

Introdução

Sistemas Distribuídos - ECOS02

Prof. Bruno Tardiole Kuehne

brunokuehne@unifei.edu.br

IESTI



1. Programa
2. Introdução
3. Exemplos de Sistemas Distribuídos
4. Enfoque no Compartilhamento de Recursos
5. Desafios
6. Tipos de Sistemas Distribuídos

Programa



1. N1

- Exercícios Semanais - 20%
- 30/04/2024 - P1 - 80%

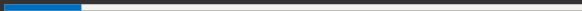
2. N2

- Exercícios Semanais - 20%
- 11/06/2024 - P2 -50%
- 18 e 25/06/2024 - Seminários - 30%

3. 02/07/2024 - Sub

- ⊙ Introdução
- ⊙ Arquitetura
- ⊙ Processos
- ⊙ Sincronização
- ⊙ Objetos Distribuídos
- ⊙ Web Services
- ⊙ Transações Distribuídas
- ⊙ Deadlock Distribuído
- ⊙ Tolerância a Falha

Introdução



Definição

“Definimos um sistema distribuído como aquele no qual os componentes de hardware e software, localizados em computadores interligados em rede, comunicam-se e coordenam suas ações apenas enviando mensagens entre si.”
(Coulouris, 2013)

Definição

“Um sistema distribuído é um conjunto de computadores independentes que se apresenta a seus usuários como um sistema único e coerente.” (Tanenbaum, 2007)

Consequências de um sistema distribuídos:

- ⊙ Concorrência: devido ao compartilhamento de recursos é uma característica recorrente em SD.
- ⊙ Inexistência de relógio global: A comunicação é basicamente feita por mensagens, mas como ordená-las em um ambiente distribuído?
- ⊙ Falhas independentes: como tratar falhas de componentes individuais de um sistemas distribuído.

Exemplos de Sistemas Distribuídos

Exemplos de sistemas distribuídos

<i><u>Finance and commerce</u></i>	eCommerce e.g. Amazon and eBay, PayPal, online banking and trading
<i>The information society</i>	Web information and search engines, ebooks, Wikipedia; social networking: Facebook and MySpace.
<i>Creative industries and entertainment</i>	online gaming, music and film in the home, user-generated content, e.g. YouTube, Flickr
<i>Healthcare</i>	health informatics, on online patient records, monitoring patients
<i>Education</i>	e-learning, virtual learning environments; distance learning
<i>Transport and logistics</i>	GPS in route finding systems, map services: Google Maps, Google Earth
<i>Science</i>	The Grid as an enabling technology for collaboration between scientists
<i>Environmental management</i>	sensor technology to monitor earthquakes, floods or tsunamis

Exemplos de sistemas distribuídos

Pesquisa na Web

- ⦿ Objetivo é indexar todo conteúdo da Web
- ⦿ Web consiste em mais de 63 bilhões de páginas
- ⦿ Destaque a empresa Google que possui uma das maiores e mais complexas instalações de SD
 - Infraestrutura física localizadas em centros de dados espalhados pelo mundo
 - Sistema de arquivo distribuído projetado para suportar arquivos grandes e otimizado para as aplicações Google

Massively multiplayer online games (MMOGs)

- ⊙ Oferecem uma experiência imersiva com a qual um número muito grande de usuário interage com um mundo virtual persistente
- ⊙ Sistemas são capazes de suportar mais de 50.000 jogadores simultâneos.
- ⊙ MMOGs representa um grande desafio e o projeto adequado de infraestrutura de comunicação pode ser um ponto chave para o sucesso de títulos do estilo.

Exemplos de sistemas distribuídos

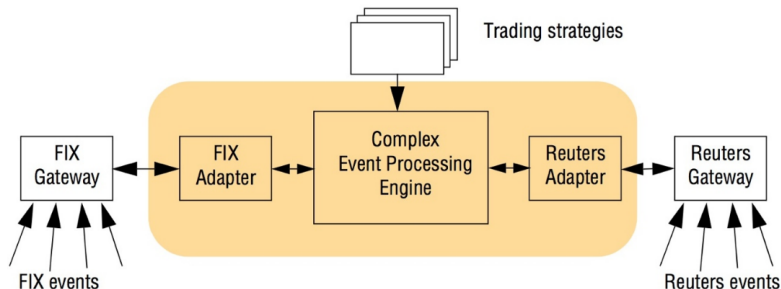
Soluções para projetos de MMOGs:

- ⊙ Arquitetura cliente-servidor: Servidor formado por *clusters*. Com a utilização de servidor centralizado, o gerenciamento do mundo virtual se torna mais simples já que todo o ambiente está em um único lugar.
- ⊙ Arquiteturas distribuídas: Servidores espalhados geograficamente. Usuários são alocados dinamicamente de acordo com a disponibilidade e tempo de resposta favorável a utilização.
- ⊙ Pesquisas tem colocado esforço em abordagens completamente descentralizadas por meio de arquiteturas *peer-to-peer*.

Exemplos de sistemas distribuídos

Negócios financeiros

- ⦿ Complexidade na ampla variedade de informações a serem consultadas em tempo real.
- ⦿ Fontes normalmente em formatos variados: dados de mercados(Reuters) e eventos FIX(Financial Information eXchange)



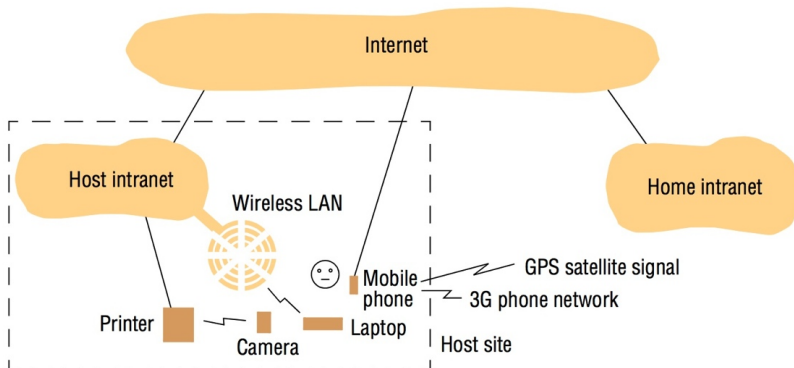
Computação móvel e ubíqua

- ⊙ Computação móvel é a capacidade do usuário se deslocar sem perder conectividade
- ⊙ Computação ubíqua ou pervasiva
 - Utilização de vários dispositivos pequenos e baratos espalhados pelo ambiente físico do usuário
 - Computação baseada em contexto

Exemplos de sistemas distribuídos

Computação móvel e ubíqua

- ⊙ Computação ubíqua e móvel se sobrepõe
- ⊙ Usuário visitante pode descobrir serviços em uma rede anfitriã



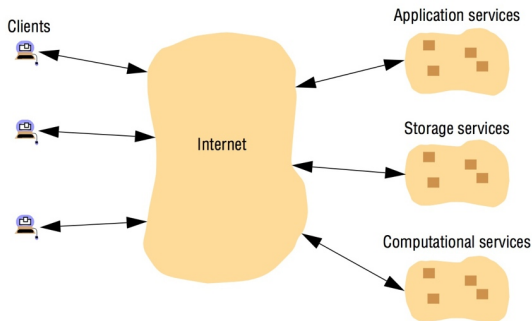
Sistemas multimídia distribuídos

- ⊙ Multimídia é a capacidade de manipular e apresentar diversos tipos de mídia. (texto, vídeo, áudio)
- ⊙ Entrega de conteúdo de maneira contínua e integrada ao usuário
- ⊙ Nova maneira de utilização de equipamentos (Tvs, Celulares, Computadores).

Exemplos de sistemas distribuídos

Computação distribuída como um serviço público

- ⦿ Recursos computacionais oferecidos como serviços (Nuvem)
- ⦿ Armazenamento, Aplicativos, Infraestrutura de hardware sempre acessados como serviços



Enfoque no Compartilhamento de Recursos

Enfoque no Compartilhamento de Recursos

Muito natural nos dias de hoje e chega a ser imperceptível

- ⊙ Redução de custos: Impressoras e discos
- ⊙ Compartilhamento de Informação: trabalhos colaborativos, redes sociais, etc.
 - Usuário tem enfoque em compartilhar a informação sem se preocupar em qual recurso físico isso será armazenado.

Desafios

Heterogeneidade

- ⊙ Redes
- ⊙ Hardware de computador
- ⊙ Sistemas operacionais
- ⊙ Linguagem de programação
- ⊙ Implementações de diferentes desenvolvedores

⊙ Heterogeneidade

○ Middleware

- CORBA – suporta várias linguagens de programação, feito por diferentes desenvolvedores – muitas vezes incompatíveis entre si
- RMI – exclusivo para JAVA

⊙ Heterogeneidade e migração de código

- Sistema operacional rodando em um servidor muitas vezes é diferente do utilizado no cliente, o que resulta no não funcionamento do código no cliente
 - Java applet foi uma maneira de resolver esse problema.

Sistemas Abertos

- ⦿ Sistema que pode ser estendido e reimplementado de várias maneiras
- ⦿ Sistemas distribuídos abertos são baseados na utilização de interfaces que especificam como o recurso deve ser utilizado
- ⦿ Para o funcionamento correto a compatibilidade entre os componentes (hardware e software) devem ser testados

Seguranças

- ⊙ Ataque cibernético é um problema recorrente. Muitas vezes com cunho financeiro, outras vezes simplesmente para derrubar algum serviço de alguma instituição e também roubar informações estratégicas
- ⊙ Confidencialidade
- ⊙ Integridade
- ⊙ Autenticações
- ⊙ Segurança Operacional

Escalabilidade

- ⦿ Sistemas devem ser projetados para crescer
- ⦿ Controlar o custo dos recursos físicos: a adição de um novo usuário deve ter um custo baixo
- ⦿ Controlar a perda de desempenho
- ⦿ Impedir que recursos de software se esgotem
- ⦿ Evitar gargalos de desempenho

Tratamento de falhas

- ⦿ Detecção de Falhas: checksum, CRC, etc.
 - Retransmissão da mensagem
 - Redundância de discos
- ⦿ Tolerância a falhas: não travar o sistema devido a uma falha
- ⦿ Recuperação de falhas: reestabelecer o sistema para o último estado consistente antes da falha
- ⦿ Redundância de serviços: servidores replicados podem atuar quando algum apresentar problema

Concorrência

- ⦿ Recursos compartilhados devem ser acessados de maneira organizada para não haver erros de consistência

Escalabilidade

- ⦿ Sistemas devem ser projetados para crescer
- ⦿ Controlar o custo dos recursos físicos: a adição de um novo usuário deve ter um custo baixo
- ⦿ Controlar a perda de desempenho
- ⦿ Impedir que recursos de software se esgotem
- ⦿ Evitar gargalos de desempenho

Transparência

- ⊙ Transparência de acesso: permite que recursos locais e remotos sejam acessados com o uso de operações idênticas
- ⊙ Transparência de localização: permite acesso a recurso sem conhecimento de endereço(físico e lógico)
- ⊙ Transparência de concorrência: permite que vários processos operem concorrentemente, usando recursos compartilhados sem interferência entre eles

Transparência

- ⦿ Transparência de replicação: permite várias instâncias dos recursos sejam usados para aumentar a confiabilidade e o desempenho, sem conhecimento das réplicas por parte dos usuários
- ⦿ Transparência de falhas: permite ocultação de falhas, possibilitando que usuários concluam suas tarefas

Transparência

- ⊙ Transparência de mobilidades: permite a movimentação de recursos e clientes dentro de um sistema, sem afetar o modo de uso dos mesmo
- ⊙ Transparência de desempenho: permite que o sistema seja reconfigurado para melhorar o desempenho a medida que as cargas variam
- ⊙ Transparência de escalabilidade: permite que o sistema cresça sem alterar a estrutura do sistema

Qualidade de Serviço

- ⊙ Garantia de propriedades não funcionais dos sistemas:
- ⊙ Confiabilidade
- ⊙ Segurança
- ⊙ Desempenho
- ⊙ Adaptabilidade

Tipos de Sistemas Distribuídos

Tipos de Sistemas Distribuídos

Sistemas de computação distribuída

⦿ Computação de cluster

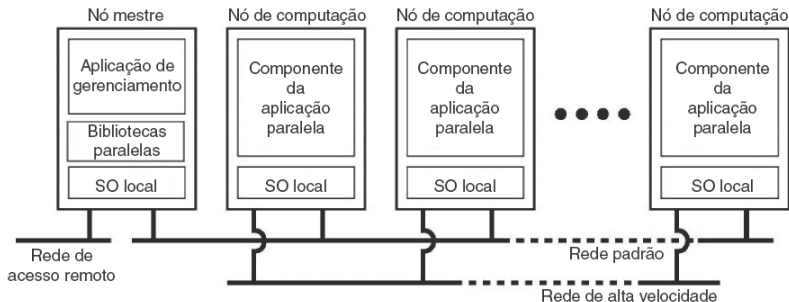


Figura 1.4 Exemplo de um sistema de computação de cluster.

Sistemas de computação distribuída

⊙ Sistema de processamento de transações

- Atômicas
- Consistentes
- Isoladas
- Duráveis

Primitiva	Descrição
BEGIN_TRANSACTION	Marque o início de uma transação
END_TRANSACTION	Termine a transação e tente comprometê-la
ABORT_TRANSACTION	Elimine a transação e restaure os valores antigos
READ	Leia dados de um arquivo, tabela ou de outra forma
WRITE	Escreva dados para um arquivo, tabela ou de outra forma

Tabela 1.3 Exemplos de primitivas para transações.

Tipos de Sistemas Distribuídos

Sistemas de computação distribuída

⦿ Integração de aplicações empresariais

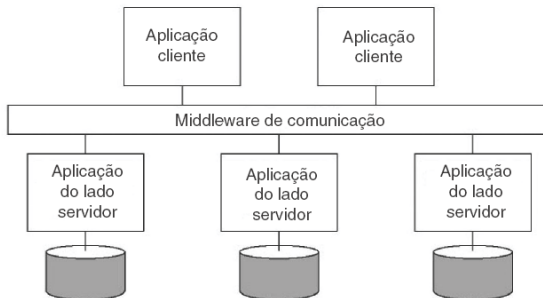




Figura 1.8 Middleware como facilitador de comunicação em integração de aplicações empresariais.

-  COULOURIS, George et al
Sistemas Distribuídos: Conceitos e Projeto
Bookman Editora, 2013
-  TANEMBAUM, Andrew S.; V. STEEN, Maarten
Sistemas Distribuídos: Princípios e Paradigmas
Pearson Prentice Hall, 2007

DÚVIDAS?