

Universidade Estadual de Mato
Grosso do Sul

Curso de Ciência da Computação

Disciplina de Algoritmos Paralelos e
Distribuídos

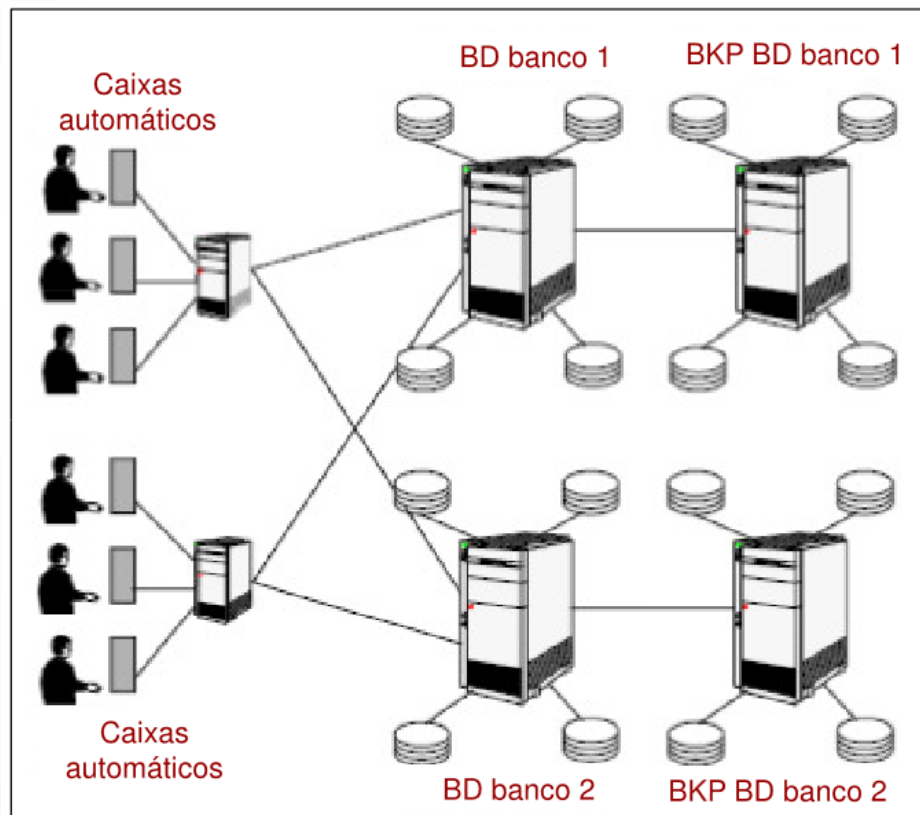
Introdução aos Sistemas Distribuídos

Introdução

- Definição: Um **sistema distribuído** é um conjunto de computadores independentes, interligados por uma rede de conexão executando um software distribuído.
- Exemplos:
 - Aplicações comerciais (reserva de bilhetes, bancos);
 - Aplicações Internet;
 - Aplicações de Acesso a informações multimídia;
 - Grupware (trabalho colaborativo).

Exemplo de Sistemas Distribuídos

- Sistema bancário:



BD = banco de dados
BKP = backup

Aspectos Importantes em um SD Bancário

- Acessos Múltiplos (concorrentes) à mesma conta.
 - Vários usuários acessam a mesma conta ou um sistema do banco (emissão de relatórios) acessa ao mesmo tempo em que usuários realizam transações.
- Transações bancárias podem envolver contas armazenadas em diferentes bancos de dados ou localidades.
- O serviço deve ser transparente ao usuário – não interessa se os dados de sua conta estão armazenados no banco 1 ou no banco 2, a aplicação não deve em nenhum momento deixar transparecer aspectos internos do sistema.

Aspectos Importantes em um SD Bancário

- O serviço deve ser confiável (não deixa objetos em estados inconsistentes) e seguro.
- Bases de dados replicadas devem ser consistentes.
- Bases de dados replicadas aumentam a tolerância a falhas, mas podem tornar o sistema não confiável.
- Tolerância a falhas.



Características

- **Concorrência**

- Em uma rede de computadores, a execução concorrente de programas é a norma;
- Pode-se compartilhar recursos como por exemplo, páginas na web ou arquivos ao se trabalhar em computadores em separado;
- A capacidade do sistema de manipular recursos compartilhados pode ser ampliada pela adição de mais recursos (mais computadores) na rede.

- **Inexistência de Relógio Global**

- Quando os programas precisam cooperar eles coordenam suas ações trocando mensagens .
- A coordenação depende de uma noção compartilhada do tempo em que as ações dos programas ocorrem.
- Não há limite para a precisão com a qual os computadores podem sincronizar seus relógios em uma rede.
- Consequência direta do fato de que a única comunicação se dá por meio do envio de mensagens na rede.

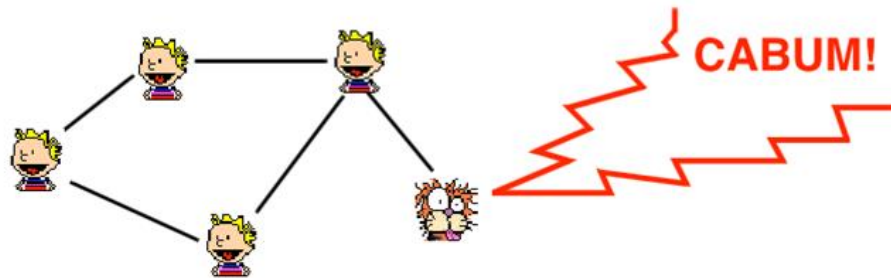
Características

- **Falhas Independentes**

- Todos os sistemas computadorizados podem falhar e é de responsabilidade dos projetistas de sistema pensar nas consequências das prováveis falhas.
- Em sistemas distribuídos, computadores individuais podem ficar isolados por falhas na rede, causando o isolamento desta máquina na rede, mas a máquina em si continua funcionando.
- Cada componente do sistema pode falhar de forma independente, deixando as outras máquinas ainda em funcionamento.

Em Resumo

Sistema Distribuído



Vantagens:

- Maior poder computacional: concorrência
- Menor atraso no acesso, que pode ser feito em um computador “próximo”.
- Separação física possibilita falhas independentes

Observação:

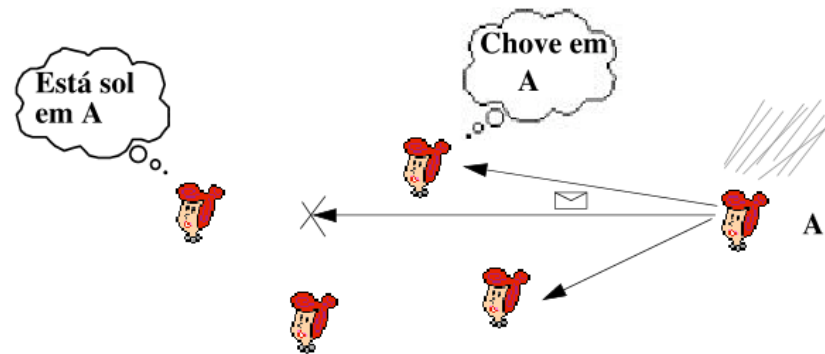
- computação distribuída × computação paralela

a transferência de informação
tem “custo” não desprezível

memória compartilhada

Em Resumo II

Sistema Distribuído



Problema:

A separação física impede que um nó tenha conhecimento do estado global do sistema em um dado instante.

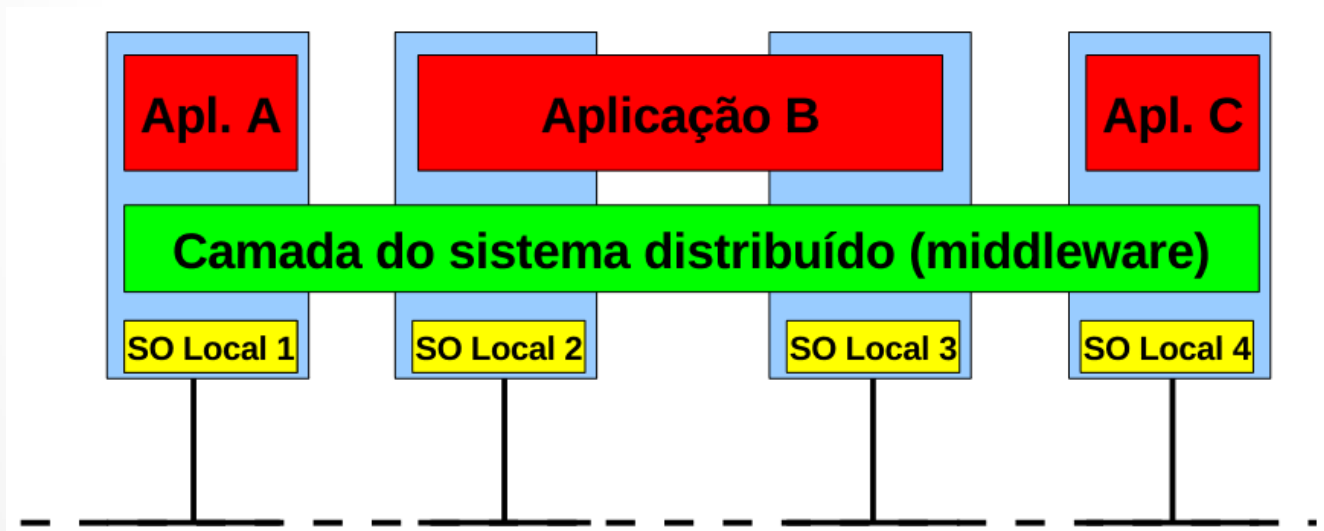
- atrasos indeterminados na comunicação
- falhas

Middleware

- Permite a visão de um sistema único suportando redes e computadores heterogêneos.
- É situada logicamente entre uma camada de nível mais alto, composta de usuários e aplicações, e uma camada subjacente que consistem em sistemas operacionais e facilidades básicas de comunicação.

Middleware

- A camada de middleware pode se estender por várias máquinas, oferecendo a mesma interface a cada aplicação.



Finalidades de um Sistema Distribuído

- Acesso a recursos

- Transparência

- Abertura

- Escalabilidade

-

-

Acesso a Recursos

- Facilitar aos usuários e aplicações acesso a recursos remotos e o compartilhamento de maneira controlada e eficiente.
 - Finalidade: Economia
- Exemplos:
 - Impressoras
 - Computadores
 - Dados
 - Páginas web
- Conectividade
 - Grupware
 - Comércio Eletrônico

Acesso a Recursos

- Problema principal: Segurança
 - Senhas e autenticação de usuários
- Rastreamento de comunicações para montar um perfil de preferências.
 - Violação de privacidade
- Spam

Transparência

- A intenção de um sistema transparente é ocultar o fato de que seus processos e recursos estão fisicamente distribuídos por vários computadores.
- Tipos de transparência:

Transparência	Descrição
Acesso	Oculta diferenças na representação de dados e no modo de acesso a um recurso
Localização	Oculta o lugar em que um recurso está localizado
Migração	Oculta que um recurso pode ser movido para outra localização
Relocação	Oculta que um recurso pode ser movido para uma outra localização enquanto em uso
Replicação	Oculta que um recurso é replicado
Concorrência	Oculta que um recurso pode ser compartilhado por diversos usuários concorrentes
Falha	Oculta a falha e a recuperação de um recurso

Transparência Acesso

- Ocultar diferenças entre arquiteturas de máquinas.
- Identificar como os dados devem ser representados.
 - Exemplo: Como nomear arquivos em SOs diferentes.

Transparência Localização

- Impossibilidade de se identificar a localização física do recurso.
- www.algumacoisa.com.br/~aula
 - Não permite identificar fisicamente onde está localizado o servidor que hospeda esse objeto

Transparência Migração

- Recursos podem migrar de uma localidade para outra, por questões de desempenho, segurança entre outros.
- Deve ser feita de forma automática pelo sistema.
- Deve manter o nome do objeto.
- Deve garantir a continuidade da comunicação.

Transparência Relocação

- Oculta que um recurso possa ser movido para outra localização durante o uso.
- Exemplos:
 - Celular se movimentando dentro de uma mesma área de cobertura.
- Um automóvel passando por várias redes de acesso sem fio, com conexão ininterrupta.

Transparência Replicação

- Permite que várias instâncias de recursos sejam usadas para aumentar a confiabilidade e o desempenho .
- Deve mascarar o conhecimento das réplicas por parte dos usuários.
- Implica na transparência de localização.
- Problemas de consistência.

Transparência Concorrência

- Compartilhamento competitivo de recursos
- Deve garantir consistência
- Travas de acesso
- Tratamento mais refinado: transações.

Transparência Falhas

- Usuário não pode perceber que um recurso deixou de funcionar bem.
- Mascaram falhas é uma das questões mais difíceis.
- Recurso morto ou muito lento?

Transparência

- Compromisso entre um alto grau de transparência e o desempenho do sistema.
- Exemplo: Aplicações de Internet tentam contatar um servidor repetidas vezes antes de desistir.
 - Dúvida: Desistir mais cedo ou permitir que o usuário cancele as tentativas.
- Sistemas embutidos: notebook e impressora local.

Abertura

- Característica que determina se um sistema pode ser estendido de diferentes maneiras:
 - Hardware: Inclusão de dispositivos de fabricantes diferentes.
 - Software:
 - Módulos de SO
 - Protocolos de Comunicação
 - Recursos compartilhados

Abertura

- Interoperabilidade
 - Até que ponto componentes de diferentes fabricantes devem coexistir e trabalhar em conjunto, baseados na confiança mútua de cada um.
 - Capacidade de uma sistema (informatizado ou não) de se comunicar de forma transparente com outro sistema.
- Portabilidade
 - Até que ponto uma aplicação desenvolvida para um sistema distribuído A pode ser executada, **sem modificação**, em um sistema distribuído B, que utilize as mesmas interfaces de A.
 - Habilidade de reusar um código existente ao invés de refazê-lo quando este é movido de um ambiente para outro. → **PADRONIZAR**

Escalabilidade

- Pode ser de três dimensões
 - Tamanho
 - Termos geográficos
 - Termos administrativos

Escalabilidade - Tamanho

- Aumento do número de usuários e/ou processos.
- Algumas limitações:

Conceito	Exemplo
Serviços centralizados	Um único servidor para todos os usuários
Dados centralizados	Uma única lista telefônica on-line
Algoritmos centralizados	Fazer roteamento com base em informações completas

Escalabilidade - Tamanho

- Serviços centralizados:
 - Serviços que são implementados por meio de apenas um único servidor que executa em uma máquina específica no sistema distribuído.
 - Possível gargalo no sistema.
- Dados centralizados:
 - Número telefônicos em um único bando de dados.
 - Saturação de todas as linhas de comunicação que o acessam.
- Algoritmos centralizados:
 - Broadcast de informações

Escalabilidade Geográfica

- Usuários ou recursos podem estar longe um dos outros.
- Problemas:
 - Dificuldade de ampliar sistemas distribuídos existentes que foram originalmente projetados para redes locais.
 - Comunicação síncrona.
 - A comunicação em redes de longa distância é inerentemente não confiável.



Escalabilidade Administrativa

- O sistema pode ser fácil de se gerenciar, mesmo que sua abrangência atinja muitas organizações diferentes.
- Problemas:
 - Políticas conflitantes em relação à utilização de recursos entre outros.
- Os usuários de um único domínio podem confiar em componentes de um sistema distribuído que residam dentro desse mesmo domínio.
- Confiança não deve ultrapassar as fronteiras do domínio.
 - Aplicações devem ser testadas para que não sofram nenhuma ação indevida.

Técnicas de Escalabilidade

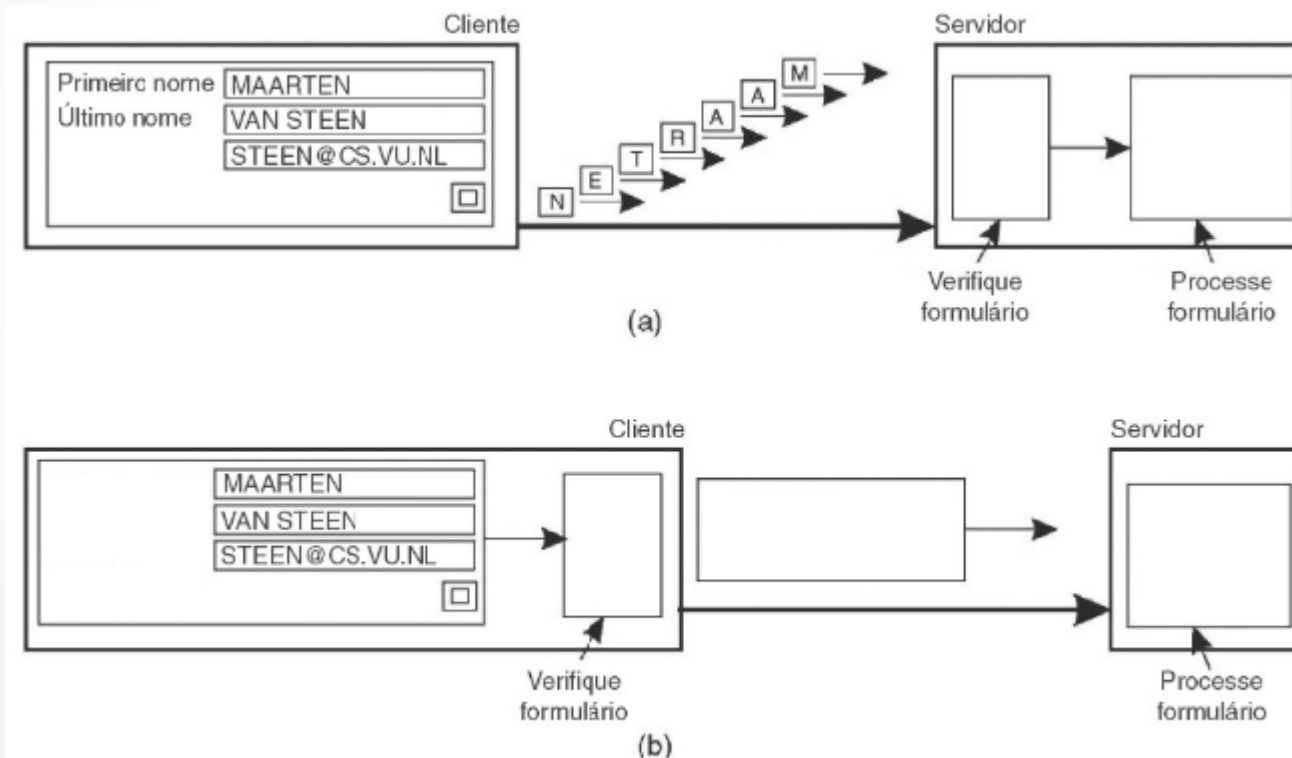
- São 3 as técnicas de escalabilidade:
 - Ocultar latências de comunicação;
 - Distribuição;
 - Replicação.

Técnicas de Escalabilidade – Ocultar latências

- Escalabilidade geográfica;
- Evitar esperar por respostas a requisições remotas;
- Comunicação assíncrona;
- Aplicações interativas devem esperar por uma resposta;
 - Possível ajuste: Reduzir comunicação global, passando parte da computação do servidor para o cliente que está requerendo o serviço.

Técnicas de Escalabilidade – Ocultar latências

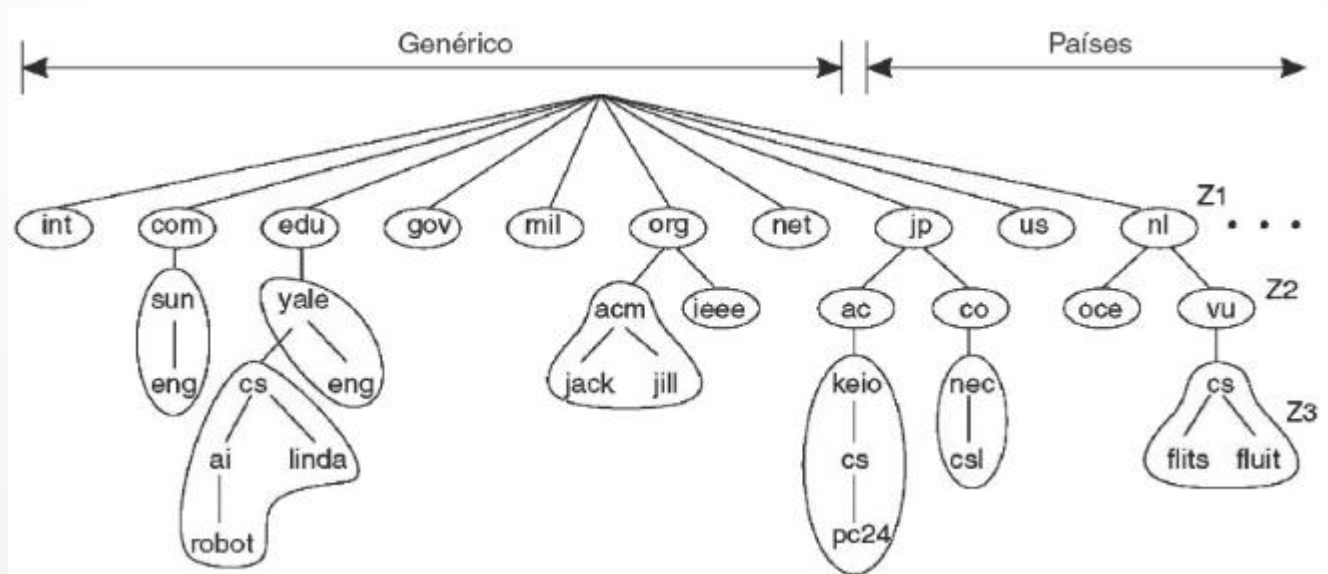
- Exemplo de acesso a banco de dados por meio de formulários.



1.2 A diferença entre deixar (a) um servidor ou (b) um cliente verificar formulários à medida que são preenchidos.

Técnicas de Escalabilidade – Distribuição

- Escalabilidade de tamanho.
- Dividir um componente em partes menores e espalhar as subpartes pelo sistema.
- Exemplo: DNS – hierarquia em árvores de domínio.



Técnicas de Escalabilidade - Replicação

- Aumenta a disponibilidade dos recursos.
- Equilibra a carga entre os componentes.
- Sistema com ampla dispersão geográfica
 - Ocultar os problemas de latência
- Cache
 - Forma especial de replicação
 - Cache é uma decisão do cliente do sistema.

Erros Comuns

- Premissas falsas adotadas ao se desenvolver “pela primeira vez” um sistema distribuído:
 - A rede é confiável;
 - A rede é segura;
 - A rede é homogênea;
 - A topologia é constante;
 - Latência zero;
 - A largura de banda é infinita;
 - O custo de transporte é zero;
 - Existe somente um administrador.

Tipos de Sistemas Distribuídos

1. Sistemas de Computação
2. Sistemas de Informação
3. Sistemas Pervasivos

Sistemas de Computação

- Computação em Cluster.
- Computação em Grade.

Sistemas de Informação

- Sistemas empresariais desenvolvidos para integrar diversas aplicações individuais, onde a interoperabilidade se mostrou “dolorosa”.
 - Sistemas de Processamento de Transações.
 - Integração de Aplicações Empresariais.

Sistemas de Informação

Processamento de Transações

- Requer primitivas especiais que devem ser fornecidas pelo sistema distribuído ou pelo sistema de linguagem.
- Exemplo de primitivas para transações.

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

Sistemas de Informação

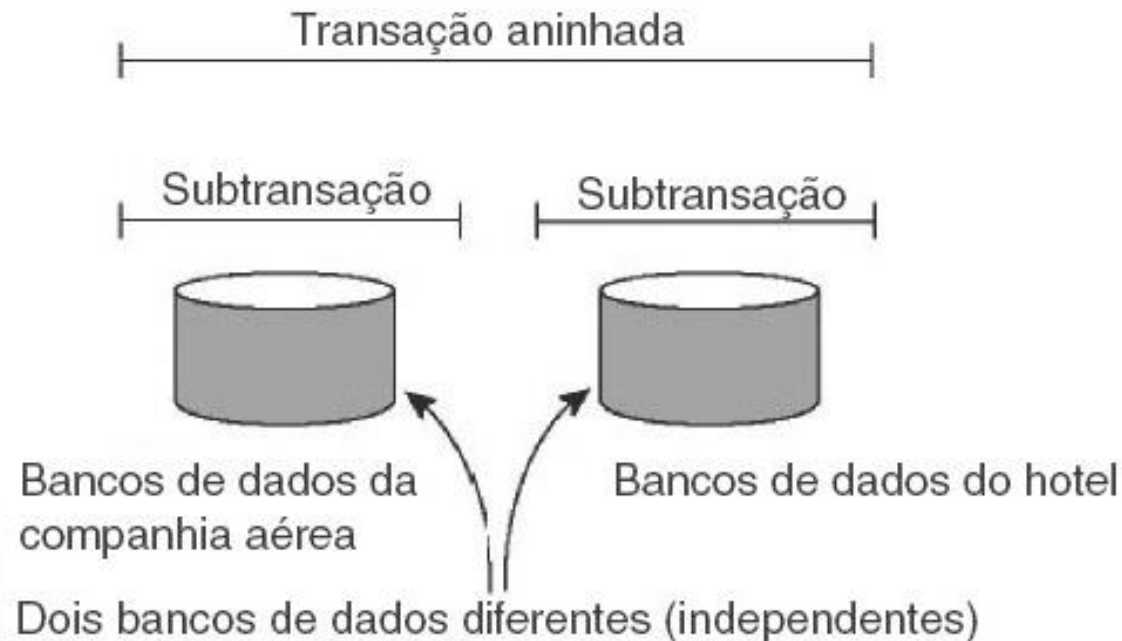
Processamento de Transações

- Características:
 - Atômicas: é indivisível para o mundo exterior.
 - Consistentes: não viola invariantes do sistema.
 - Isoladas: transações concorrentes não interferem umas com as outras.
 - Duráveis: uma vez comprometida uma transação, as alterações são permanentes.

Sistemas de Informação

Processamento de Transações

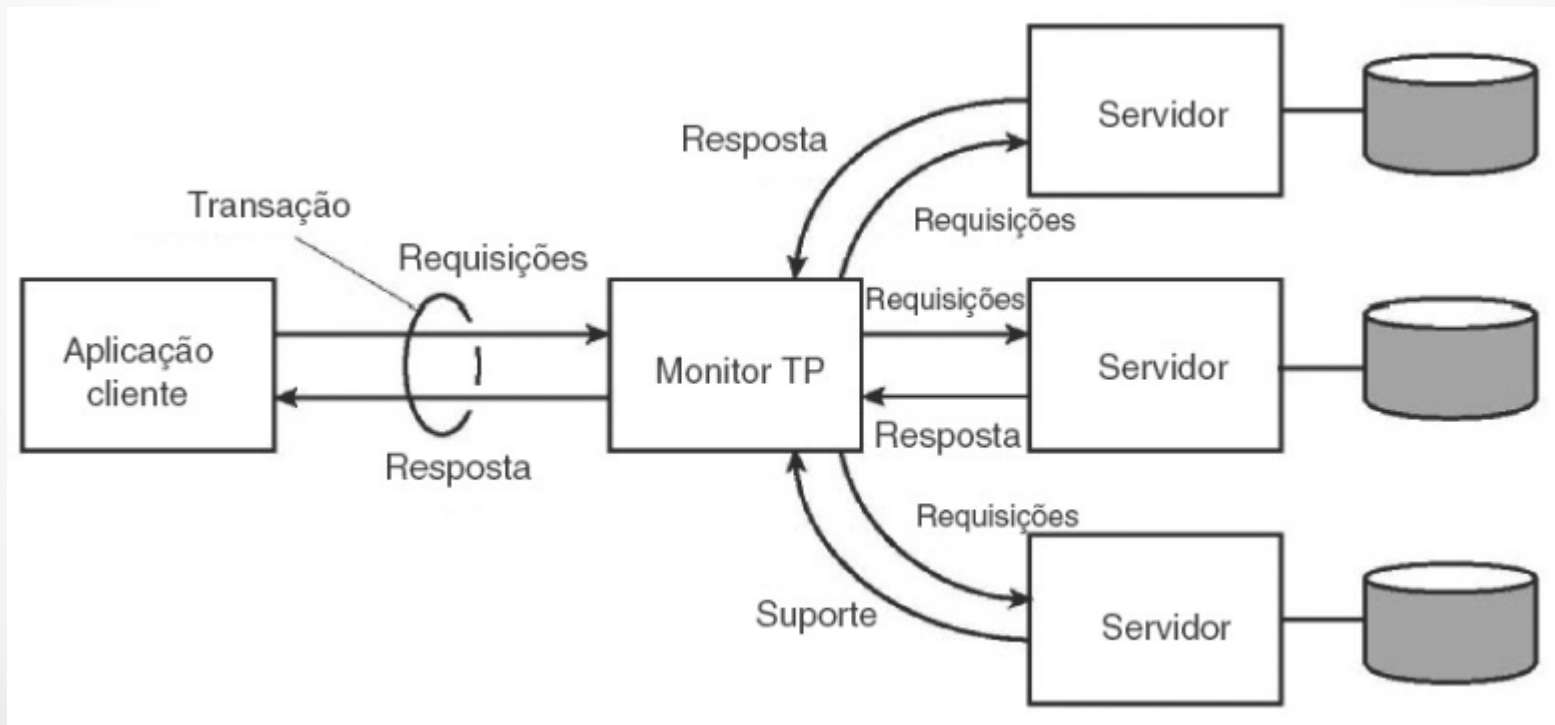
- Transação aninhada: A transação é construída com base em uma quantidade de subtransações.



Sistemas de Informação

Processamento de Transações

- **Monitor de Processamento de Transação:** permite que uma aplicação acesse vários servidores/banco de dados.

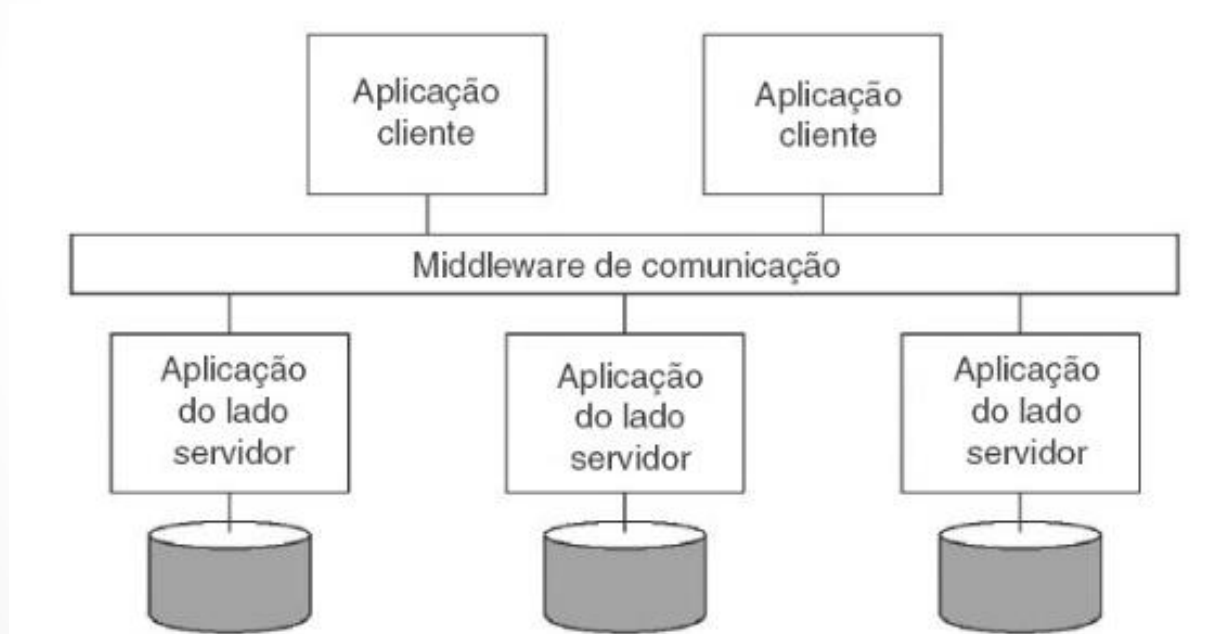


Sistema de Informação – Integração de Aplicações Empresariais

- Aplicações que requerem muito mais em termos de comunicação, e não somente modelo de requisição/resposta.
- Middleware de comunicação:
 - Chamada de Procedimento Remoto.
 - Invocações de Método Remoto.
 - Middleware Orientado a mensagem.

Sistema de Informação – Integração de Aplicações Empresariais

- Middleware como facilitador de comunicação em integração de aplicações empresariais.



Sistema de Informação

Middleware de Comunicação

- Chamada de Procedimento Remoto (RPC)
- Um componente de aplicação pode enviar a um outro componente de aplicação
- Requisição e resposta são empacotadas em mensagens
- Invocação de Método Remoto (RMI)
 - O RMI é semelhante ao RPC, a diferença é que ele (RMI) funciona com objetos ao invés de usar aplicações.

Sistemas de Informação

Middleware de Comunicação

- Desvantagens do RPC e doRMI
 - Os componentes da comunicação devem estar ligados e em funcionamento.
 - Precisam saber exatamente como se referir um ao outro.
- Middleware Orientado a Mensagem (MOM)
 - Aplicações enviam mensagens a pontos lógicos de contato.
 - O middleware se encarrega de entregar todas as mensagens destinadas a uma aplicação.

Sistemas Pervasivos

- O comportamento mais esperados desses sistemas é o **instável**.
- Sistemas pervasivos são dispositivos de computação móvel e embarcados.
 - São pequenos
 - Alimentados por bateria
 - Mobilidade
 - Conexão sem fio
- Requisitos para aplicações Pervasivas:
 - Adotar mudanças contextuais
 - Incentivar composição ad-hoc
 - O compartilhamento é padrão

Tipos de Sistemas Pervasivos

- Sistemas domésticos
- Sistemas eletrônicos para tratamento de saúde
- Redes de sensores

Exercícios e Bibliografia

- Este material tem sua fonte no livro:
 - Sistemas Distribuídos: Princípios e Paradigmas.
 - Andrew Tanenbaum e Marteen Steen
- Exercícios:
 1. Fazer a leitura do capítulo 1 (introdução) do livro acima citado.
 2. Fazer todos os exercícios do capítulo citado no item 1, com exceção dos exercícios referentes a **tarefas de laboratório**.