas em sistemas distribuídos

---

# Paradigma de programação

---

- processos e mensagens
- chamada remota de procedimento
- memória compartilhada distribuída
- eventos (*publish/subscribe*)

# Depuração de aplicações distribuídas

---

- difícil reprodução de erros
- teste isolados e testes de integração
- ambientes de simulação
- alto custo de desenvolvimento
- alto custo de manutenção

# Confiabilidade e disponibilidade do meio

---

- falhas de rede
- falhas de processadores
- falhas
  - *fail-silent*
  - bizantinas
- redundância: alto custo
  - física
  - lógica

# Tipos de falhas

---

Classe da falha	Afeta	Descrição
Parada por falha	Processo	O processo pára e permanece parado. Outros processos podem detectar esse estado.
Colapso	Processo	O processo pára e permanece parado. Outros processos podem não detectar esse estado.
Omissão	Canal	Uma mensagem inserida em um <i>buffer</i> de envio nunca chega no <i>buffer</i> de recepção do destinatário.
Omissão de envio	Processo	Um processo conclui um envio, mas a mensagem não é colocada em seu <i>buffer</i> de envio.
Omissão de recepção	Processo	Uma mensagem é colocada no <i>buffer</i> de recepção de um processo, mas esse processo não a recebe efetivamente.
Arbitrária (bizantina)	Processo ou canal	O processo/canal exibe comportamento arbitrário: ele pode enviar/transmitir mensagens arbitrárias em qualquer momento, cometer omissões; um processo pode parar ou realizar uma ação incorreta.



# Segurança do meio

---

- ataques
  - acessam serviços que atendem via rede
  - exploram fragilidades de código
    - alteram o código do serviço
    - instalam código “maléfico” ao sistema
  - fazem sobrecarga de requisições
- criptografia
- assinatura digital
- autenticação
- autorização
- firewalls

# Desempenho das aplicações

---

- latência
- não-determinismo
- exigências de tempo real
  - sincronismo de eventos

# Consistência das aplicações

---

- sincronismo de relógios físicos
- sincronismo de eventos (relógios lógicos)
- atomicidade no envio de mensagens
- acesso concorrente
  - compartilhamento de recursos
  - servidores *multi-threaded*
- transação distribuída
- tolerância a falhas
  - backward recovery
  - forward recovery

# Escalabilidade dos sistemas

---

- desempenho estável
  - crescimento no número de usuários
  - crescimento no volume de dados
  - crescimento no volume de requisições
- técnicas básicas
  - replicação
  - cache

# Gerações de Sistemas Distribuídos

---

<i>Sistemas distribuídos</i>	<i>Primitivos</i>	<i>Adaptados para Internet</i>	<i>Contemporâneos</i>
<i>Escala</i>	Pequenos	Grandes	Ultragrandes
<i>Heterogeneidade</i>	Limitada (normalmente, configurações relativamente homogêneas)	Significativa em termos de plataformas, linguagens e <i>middleware</i>	Maiores dimensões introduzidas, incluindo estilos de arquitetura radicalmente diferentes
<i>Sistemas abertos</i>	Não é prioridade	Prioridade significativa, com introdução de diversos padrões	Grande desafio para a pesquisa, com os padrões existentes ainda incapazes de abranger sistemas complexos
<i>Qualidade de serviço</i>	Em seu início	Prioridade significativa, com introdução de vários serviços	Grande desafio para a pesquisa, com os serviços existentes ainda incapazes de abranger sistemas complexos

# Heteregoneidade

---

- protocolos de rede
- sistemas operacionais
- linguagens de programação



# Transparência em Sistemas Distribuídos

---

# Transparência

---

- acesso
- localização
- falha
- replicação
- migração
- concorrência
- desempenho
- paralelismo
- escalabilidade
- persistência
- transação



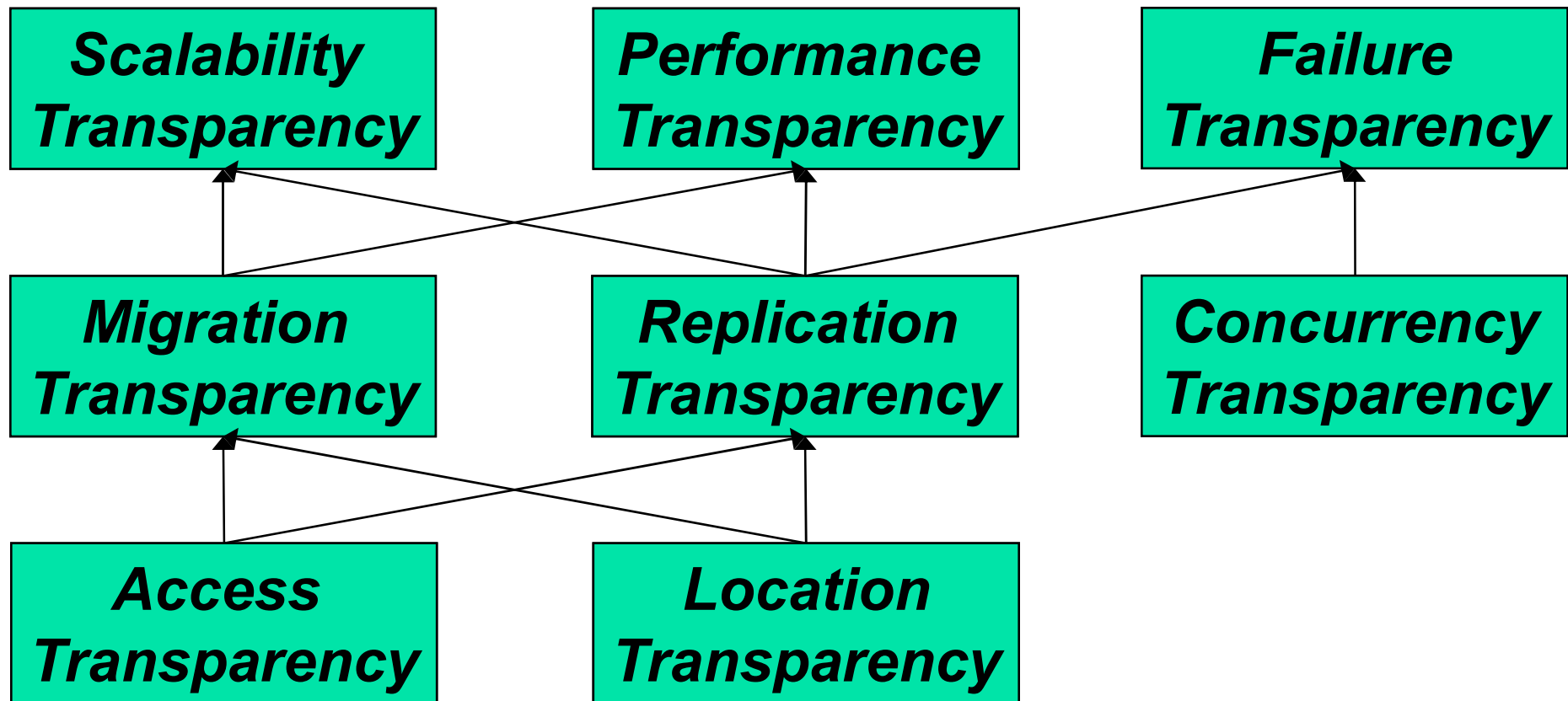
# Transparência

---

- Um sistema distribuído deve ser percebido por seus usuários e pelos programadores de aplicações como um sistema único e coeso
  - ao invés de uma coleção de máquinas separadas
- Várias dimensões de transparência identificadas pelo modelo ISO RM-ODP
  - Modelo de Referência para Sistemas Distribuídos Abertos
- Representam as diversas propriedades que um sistema distribuído deve possuir

# Transparências de Distribuição

---



# Transparência de Acesso

---

- Permite que objetos e informações remotas sejam acessados usando operações idênticas
- Mascara as diferentes formas de acesso empregadas por cada tecnologia utilizada
- Exemplos:
  - Operações de acesso a um sistema de arquivos distribuído com NFS (*Network File System*)
  - Navegação na WEB
  - Consultas em SQL

# Transparência de Localização

---

- Permite que objetos e informações sejam acessados sem o conhecimento de sua localização
- Exemplos:
  - Arquivos acessados via NFS
  - Páginas na WEB (\*)
  - Tabelas em um banco de dados distribuído

# Transparência de Concorrência

---

- Permite que vários processos operem concorrentemente usando objetos de informação compartilhados sem interferirem entre si
- Exemplos:
  - NFS
  - Caixa eletrônico
  - Sistema gerenciador de bancos de dados (SGBD)

# Transparência de Replicação

---

- Permite que múltiplas instâncias de objetos de informação sejam usados para melhorar o desempenho e a confiabilidade
- Sem que os usuários ou programadores de aplicações tomem conhecimento da existência das réplicas
- Exemplos:
  - SGBD distribuído
  - Espelhamento de páginas WEB

# Transparência de Falhas

---

- Mascara a ocorrência de falhas
- Permite que usuários e aplicações completem suas tarefas normalmente a despeito de falhas em alguns componentes do sistema
- Exemplo:
  - Transações em um SGBD

# Transparência de Migração

---

- Permite a movimentação de um objeto dentro do sistema distribuído sem afetar as operações dos usuários ou dos programas de aplicação
- Duas variantes:
  - **Migração propriamente dita:** com relação ao objeto migrado
  - **Relocação:** com relação a outros objetos no sistema
- Exemplos:
  - NFS
  - Páginas WEB



# Transparência de Desempenho

---

- Permite que o sistema distribuído seja reconfigurado para melhorar o desempenho para refletir mudanças na carga de processamento
- Através de replicação e migração
- Exemplo:
  - Utilitário **make** distribuído
    - Programa é compilado em várias máquinas em paralelo, transparentemente para o usuário

# Transparência de Escala

---

- Permite que o sistema e as aplicações possam ser expandidos em escala sem a necessidade de mudanças em sua estrutura ou nos algoritmos utilizados
- Exemplo:
  - WWW
  - Bancos de dados distribuídos

# Pontos-Chave

---

- O que é um Sistema Distribuído
- Adoção de sistemas distribuídos é regida por requisitos não-funcionais
- Necessidades de distribuição são transparentes aos usuários e projetistas de aplicações
- Várias dimensões de transparência
- Dimensões de transparência dependem entre si