

Comunicação em SD

Sistemas Distribuídos

2024

Professor Vinícius Hax

- Arquiteturas de sistemas distribuídos
 - Centralizadas: alguns nós são mais importantes do que outros
 - Duas camadas (cliente e servidor)
 - Exemplo: DNS, E-mail
 - Três camadas
 - Exemplo: Web (cliente – servidor de aplicação – servidor de banco de dados)
 - P2P (Peer-to-peer): todos os nós são iguais
 - Exemplo: Napster
 - Híbrida: existem categorias de nós
 - Exemplos: BitTorrent, Blockchain
- Sockets

- A comunicação é essencial em sistemas distribuídos. Porém também existem várias formas de fazer.
- A principal maneira de classificar é:
 - Comunicação direta
 - Comunicação indireta
- Na comunicação DIRETA quem manda a mensagem tem que dizer para quem a mensagem se direciona

- ❖ O desacoplamento é a ligação entre quem manda e quem recebe uma mensagem
- ❖ O desacoplamento pode ser espacial ou temporal ou ambos
 - ❖ Espacial: A mensagem é enviada mas não se sabe quem recebe
 - ❖ Temporal: A mensagem é enviada mas não se sabe QUANDO ela será recebida

Comunicação acoplada e desacoplada

		Espacial	
		Sim	Não
Temporal	Sim	Conversa pessoalmente	Pessoa falando em um mega-fone (quem está ouvindo?)
	Não	E-mail / Whatsapp	Pessoa grava um áudio e vídeo e divulga na Internet

● Principais tipos

- Requisição resposta
- Chamada de procedimento remoto
- Invocação de método remoto

■ Principais tipos

■ Requisição-resposta

- Aqui temos primitivas do tipo send e receive que estabelecem comunicação direta entre dois ou mais hosts

- Exemplo: socket TCP/UDP

■ Chamada de procedimento remoto

- São chamadas funções, porém algumas das funções são implementadas em hosts remotos

- Exemplo: RPC (Remote Procedure Call)

■ Invocação de método remoto

- São chamados métodos de objetos porém ou os alguns dos métodos ou os próprios objetos podem estar em hosts remotos

- Exemplo: RMI

Conteúdos

UNIDADE I – Conceitos Fundamentais de Sistemas Distribuídos

- 1.1 Conceitos básicos
- 1.2 Definição
- 1.3 Modelos
- 1.4 Vantagens e desvantagens

UNIDADE II – Comunicação nos Sistemas Distribuídos

- 2.1 Comunicação cliente-servidor
- 2.2 Comunicações em grupo
- 2.3 Chamadas de procedimento remoto

UNIDADE III - Processos e Processadores

- 3.1 Linhas de controle
- 3.2 Alocação de processadores
- 3.3 Modelos de sistema

UNIDADE IV - Sincronização em Sistemas Distribuídos

- 4.1 Sincronização de relógios
- 4.2 Exclusão mútua
- 4.3 Algoritmos eletivos
- 4.4 Transações atômicas
- 4.5 *Deadlock* em sistemas distribuídos

- Dois trabalhos valendo cinco pontos em cada etapa

Conteúdo

- Conceitos de sistemas distribuídos

O que é um sistema?

- ❖ “é um conjunto de elementos interdependentes de modo a formar um todo organizado.”

- 📍 <https://pt.wikipedia.org/wiki/Sistema>

- ❖ “A system is a group of interacting or interrelated elements that act according to a set of rules to form a unified whole.”

- 📍 <https://en.wikipedia.org/wiki/System>

E um sistema **DISTRIBUÍDO**?

“Um sistema distribuído é um conjunto de computadores independentes que se apresenta a seus usuários como um sistema único e coerente”
Tanenbaum & Van Steen

Um mainframe com seus terminais seria um sistema distribuído?

O que é um terminal?

<https://www.youtube.com/watch?v=RuZUPpmXfT0>



Um mainframe com seus terminais seria um sistema distribuído?

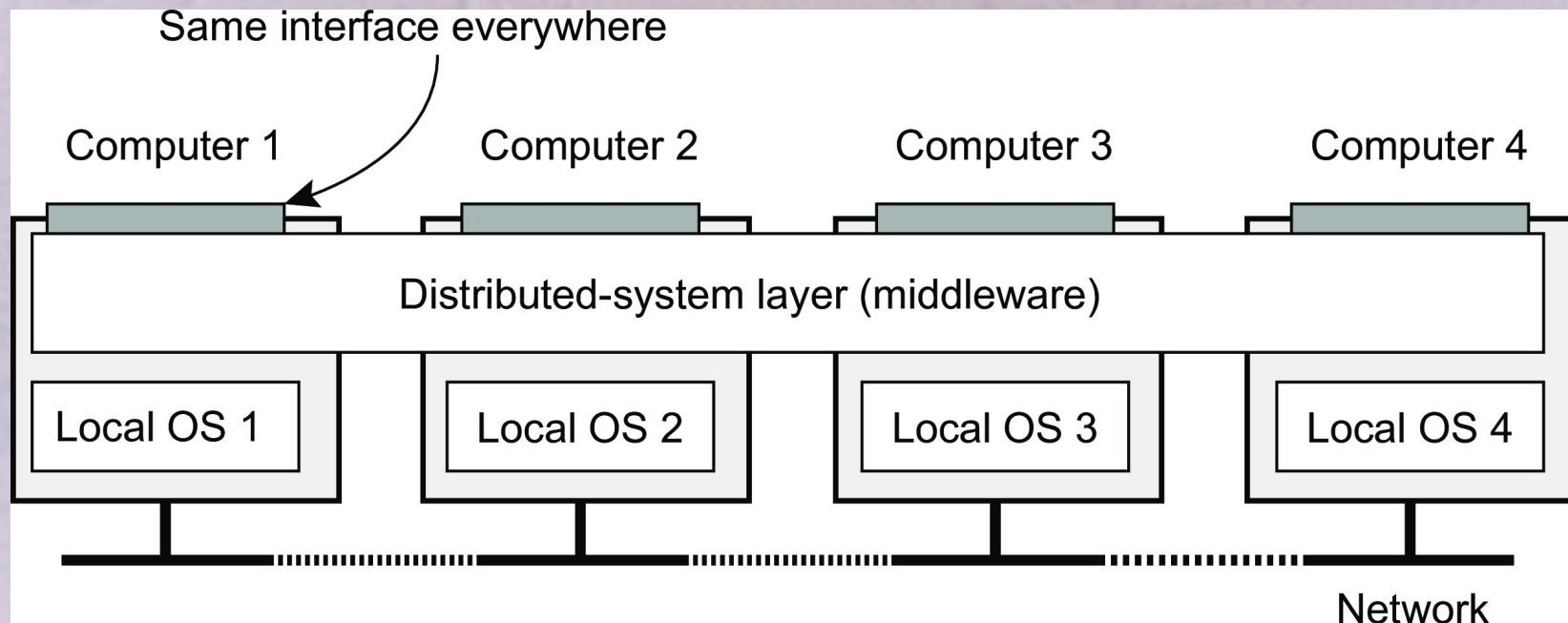
R: Não, pois os terminais não tem função independente

Exemplos de sistemas distribuídos

- Armazenamento de arquivos baseado em nuvem (Google Drive, Dropbox, etc)
- Aplicações de streaming (Netflix, PrimeVideo, YouTube)
- Serviços de e-mail de grandes provedores (Gmail, Outlook, etc)
- Hospedagem web com base em virtualização

Características comuns

- Os detalhes dos elementos e como eles se comunicam estão ocultos dos usuários
- A expansão é relativamente fácil
- Camada de middleware facilita a programação/configuração



Características desejáveis

- Acesso a recursos de maneira controlada e eficiente
- Transparência
 - Acesso: como os dados são armazenados
 - Localização
 - Migração
 - Replicação
 - Concorrência
 - Falha
- Abertura: regras bem definidas, documentadas e passíveis de alteração
- Escalabilidade: o sistema continua funcionando mesmo com aumento nas demandas

Tipos de escalabilidade

- Tamanho: é possível atender mais usuários ou acrescentar recursos ao sistema?
- Geográfica: o que acontece se os usuários estiverem a uma distância maior
- Administrativa: é possível adicionar mais administradores? O sistema suporta hierarquias de permissões?

O problema do relógio global

- Imagine um conjunto de servidores web que devem entregar um mesmo conteúdo.
 - O ideal é que a carga entre eles seja distribuída para evitar sobrecarga em algum nó
 - Um algoritmo simples: contar quantos usuários cada servidor atende de tempos em tempos. Se alguém está acima da média o pedido pode ser repassado para outro servidor
 - Problema: quando os servidores devem comunicar aos outros a sua carga se não existe um relógio global?

Comunicação síncrona x assíncrona

- **Síncrona:** totalmente dependente do tempo → um de cada vez, possivelmente de forma alternada
 - Exemplo: ligação telefônica / vídeo-chamada
 - Vantagem: programação mais simples → procedural
 - Desvantagem: o que fazer com o remetente enquanto o destinatário não responde? E se a resposta nunca vier?
- **Assíncrona:** cada participante responde no seu próprio tempo
 - Exemplo: whatsapp / e-mail
 - Vantagem: cada agente se comunica quando deseja e/ou pode
 - Desvantagem:
 - programação mais complicada
 - O tempo total da comunicação pode ser maior

- Uma maneira de lidar com a escala é a replicação
- Podemos replicar serviços e algoritmos ... mas e os dados?
 - Na maioria dos sistemas as operações de consulta são mais frequentes do que as atualizações (inserções, edições e remoções)
 - Graças a isso podemos usar o mecanismo de cache, ou seja, manter temporariamente uma cópia dos dados em outro lugar
 - Resolve muitos problemas mas ...
 - ... gera outros
 - Como armazenar o cache?
 - Como lidar com a inconsistência?

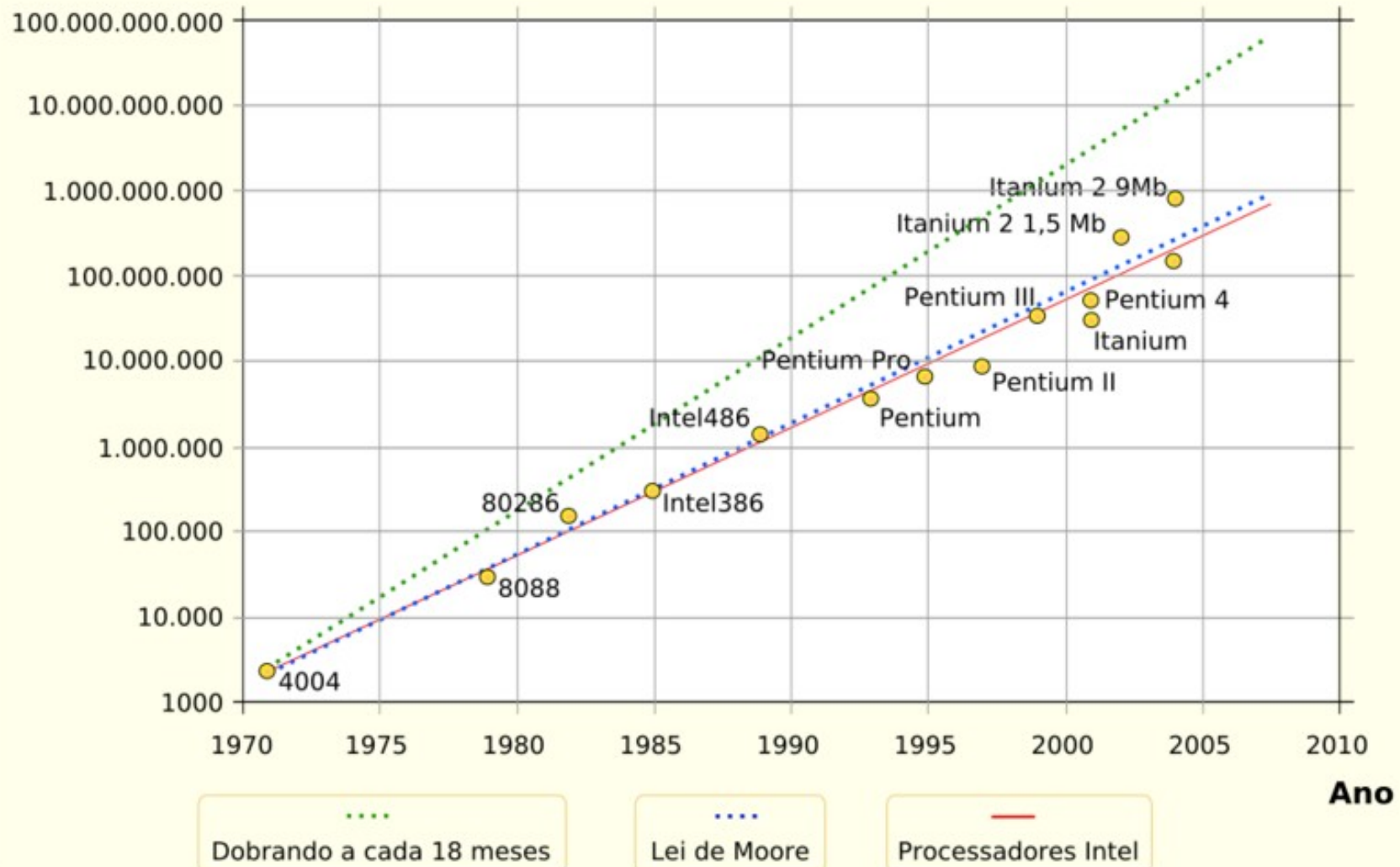
Armadilhas comuns ao desenvolver sistemas distribuídos

- ❖ A rede é confiável
- ❖ A rede é segura
- ❖ A rede é homogênea
- ❖ A topologia não muda
- ❖ A latência é rede
- ❖ A largura de banda é infinita
- ❖ O custo de transporte é zero
- ❖ Só existe um administrador no sistema

Tipos de sistemas distribuídos

Lei de Moore

Número de Transistores



Tipos de sistemas distribuídos

- ❶ Sistemas de computação
- ❷ Sistemas de informação
- ❸ Sistemas pervasivos ou ubíquos

Tipos de sistemas distribuídos (2)

- Sistemas de computação distribuídos: ênfase em algoritmos executados em paralelos
 - Pensados como alternativas aos desafios da Lei de Moore
 - Exemplo típico: modelos de previsão meteorológica
 - Subtipos:
 - Cluster: hardwares semelhantes
 - Computação em grade: hardwares distintos (eventualmente geograficamente separados)
 - Ex: projeto Seti@Home

Tipos de sistemas distribuídos (3)

- Sistemas de informação distribuídos: Sistemas acessados de forma tradicional mas que no backend tem características de sistemas distribuídos
 - Muitas vezes utilizados para atender à demandas empresariais
 - O aspecto distribuído surge principalmente para atender as demandas de escala
 - Exemplo: sistemas web

Tipos de sistemas distribuídos (4)

- Sistemas ubíquos ou pervasivos
 - Ubíquo → Aquilo que está presente em todos os locais, muitas vezes de forma invisível
 - Pervasivo → Aquilo que está sempre disponível
 - Sub-divisões
 - Sistemas domésticos → Sistemas em ambientes domésticos
 - Sistemas para tratamento de saúde → Voltadas para finalidade médica
 - Redes de sensores → Sistemas baseados em dezenas de sensores que medem alguma grandeza e cujos dados agregados possuem alguma utilidade
 - Ex: Como está a humidade no lado norte da fazenda?

- ❖ “Sistemas distribuídos”, Tanenbaum & Van Steen
- ❖ Imagens
 - ❖ Slide 13: Captura própria
 - ❖ Slide 16: Tanenbaum & Van Steen
 - ❖ Slide 23: Wikipédia