



Universidade Tecnológica Federal do Paraná
Departamento Acadêmico de Computação
Bacharelado em Ciência da Computação

Sistemas Distribuídos

Introdução e Caracterização

Prof. Rodrigo Campiolo

12/08/19

Tópicos

- Introdução
- Exemplos
- Tendências
- Características

Introdução

- O que é um Sistema Distribuído (SD)?



Fonte: <http://www.ejbtutorial.com/wp-content/uploads/2013/09/distributed-networks-298x248.jpg>

Introdução

- Várias definições
 - **Mullender:** “Um SD é um sistema com vários EP e vários DA, conectados entre si por uma rede”.
 - **Lampport:** “Um SD é aquele que não permite que você faça seu trabalho quando ocorre uma pane em uma máquina que você nem sabia que existia”.
 - **Tanenbaum:** “Uma coleção de computadores independentes que aparentam a seus usuários como um único sistema coerente”.
 - **Kshemkalyani e Singhal:** “Uma coleção de entidades independentes que cooperam para resolver um problema que não poderia ser resolvido individualmente”.
 - **Coulouris:** “Um sistema distribuído é aquele no qual os componentes localizados em computadores interligados em rede se comunicam e coordenam suas ações apenas passando mensagens”.

Introdução

- Definição

Um Sistema Distribuído consiste em um conjunto de **computadores independentes interligados por uma rede**, onde os componentes de **software e hardware se comunicam e coordenam** suas ações por meio de **troca de mensagens**. Além disso, esse conjunto se apresenta como um **único sistema coerente**.

Introdução

- Motivação
 - Compartilhamento de recursos
 - Realização de múltiplas tarefas
 - Facilidade de acesso a dados e recursos remotos
 - Aplicações inerentemente distribuídas
 - Confiabilidade
 - Taxa desempenho/custo
- **Algo mais?**

Introdução

- Características
 - **Concorrência:** acesso a recursos e execução de programas concorrentemente.
 - **Inexistência de relógio global:** coordenação por troca de mensagens.
 - **Independência de falhas:** componentes podem falhar independentemente (rede, máquina, software).

Exemplos

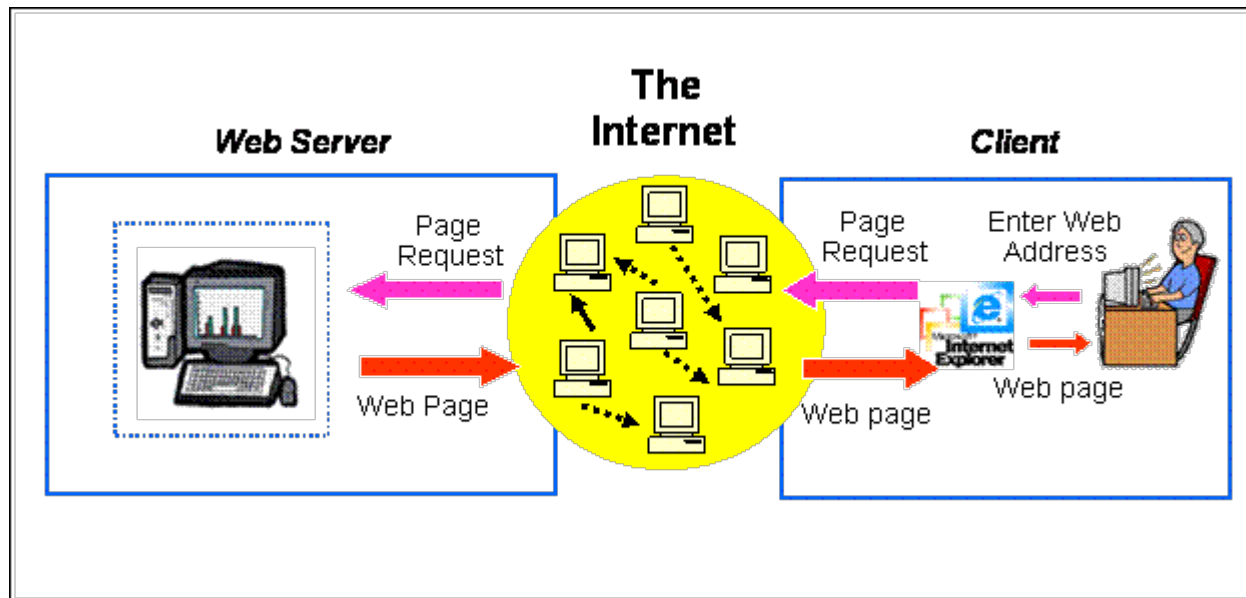
- Motores de busca
- Sistemas de reserva de viagens
- Sistemas de agências bancárias
- Jogos online
- Sistemas de compartilhamento de arquivos (p. ex. Bittorrent)
- Sistemas de controle acadêmico
- **Web**

Exemplos

- Web
 - Publicação e acesso de recursos e serviços na Internet
 - Componentes principais
 - HTML
 - URL
 - HTTP
 - Arquitetura cliente/servidor

Exemplos

- Web

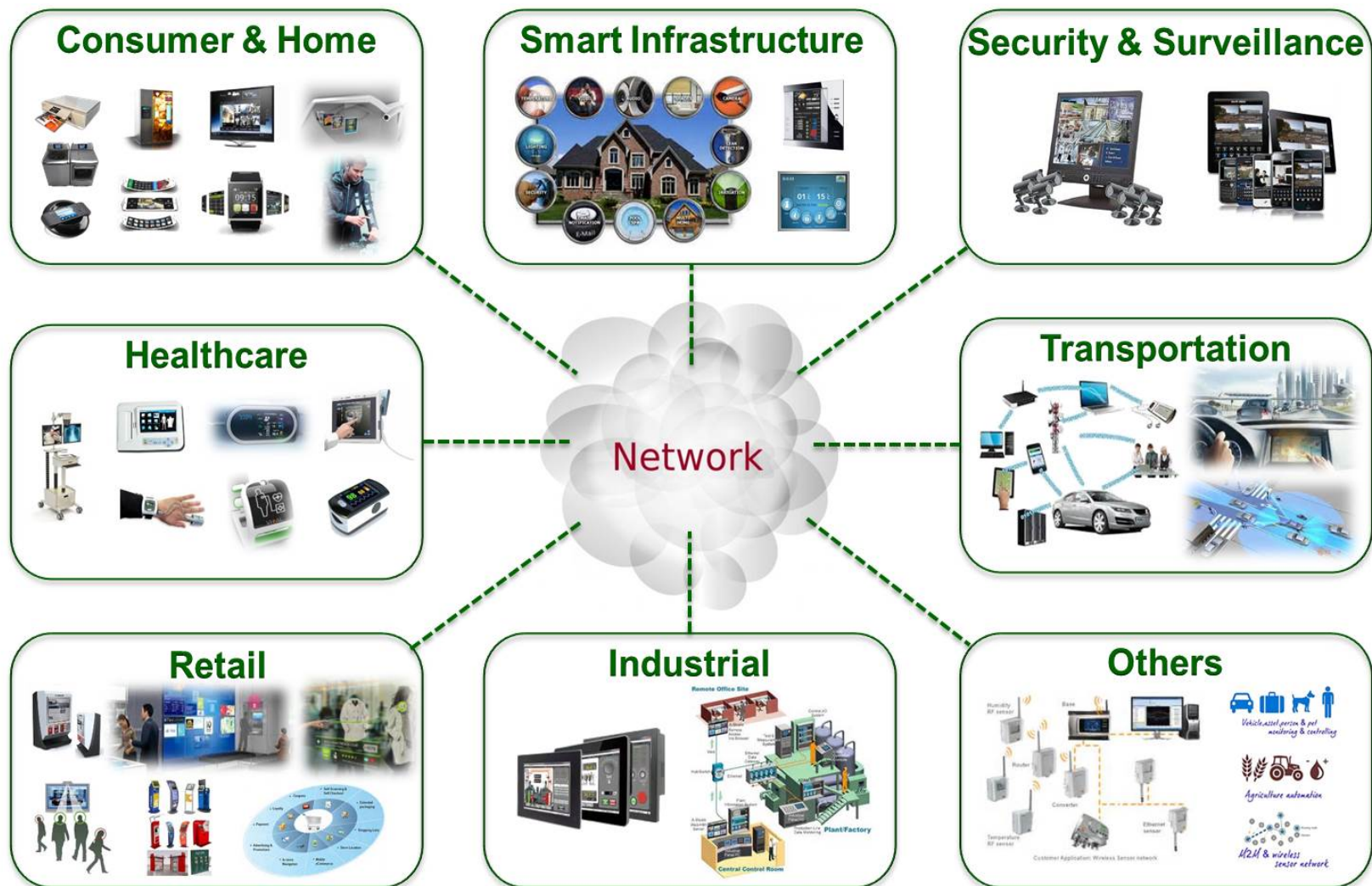


Exemplos

- Web
 - URI (*Uniform Resource Identifier*) rfc2396
 - URL (*Uniform Resource Locator*)
 - URN (*Uniform Resource Name*)
 - Exemplos:
 - urn:xmlorg:objects:dtd:xml:docbook:v4.1.2
 - <https://www.ietf.org/rfc/rfc3120.txt>
 - URL

`http://` `servername` `[:port]` `[/pathName]` `[?query]` `[#fragment]`

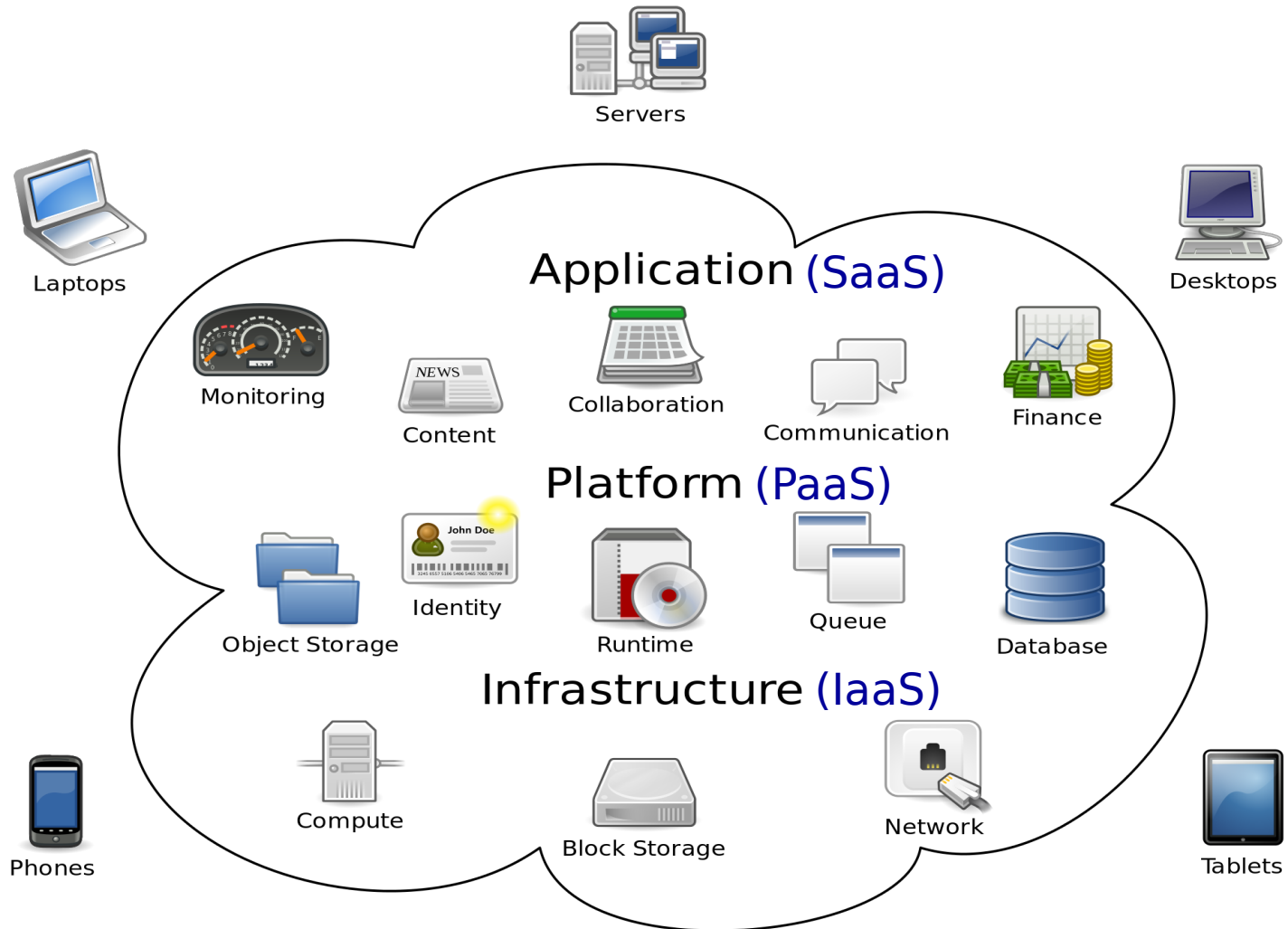
Tendências



Vivante and the Vivante logo are trademarks of Vivante Corporation. All other product, image or service names in this presentation are the property of their respective owners. © 2013 Vivante Corporation

Computação ubíqua e IoT (Internet of Things)

Tendências



Cloud Computing

Características

- Heterogeneidade
 - Redes
 - Hardware
 - Sistemas Operacionais
 - Linguagens de Programação
 - Implementações por diferentes desenvolvedores
- Soluções: Middleware, uso de padrões

Características

- Heterogeneidade
 - Exemplos:
 - Diferentes tecnologias de redes (ATM, GSM, Ethernet)
 - Intel (little-endian) e IBM (big-endian)
 - Big-endian comum em redes e little-endian em microprocessadores
 - Quebra de linha em diferentes SOs: Windows (\n\r) e Linux (\n)
 - Codificação (UTF-8, ISO 8859-15)
 - String em C versus String em Java

big-endian: byte mais significativo no menor endereço.

little-endian: byte menos significativo no menor endereço.

Características

- Abertura
 - Um sistema aberto pode ser estendido e implementado de diferentes formas
 - **Publicação de interfaces**
 - Exemplos: RFC (Request for Comments), CORBA, documentação
 - Desafio: juntar componentes desenvolvidos por diferentes pessoas.

Características

- Segurança
 - Proteger a informação
 - Garantir a disponibilidade
- Desafios: Ataques de DDoS, Roubo de dados,

Características

- Escalabilidade
 - Controlar custo recursos físicos
 - Exemplo: aumento de usuários, gasto de energia.
 - Controlar a perda de desempenho
 - Exemplo: algoritmos com estruturas hierárquica versus lineares
 - Prevenir o esgotamento de recursos
 - Exemplo: IPv4 versus IPv6
 - Evitar gargalos de desempenho
 - Exemplo: descentralização de recursos (DNS)

Características

- Escalabilidade
 - Soluções
 - Replicação de dados
 - Cache
 - Múltiplos servidores
 - ...

Características

- Tratamento de Falhas
 - Hardware e software **falham**.
 - Em SD essas falhas geralmente são falhas parciais.
 - Técnicas:
 - Detecção de falhas (checksum)
 - Mascaramento de falhas (retransmissão)
 - Tolerância a falhas (timeout na Web)
 - Recuperação de falhas (rollback)
 - Redundância (rotas, recursos, servidores)

Características

- Concorrência
 - Acesso a um recurso por vários clientes ao mesmo tempo.
 - Uso de múltiplas threads ou processos
 - Desafio: garantir a operação e consistência do sistema.
 - Algumas soluções: Controle de concorrência, redundância, limitar o acesso.

Características

- Transparência
 - Para o usuário
 - Para o programador
- ANSA e o RM-ODP definem 8 tipos de transparência: acesso, localização, concorrência, replicação, falhas, mobilidade, desempenho, escalabilidade.
- Transparência de rede: acesso + localização

Características

- **Transparência**
 - **Acesso:** recursos remotos e locais são acessados com as mesmas operações;
 - **Localização:** acesso aos recursos sem saber sua localização;
 - **Concorrência:** vários processos acessam os mesmos recursos compartilhados sem interferência;
 - **Replicação:** múltiplas instâncias de recursos são usadas sem conhecimento sobre réplicas;
 - **Falhas:** permite aos usuários completarem suas tarefas independente de falhas em software ou hardware;
 - **Mobilidade:** permite a movimentação de recursos e clientes sem afetar a operação dos usuários;
 - **Desempenho:** permite ao sistema ser reconfigurado para melhorar o desempenho;
 - **Escala:** permite mudanças de escala sem alterar aplicações ou algoritmos.

Questão

[POSCOMP 2015] Um dos objetivos do projeto de um Sistema Distribuído é fornecer transparência, ocultando aspectos distribuídos dos usuários do sistema. Um sistema transparente proporciona um ambiente em que os seus componentes apresentam-se logicamente centralizados, mesmo fisicamente separados. Entre os vários tipos de transparência que os sistemas distribuídos podem fornecer, o ocultamento do fato de que há várias cópias de um recurso disponíveis no sistema é conhecido como:

- (A) transparência de acesso.
- (B) transparência de transação.
- (C) transparência de replicação.
- (D) transparência de concorrência.
- (E) transparência de migração.

Questão

[POSCOMP 2011] Qual dos parâmetros a seguir tem maior impacto sobre o desempenho de algoritmos distribuídos?

- a) O volume total de dados transferidos.
- b) A transparência de dados.
- c) A transparência de execução.
- d) A política de escalonamento de tarefas em cada nó do sistema.
- e) O número de mensagens trocadas.

Referências

COULOURIS, George F; DOLLIMORE, Jean; KINDBERG, Tim; BLAIR, Gordon. **Sistemas distribuídos: conceitos e projeto**. 5. ed. Porto Alegre: Bookman, 2013.