

Sistemas Distribuídos

Aula 1 – Introdução e visão geral

DCC/IM/UFRRJ

Marcel William Rocha da Silva

Objetivos da aula

- Introdução e conceitos básicos
 - *Middleware*
- Metas de um sistema distribuído
- Tipos de sistemas distribuídos

Um pouco de história...

- No passado: poucos computadores no mundo
 - *Mainframes* – computadores de grande porte
 - Alta capacidade de processamento (para os padrões da época)
 - Pouca, ou nenhuma, comunicação entre eles
- Depois: aumento da capacidade de processamento, computador pessoal (PC) e redes de comunicação
 - Muitos computadores
 - Muita capacidade de processamento
 - Capacidade de transmissão de dados entre computadores

Resultado

- Inúmeras possibilidades de novas aplicações
 - Sistemas compostos por vários computadores – possivelmente em escala global
 - Trocando grandes volumes de informação – via redes locais e Internet
- **Advento dos sistemas distribuídos!**

O que é um SD?

- Definições clássicas:

“Conjunto de computadores independentes que se apresenta a seus usuários como um sistema único e coerente”

- *Tanenbaum et al.*

“Conjunto de componentes de hardware ou software, localizados em computadores interligados em rede, que se comunicam e coordenam suas ações através do envio de mensagens”

- *Coulouris et al.*

O que é um SD?

- **Componentes**

- **Software** ou **hardware** independentes
- Podem ser heterogêneos
 - Ex.: mainframes, laptops, nós sensores, etc
- Colaboração entre componentes é o cerne de um sistema distribuído!

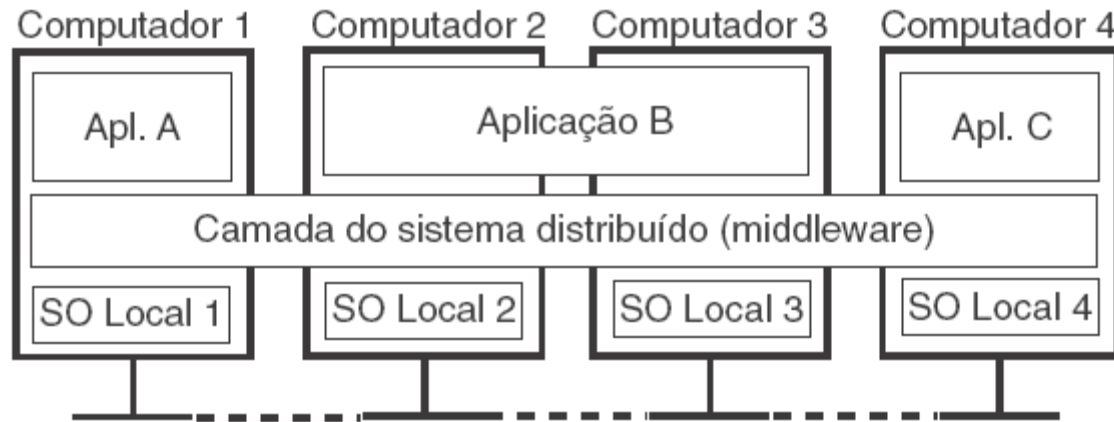
- **Usuários**

- Pessoas ou outros componentes
- Interação com o sistema deve ser consistente e uniforme, independente de onde acontece
- Expansão ou falhas do sistema devem ser transparentes

Como lidar com componentes heterogêneos?

- Utilizar uma camada de software!
- ***Middleware***
 - Camada de software situada entre a camada de nível mais alto (aplicações e usuários) e as camadas de nível mais baixo (SO, detalhes de comunicação e hardware)

Middleware



- Oculta a distribuição
 - Aplicação executada em diferentes locais geograficamente
- Oculta a heterogeneidade
 - Diferentes hardwares, SOs, protocolos de comunicação, etc

Metas

- Características desejáveis em um SD
 1. Acesso
 2. Transparência
 3. Abertura
 4. Escalabilidade

Meta 1: Acesso a recursos

- Facilitar aos usuários e aplicações o acesso a recursos remotos e o seu compartilhamento de maneira eficiente
- Objetivo → **economia**
- Recursos de alto custo podem ser compartilhados e utilizados melhor
 - Impressoras, servidores caros, dados, etc
- Problema → **segurança**
 - Torna-se crítica devido ao compartilhamento

Meta 2: Transparência da distribuição

- Ocultar o fato de que componentes e recursos estão geograficamente distribuídos
- Tipos de transparência
 - Acesso
 - Localização
 - Migração
 - Relocação
 - Replicação
 - Concorrência
 - Falha

Meta 2: Transparência da distribuição

- **Acesso**

- Ocultar diferenças entre a representação dos dados e o modo de acesso aos recursos
- Ex.: inteiros *little endian* e *big endian*, arquivos de texto com codificação diferentes

- **Localização**

- Ocultar a localização física dos recursos e componentes do sistema → Nomeação!
- Ex.: Onde fica **www.facebook.com** ?

Meta 2: Transparência da distribuição

- **Migração**

- Recursos podem ser mudados de lugar sem afetar o modo como podem ser acessados
- Ex.: mudança de um servidor Web

- **Relocação**

- Recursos podem ser mudados de lugar no sistema (relocados) enquanto são acessados
- Ex.: Telefone celular em movimento

Meta 2: Transparência da distribuição

- **Replicação**
 - Ocultar a existência de múltiplas cópias de um mesmo recurso
 - Permite aumentar a disponibilidade e o desempenho
- **Concorrência**
 - Ocultar a existência de múltiplos usuários acessando o mesmo recurso simultaneamente
 - Necessário garantir a consistência
- **Falha**
 - Ocultar falhas em um recurso e a subsequente recuperação do mesmo

Problemas da transparência

- Nem sempre a transparência é um bom negócio
 - Deve ser considerada em conjunto com outras questões, como **desempenho** e **facilidade de uso**
- Exemplo:
 - Distância geográfica entre recursos e componentes
 - Grandes atrasos
 - Falha e recuperação de componentes
 - Garantir coerência de réplicas

Meta 3: Abertura

- Oferecer serviços de acordo com regras padronizadas que descrevem a sintaxe e a semântica desses serviços
 - Ex.: Protocolos em redes de computadores
- Em SDs → **interfaces**
 - Linguagem de descrição de interface (IDL)
 - Funções disponíveis, parâmetros, valores de retorno

Meta 3: Abertura

- O que se ganha com a abertura?
- **Interoperabilidade**
 - Até que ponto duas implementações de componentes diferentes conseguem coexistir e trabalhar em conjunto
- **Portabilidade**
 - Uma aplicação distribuída criada para um sistema distribuído A pode ser usada sem modificação para acessar um sistema B que implemente as mesmas interfaces

Meta 4: Escalabilidade

- Três dimensões:
 - **De tamanho**
 - Facilidade de adicionar usuários e recursos
 - **Geográfica**
 - Facilidade de ter usuários e recursos longe uns dos outros
 - **Administrativa**
 - Facilidade de gerenciar mesmo que abranja múltiplas organizações administrativas

Meta 4: Escalabilidade de tamanho

- **Problema:** Geralmente é a centralização!
 - **Serviços** centralizados → gargalo!
 - Muitos usuários acessando o mesmo servidor
 - **Dados** centralizados → também é gargalo!
 - Muitos usuários e armazenamento limitado
 - **Algoritmos** centralizados
 - Sobrecarga de mensagens enviadas à um único nó
 - Falha de um nó bloqueia o algoritmo

Meta 4: Escalabilidade de tamanho

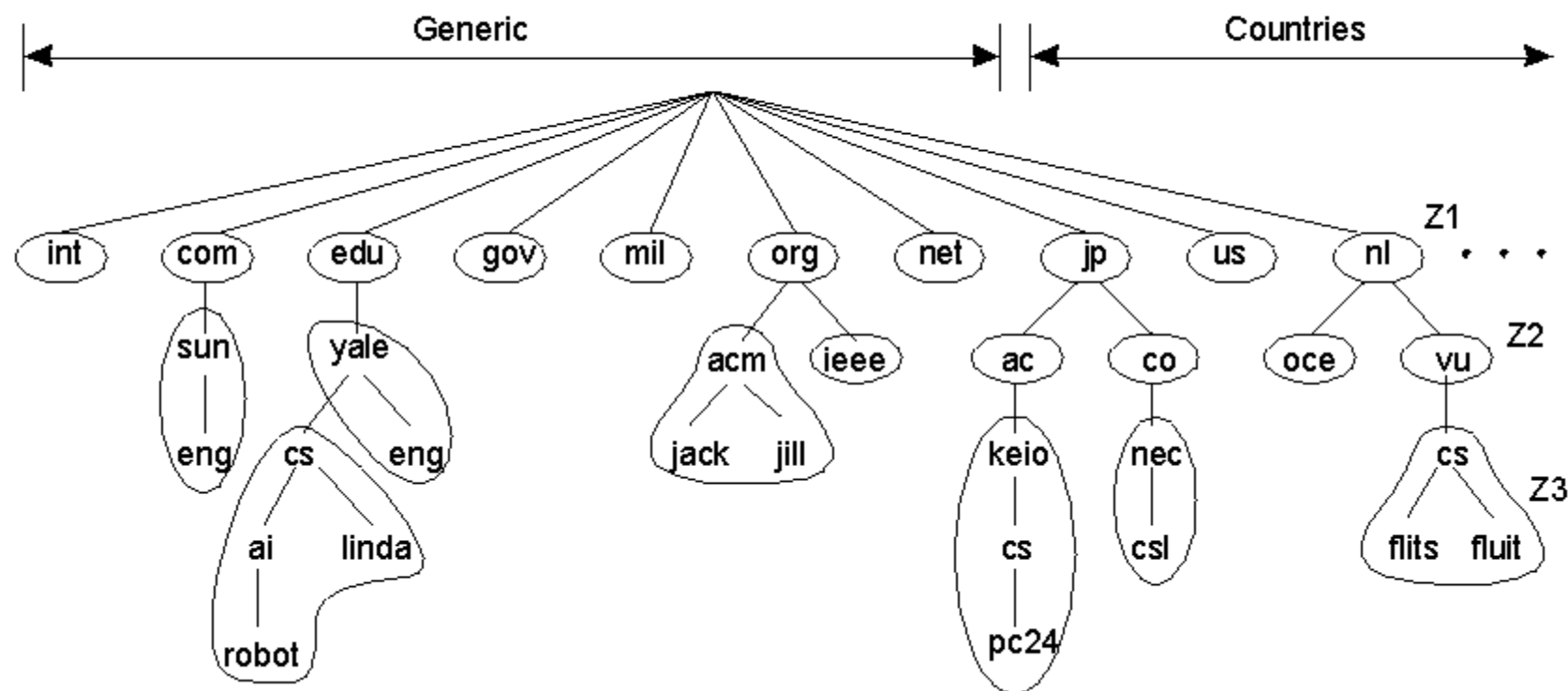
- Soluções

- **Serviços** centralizados → replicação
 - Múltiplas cópias do recurso ou componente
- **Dados** centralizados → distribuição
 - Armazenar a informação de maneira distribuída
- **Algoritmos** centralizados → algoritmos distribuídos
 - Que não dependam de informação ou estados globais

Meta 4: Escalabilidade de tamanho

- Soluções

- Ex.: DNS



Meta 4: Escalabilidade geográfica

- **Problemas** causados pelas longas distâncias
 - Grandes retardos
 - **Comunicação síncrona** é problemática → esperar o resultado de uma solicitação antes de prosseguir
 - Afeta também o desempenho
 - Maior taxa de falhas nas mensagens
 - Impossível comunicação em broadcast (de um para todos)
 - Mecanismos de localização de serviços usados em redes locais não se aplicam

Meta 4: Escalabilidade geográfica

- Soluções

- Grandes retardos

- **Comunicação assíncrona** → executar outras tarefas enquanto espera o resultado da solicitação anterior
 - Replicação de recursos e componentes

- Falhas nas mensagens

- Mecanismos de ACKs e retransmissões

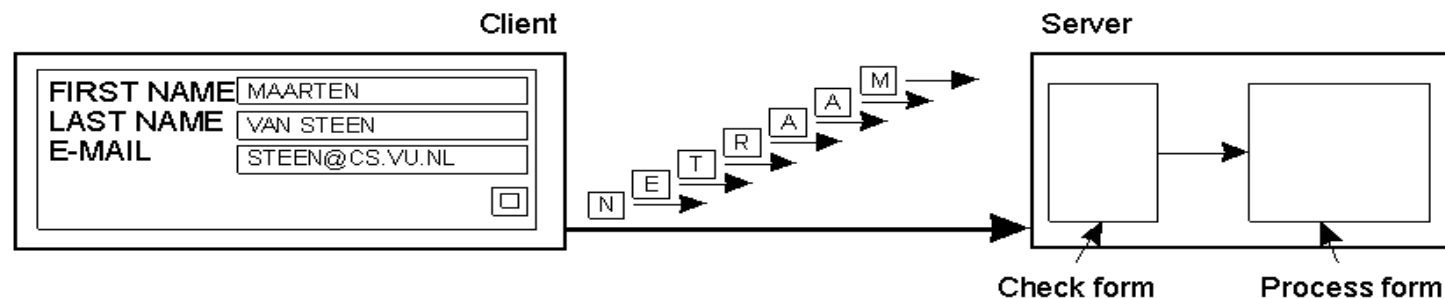
- Impossível comunicação em broadcast

- Sistemas de localização avançados

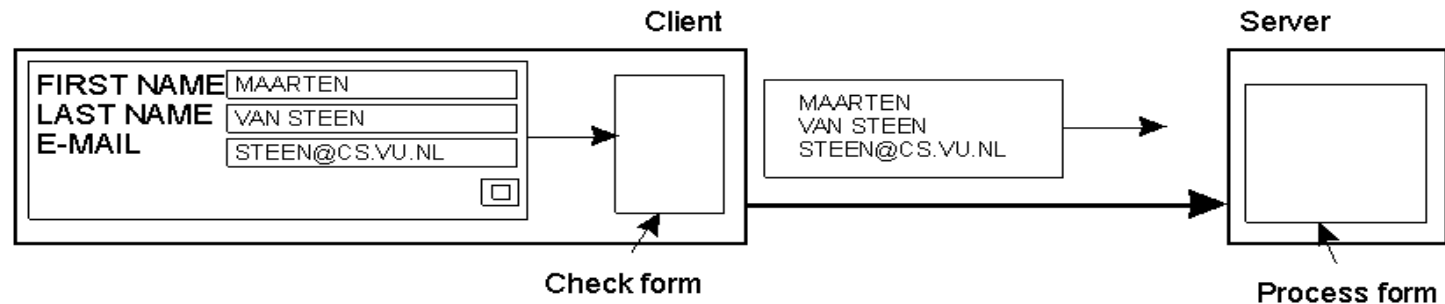
Meta 4: Escalabilidade geográfica

- Soluções

- Ex.: Grandes atrasos em serviços Web



(a)



(b)

Meta 4: Escalabilidade administrativa

- Problemas

- Políticas de uso e tarifação

- Quais recursos estão disponíveis para usuários de outro domínio?

- Gerenciamento

- Quais usuários acessam os recursos

- Segurança

- Um recurso ou usuário é confiável?

Meta 4: Escalabilidade administrativa

- Soluções

- Baseado em políticas definidas pelos administradores
- Difícil uma solução transparente
- Garantir a segurança pode inviabilizar o acesso aos recursos de outro domínio

Ciladas

- Premissas falsas que não podem ser assumidas por um desenvolvedor de sistemas distribuídos
 - A rede é confiável
 - A rede é segura
 - A rede é homogênea
 - A topologia não muda
 - A latência é zero
 - A largura de banda é infinita
 - O custo do transporte é zero
 - Há somente um administrador



Tipos de sistemas distribuídos

- **Sistemas de computação**
- **Sistemas de informação**
- **Sistemas pervasivos (ubíquos)**

Tipos de sistemas distribuídos

- **Sistemas de computação**
 - Tarefas de computação de alto desempenho
 - Unir pequenos computadores para construir um único com alto poder computacional
 - Exemplos:
 - Clusters
 - Grades computacionais

Tipos de sistemas distribuídos

- **Sistemas de informação**

- Integração de diferentes aplicações de rede para a troca de informações e criação de novos serviços
- Diferentes maneiras de comunicar
 - Transações distribuídas básicas (requisição/resposta)
 - RPC (chamada de procedimento remota)
 - RMI (invocação de método remoto)
 - MOM (*middleware* orientado a mensagem) (publicar/subscrever)

Tipos de sistemas distribuídos

- **Sistemas pervasivos** (ubíquos)
 - Componentes móveis, autônomos e com possível escassez de recursos
 - Limitados em energia, banda, armazenamento e tamanho
 - Exemplos:
 - Redes de sensores
 - Sistemas domésticos
 - Sistemas de tratamento de saúde

Resumo

- Sistemas distribuídos
 - Muito usados no nosso dia-a-dia
 - Baseados em um conjunto de diferentes tecnologias
 - Comunicação, rede, programação
 - Conceitos básicos e problemas são importantes para a implementação, gerenciamento e uso de sistemas distribuídos