

Cap. 01 – Introdução

1.1 – Definição de Sistemas Distribuídos

1.2 – Metas

1.2.1 Disponibilização de Recursos

1.2.2 Transparência de Distribuição

1.2.3 Sistema Distribuído Aberto

1.2.4 Escalabilidade

1.2.5 Pitfalls

1.3 – Tipos de Sistemas Distribuídos

1.3.1 Sistemas de Computação Distribuídos

1.3.2 Sistemas de Informação Distribuídos

1.3.3 Sistemas Distribuídos Pervasivo

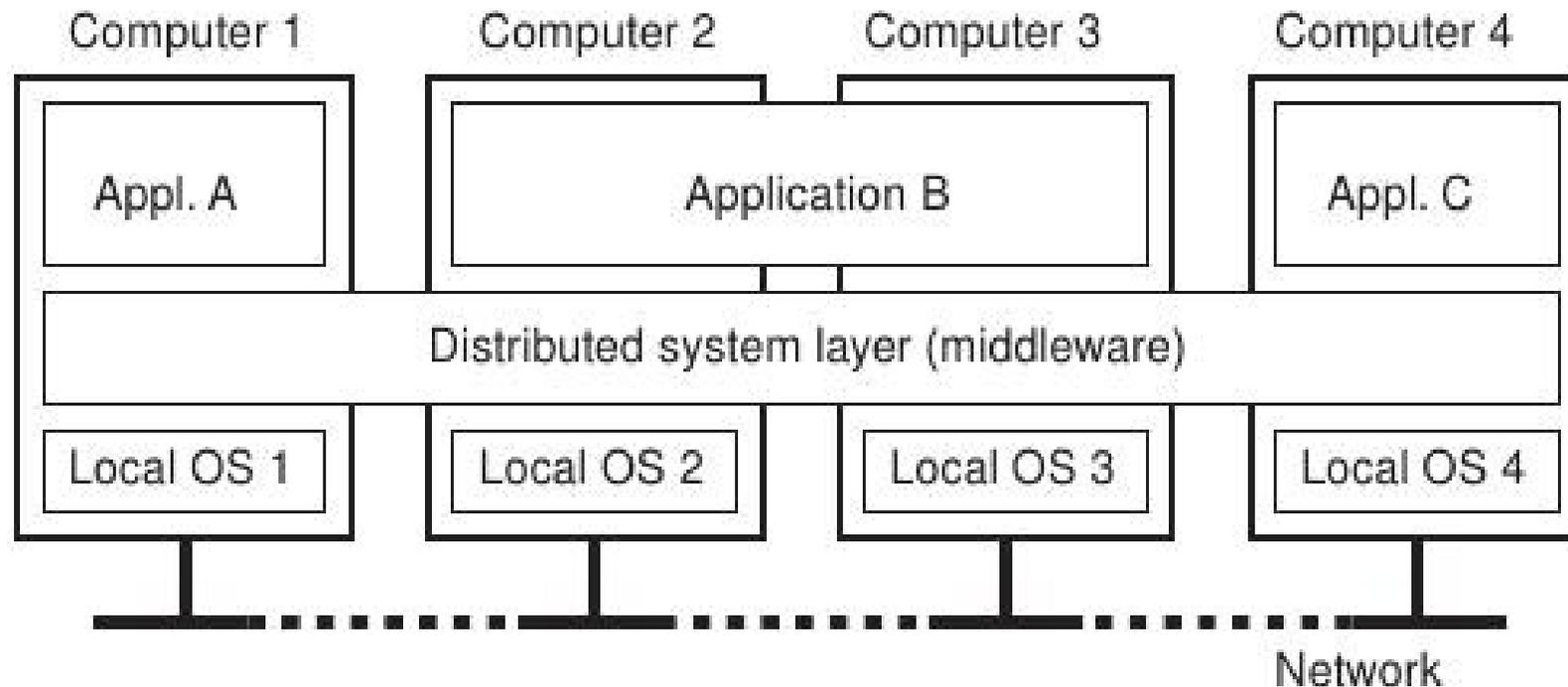
Referências Bibliográficas

- Andrew S. Tanenbaum; Maarten van Steen - Distributed Systems: Principles and Paradigms, Prentice-Hall, 2007, ISBN-10: 0132392275, ISBN-13: 9780132392273
... Lectures dos autores Andrew S. Tanenbaum e Maarten van Steen (“www.cs.vu.nl” e “www.distributed-systems.net/”)
- George Coulouris; Jean Dollimore; Tim Kindberg – Sistemas Distribuídos: Conceitos e Projeto, Bookman, 4th Edition, 2007, ISBN 9788560031498
- Notas de Aula do Prof. Ricardo Anido do Instituto de Computação (IC) da UNICAMP - www.ic.unicamp.br/~ranido/

Introdução – 1.1 Definição de Sistema Distribuído

1.1 – Definição de Sistema Distribuído

- sistema distribuído – software que garante aos seus usuários enxergarem um conjunto de computadores com um sistema único e consistente.
 - 02 aspectos: computadores independentes e sistema único



... 1.1 – Definição de Sistema Distribuído

- Para suportar diferentes sistemas computacionais bem como diferentes redes enquanto oferece uma imagem única do sistema, sistemas distribuídos são geralmente organizados como uma camada de software disposta entre as aplicações distribuídas e o sistema operacional de rede.
- ... um sistema distribuído provê o meio para componentes de uma aplicação distribuída simples comunicar-se umas com as outras, mas também permitir a comunicação de diferentes aplicações.
- ... ao mesmo tempo o sistema distribuído esconde, da melhor forma possível, as diferenças de “hardware” e de sistema operacional de cada aplicação.

1.2.1 – Disponibilização de Recursos

- **Disponibilização de Recursos** – facilitar o acesso de recursos remotos pelos usuários, bem como compartilhá-lo de modo controlado e eficiente
- ... conectar usuários a recursos facilita a colaboração e a troca de informação, o que pode ser exemplificado pelo sucesso da Web.
- ... entretanto, a medida que a conectividade e compartilhamento aumentam, a segurança se torna mais importante – de modo geral sistemas oferecem pouca proteção contra escutas e intrusões.

1.2.2 – Transparência de Distribuição

- **Distribuição Transparente** – esconder o fato de processos e recursos do sistema distribuído se encontrarem fisicamente espalhados (distribuídos) em múltiplos computadores.
- ... um sistema distribuído é transparente se é capaz de se apresentar aos seus usuários e aplicações como um sistema único.
- característica de um sistema distribuído – transparência de forma otimizada do compartilhamento e gerenciamento de recursos físico e/ou lógicos.
- ... transparência – dividida em várias categorias: acesso; localização; migração; replicação; concorrência; falhas.

... 1.2.2 – Transparência de Distribuição

Transp.	Description
Access	Hides differences in data representation and invocation mechanisms
Location	Hides where an object resides
Migration	Hides from an object the ability of a system to change that object's location
Relocation	Hides from a client the ability of a system to change the location of an object to which the client is bound
Replication	Hides the fact that an object or its state may be replicated and that replicas reside at different locations
Concurrency	Hides the coordination of activities between objects to achieve consistency at a higher level
Failure	Hides failure and possible recovery of objects

... 1.2.2 – Transparência de Distribuição

- grau de transparência – alcançar transparência de distribuição de forma completa é exigir muito, pois:
 - usuários podem se encontrar em diferentes lugares;
 - esconder falhas de rede e nós é impossível na teoria e na prática;
 - não é possível distinguir um computador com falha de um lento;
 - como saber se um servidor completou uma tarefa antes da pane;
 - transparência completa diminui a performance:
 - manutenção da cache atualizada como servidor;
 - descarga de operações de escrita para o disco como forma de aumentar a tolerância a falhas

1.2.3 – Sistema Distribuído Aberto

- Sist. Distribuído Aberto – deve ser capaz de interagir com serviços de outros sistemas abertos a despeito da infraestrutura.
 - conformidade com interfaces bem definidas;
 - suporte a portabilidade de aplicações;
 - facilidade de interoperabilidade.
- ... em sistemas distribuídos, serviços são geralmente especificados através de interfaces, que são frequentemente descritas em IDL – Interface Definition Language.
- ... se especificada apropriadamente, a definição de uma interface permite que um processo arbitrário que necessite de desta interface se comunicar com um processo que provê esta interface.

... 1.2.3 – Sistema Distribuído Aberto

- Para se tornar Aberto, um Sistema Distribuído deve tornar-se independente da heterogeneidade do ambiente subjacente, ou seja, do “hardware”, plataforma e linguagens.
- **Interoperabilidade** – caracteriza a extensão na qual duas implementações de sistema ou de componentes de diferentes fabricantes podem trabalhar conjuntamente simplesmente confiando nos serviços uma da outra;
- ... naturalmente que estes serviços devem ser especificados sob uma padrão comum acordado entre as partes.
- **Portabilidade** – caracteriza em que medida um aplicação desenvolvida para um Sist. Distribuído A pode ser executada sem modificação em um Sist. Distribuído B que implementa a mesma interface do Sist. Distribuído A.

1.2.4 - Escalabilidade

- **Observação:** Muitos desenvolvedores de sistemas distribuídos modernos usam o adjetivo “scalable” sem especificar claramente o por que seus sistemas escalam.
- Escalabilidade de um Sistema pode ser medida em 03 dimensões:
 - nro de usuários e/ou processos (escalabilidade de tamanho);
 - distância máxima entre nós (escalabilidade geográfica);
 - nro de domínios possíveis (escalabilidade administrativa).
- **Observação.:** maioria dos sistemas leva, e até um certo ponto, a escalabilidade de tamanho. Hoje em dia, os desafios estão presentes na escalabilidade geográfica e administrativa.

Introdução – 1.2 Metas

... 1.2.4 - Escalabilidade

- Quando um sistema precisa ser escalado, nos deparamos com uma série de problemas a serem resolvidos.
- ... se mais usuários e recursos devem ser suportados, o confronto se dará pelas limitações por conta dos serviços, dados ou algoritmos centralizados.
- Alguns exemplos de Limitações de Escalabilidade:
 - Serviços Centralizados – servidor único para todos os usuários;
 - Dados Centralizados – um banco de dados único para telefones;
 - Algoritmos Centralizados – roteamento baseado em informações globais.

Introdução – 1.2 Metas

... 1.2.4 - Escalabilidade

- Escalabilidade Geográfica - contempla seus próprios problemas.
- ... uma das principais razões da dificuldade em escalar sistemas distribuídos que foram projetados para redes locais é o fato de serem baseados em comunicação síncrona.
- Então como escalar geograficamente o sistema? ... ao dispor o sistema para além das redes locais, variáveis como banda da rede e latência na comunicação podem comprometer a performance.
- ... o desenvolvimento de aplicações interativas utilizando comunicação síncrona em redes metropolitanas requer cuidados e não simplesmente paciência.
- ... outro aspecto que deve ser considerado é que a comunicação em redes metropolitanas é inerentemente não confiável e virtualmente ponto a ponto.

Introdução – 1.2 Metas
... 1.2.4 - Escalabilidade

- Escalabilidade Geográfica está fortemente relacionada a problemas oriundos das soluções centralizadas que impedem a “Size Scalability”.
- ... se temos um sistema com muitos componentes centralizados, a escalabilidade geográfica estará comprometida em razão dos problemas de confiabilidade e performance decorrentes da comunicação em “Wide Area Networks”.
-

Introdução – 1.2 Metas
... 1.2.4 - Escalabilidade

- Escalabilidade Administrativa – o principal problema a ser resolvido é o de políticas conflitantes relacionadas ao uso, gerenciamento e segurança dos recursos.
- ... em essência, usuários confiam em seus administradores de sistema, no entanto, esta confiança não se expande naturalmente sobre diferentes domínios administrativos.
- Na expansão de um Sist. Distribuído em outro domínio administrativo, duas medidas de segurança são necessárias:
 - ambos os sistemas (distribuídos e domínio) necessitam se proteger de ataques maliciosos um do outro.

Introdução – 1.2 Metas

Técnicas para Escalabilidade

- Na maioria dos casos, problemas de escalabilidade em sistemas distribuídos aparecem como problemas de performance decorrentes da limitação na capacidade dos servidores e da rede.
- Há basicamente 03 técnicas: esconder latências de comunicação; distribuição dos dados e da computação; técnica de replicação e armazenamento local.
- Esconder Latências de Comunicação – evitar o uso de “request/reponse” e avançar no ciclo de vida (trabalho)
 - utilização da comunicação assíncrona;
 - oferecer suporte para que respostas que cheguem sejam devidamente tratadas por rotinas de tratamento de requisições;

Introdução – 1.2 Metas

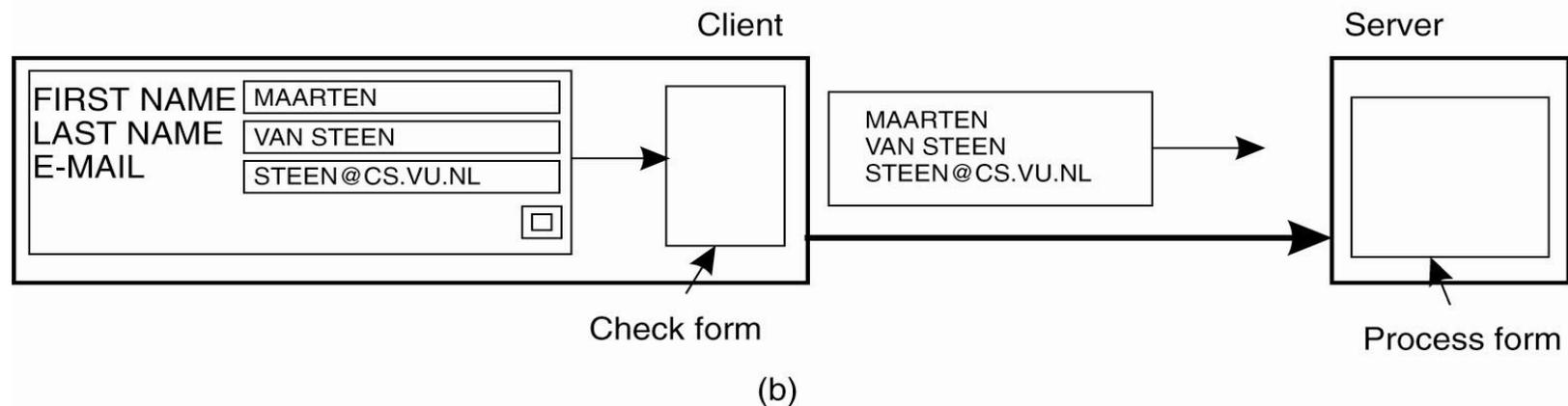
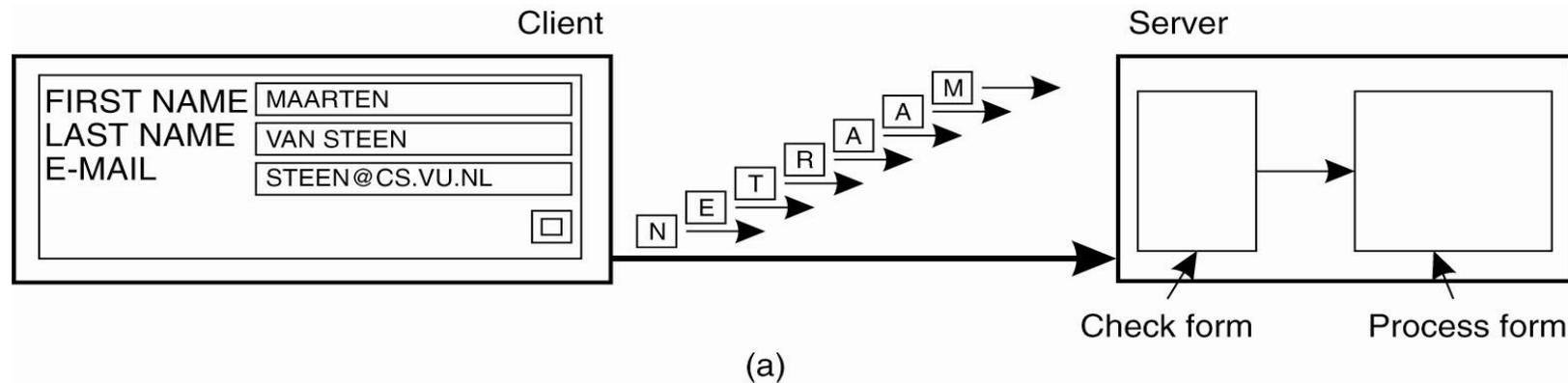
Técnicas para Escalabilidade

- Esconder Latências de Comunicação – evitar o uso de “request/reponse” bem como efetuar alguma computação
 - utilização da comunicação assíncrona;
 - oferecer suporte para que resposta que cheguem seja devidamente tratadas por rotinas de tratamento de requisições;
- Problema: Considere uma Aplicação Interativa que após o envio de uma requisição não há mais nada por fazer! e.g. acesso a um banco de dados utilizando “forms”.
 - solução: enviar o código de preenchimento do formulário para o cliente, de modo que os campos sejam verificados, o que possibilita ao fim do preenchimento o envio do formulário completo.

Introdução – 1.2 Metas

Técnicas para Escalabilidade

- Solução: enviar o código de preenchimento do formulário para o cliente, o que possibilita ao fim do preenchimento o envio completo do formulário e a eliminação do problema.



Introdução – 1.2 Metas

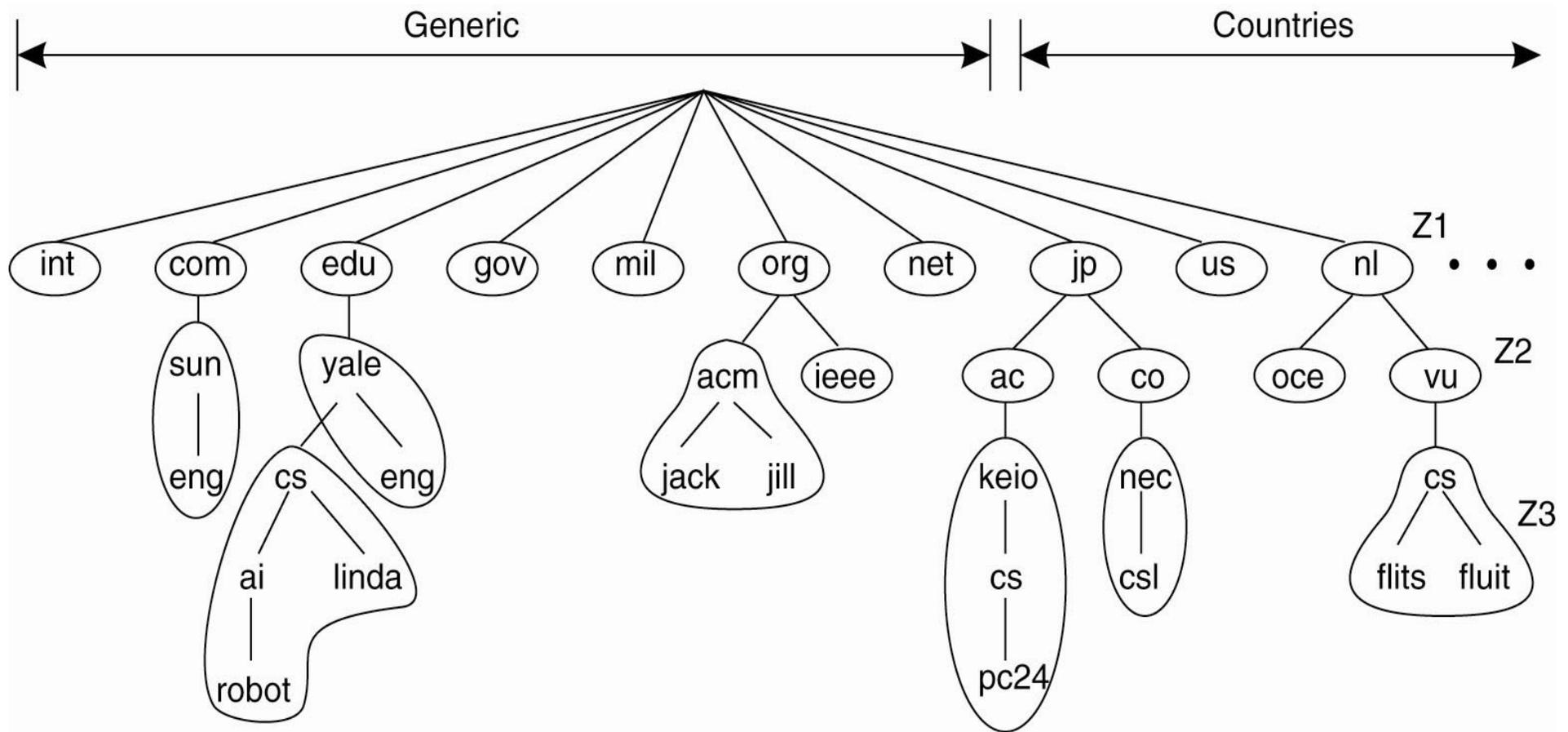
Técnicas para Escalabilidade

- Distribuição dos Dados e da Computação – particionar dados e computação e espalhá-los no sistema distribuído.
 - mover computação para os clientes (Java Applets)
 - Serviço DNS (Domain Name Service) – descentralizado
 - World Wide Web – serviço de informação descentralizado
- WWW – fisicamente distribuída em 1000 de 1000 de servidores, cada qual responsável por um dado número de documentos web.
- ... o nome do servidor que manipula um dado documento está codificado na URL (Universe Resource Locator), o que por outro lado justifica a escalabilidade da Web.

Introdução – 1.2 Metas

Técnicas para Escalabilidade

- Serviço DNS (Domain Name Service) – descentralizado



Introdução – 1.2 Metas

Técnicas para Escalabilidade

- Replicação e Armazenamento Local – efetuar cópias de dados em máquinas distintas
 - Servidores de Arquivos e Bancos de Dados replicados;
 - Sítios Web espalhados;
 - Caches Web (navegadores e proxies);
 - Cache de Arquivos (servidores e clientes).
- Replicação e “Caching” se diferem no tocante a quem toma a decisão de solicitar a cópia do dado (recurso).
- Adicionalmente, “caching” se dá por demanda (solicitação explícita do cliente) enquanto a replicação é frequentemente planejada com antecedência.
- Replicação/Caching => levam ao problema de consistência.

Introdução – 1.2 Metas

Técnicas para Escalabilidade

- Observações quanto a Aplicação das Técnicas:
 - existência de múltiplas cópias pode levar a inconsistências, pois alteração de uma cópia há torna distinta das demais;
 - manter cópias sempre consistentes de maneira geral requer sincronização global para cada modificação, o que por outro lado dificulta a escalabilidade.
- Se podemos tolerar inconsistências, podemos reduzir a necessidade de sincronização global. Por outro lado, tolerar inconsistência é dependente da aplicação.

1.2.5 - Pitfalls

- Muitos sistemas distribuídos são desnecessariamente complexos devido a erros de projeto que necessitaram ajustes posteriores.
- Há muitas falsas premissas:
 - rede confiável ?
 - rede é segura ?
 - rede é homogênea ?
 - topologia não muda ?
 - latência é próxima de zero ?
 - banda da rede é infinita ?
 - transporte é de custo zero ?
 - há um único administrador ?

Introdução – 1.2 Metas

... 1.2.5 - Pitfalls

- Muitas das suposições listadas estão relacionadas com as propriedades únicas dos sistemas distribuídos: confiabilidade; segurança, heterogeneidade; topologia da rede; latência e banda; custo do transporte e domínios administrativos.
- ... quando do desenvolvimento de aplicações não distribuídas, muitos dessas questões não são consideradas => logo escalar uma aplicação não distribuídas contempla uma série de armadilhas em função dos requisitos em cima dos quais a mesma foi concebida.
- Outros aspectos como questões de gerenciamento devem também ser considerados, pois tratam com falsas suposições tais como: custo zero de transporte e domínio administrativo único.

1.3 – Tipos de Sistemas Distribuídos

- Podemos categorizar os Sistemas Distribuídos em:
 - Sistemas de Computação Distribuídos;
 - Sistemas de Informação Distribuídos;
 - Sistemas Distribuídos Pervasivos

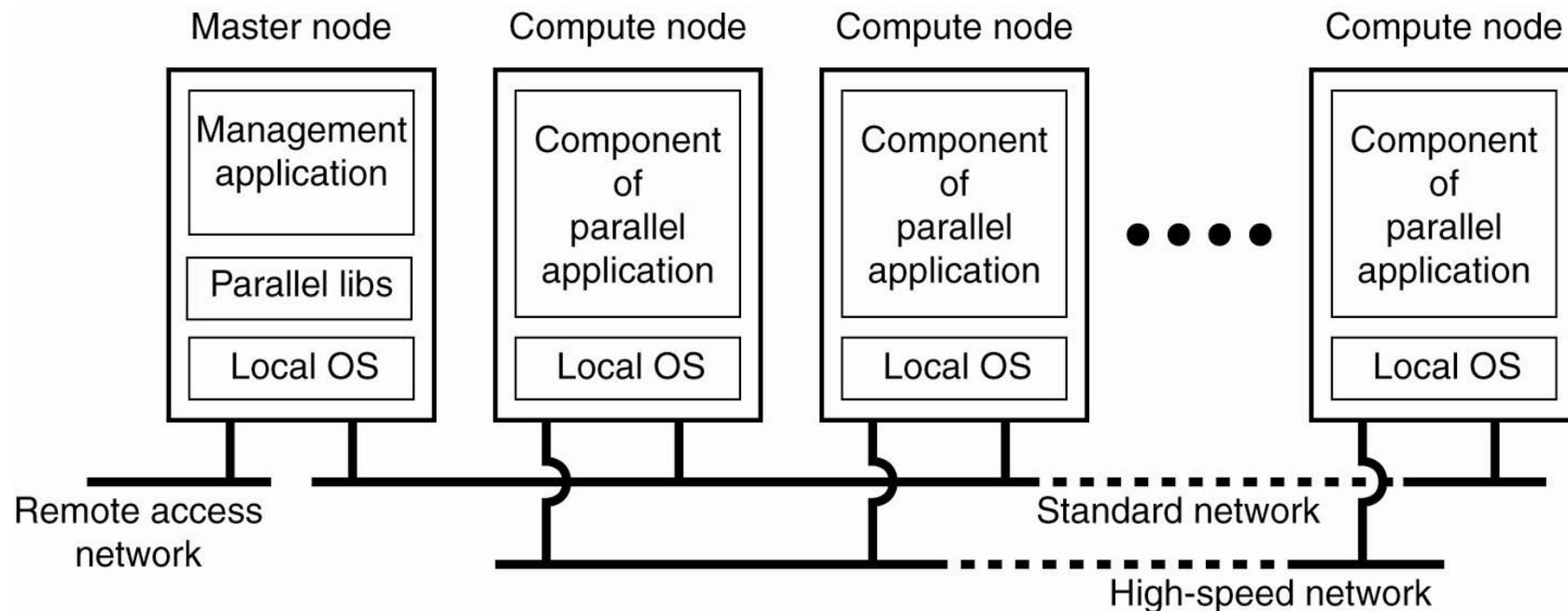
1.3.1 – Sistemas de Computação Distribuídos

- Uma classe importante de Sistemas Distribuídos está configurada para Computação de Alto Desempenho, que por sua vez pode ser subdividida em: Computação em Cluster e Computação em Grid.
- **Computação em Cluster** – os sistemas computacionais são consistidos de um coleção de estações de trabalho similares, fortemente conectadas por uma rede de alta velocidade com um nó funcionando como gerenciador.
- **Computação em Grid** – construído como uma federação de sistemas computacionais, onde cada sistema está sob um domínio administrativo e, portanto, pode contemplar heterogeneidade de “hardware”, “software” e tecnologia de rede.

Introdução – 1.3 Tipos de Sistemas Distribuídos

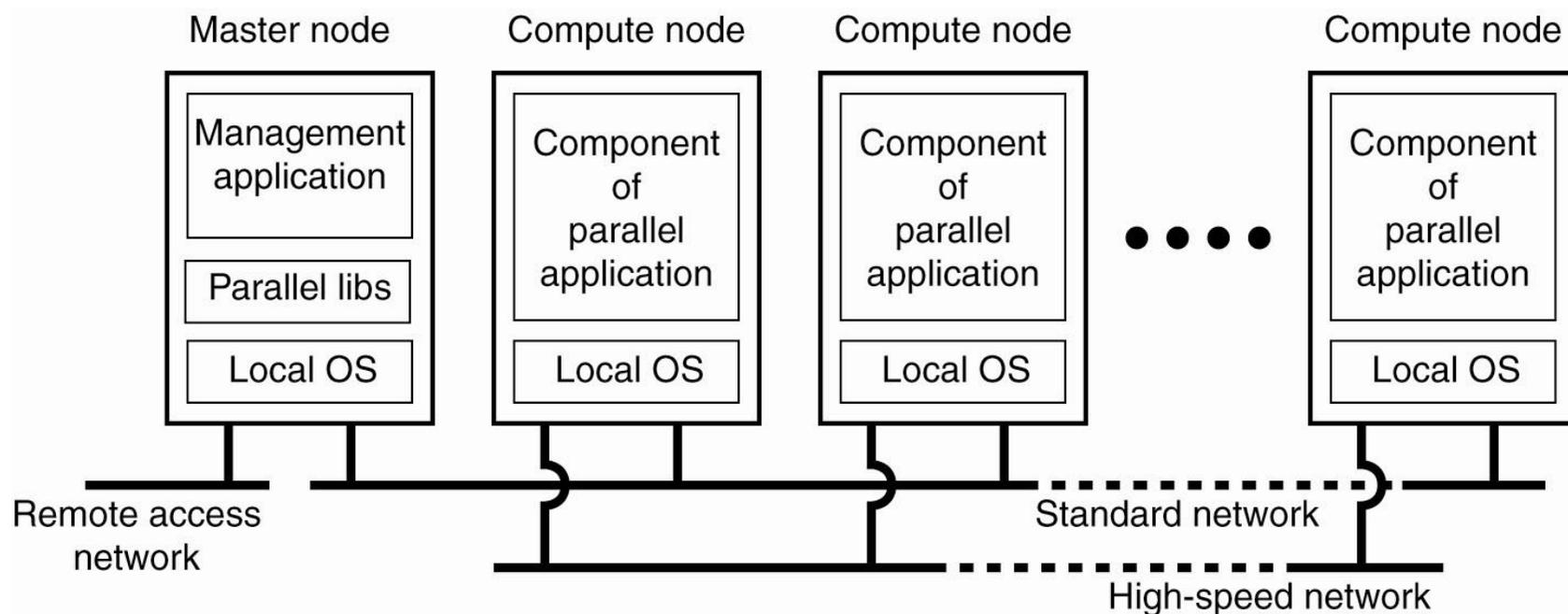
Computação em Cluster

- **Computação em Cluster** – os sistemas computacionais são consistidos de um coleção de estações de trabalho similares, fortemente conectadas por uma rede de alta velocidade com um nó funcionando como gerenciador.



Introdução – 1.3 Tipos de Sistemas Distribuídos ... Computação em Cluster

- Virtualmente em todos os casos, computação em cluster é utilizada para programação paralela na qual um único programa é executado em paralelo em múltiplas máquinas.
- e.g.: Linux-based Beowulf Cluster (configuração abaixo).



Introdução – 1.3 Tipos de Sistemas Distribuídos ... Computação em Cluster

- Linux-based Beowulf Cluster - uma parte importante do “middleware” é constituída por bibliotecas responsáveis pela execução de programas em paralelo.
- ... neste contexto, muitas destas bibliotecas provêm comunicação baseada em mensagens, mas não são hábeis para tratar processos defeituosos, segurança, etc.
- MOSIX – disponibiliza uma única imagem de sistema para o “cluster”, ou seja, oferece para cada processo transparência completa pois se comporta com um simples computador.

Computação em GRID

- **Computação em Grid** – construído como uma federação de sistemas computacionais/recursos, onde cada sistema está sob um domínio administrativo e, portanto, pode contemplar heterogeneidade de “hardware”, “software” e tecnologia de rede.
- ... tipicamente, recursos consistem de servidores(e.g., incluindo supercomputadores), “storages” e bancos de dados.
- ... dispositivos de rede especiais tais como sensores podem ser também contemplados.

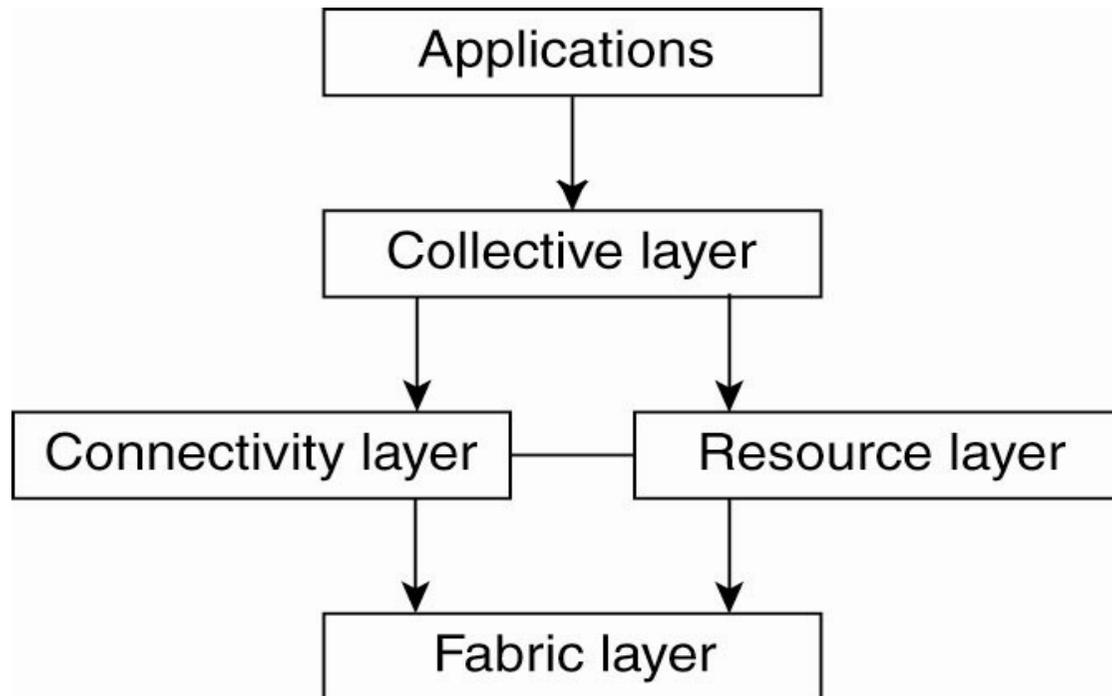
Introdução – 1.3 Tipos de Sistemas Distribuídos

Computação em GRID

- Considerando tais aspectos, a concepção de Computação em Grid envolve acesso a recursos de diferentes domínios administrativos e, através de usuários e aplicações que pertençam a uma organização virtual específica.
- ... para permitir colaborações, Grids geralmente usam organizações virtuais.
- ... em essência, um grupo de usuários (IDs) que permitem gerenciar autorizações e alocação de recursos.

Introdução – 1.3 Tipos de Sistemas Distribuídos ... Computação em GRID

- Por esta razão, a Arquitetura Proposta por Foster et al.:



Introdução – 1.3 Tipos de Sistemas Distribuídos ... Computação em GRID

- “Fabric Layer” - disponibiliza interfaces para recursos locais em “sites” específicos (note que estas interfaces são adaptadas para permitir compartilhamento dentro de uma organização virtual).
- “Connectivity Layer” - consiste de protocolos de comunicação para suportar transações em “grid” que necessitem utilizar múltiplos recursos.
- ... adicionalmente, contempla protocolos de segurança para autenticar usuários e recursos.
- “Resource Layer” - responsável pelo gerenciamento de recurso, assim utiliza as funções disponibilizadas pela camada de conectividade e invoca diretamente as interfaces disponibilizadas pela camada de fábrica.

Introdução – 1.3 Tipos de Sistemas Distribuídos ... Computação em GRID

- “Collective Layer” - consiste de serviços para descoberta e alocação de recursos bem como escalonamento de tarefas sobre múltiplos recursos, dados replicados e etc.
- ... por oferecer uma larga gama de serviços na organização virtual, esta camada de protocolos diferentes para diferentes propósitos.
- “Application Layer” - consiste de aplicações que operam dentro da organização virtual e as quais são responsáveis pelo ambiente de computação em “grid”.

1.3.2 – Sistemas de Informação Distribuídos

- Outra importante classe de sistemas distribuídos é encontrada em organizações que foram confrontadas com a qualidade das aplicações de rede, mas para as quais a interoperabilidade acabou por se tornar a experiência dolorosa.
- ... podemos distinguir vários níveis de integração:
 - integração no baixo nível – pode contemplar o agrupamento de requisições em uma requisição maior de modo que seja executada como uma transação distribuída;
 - integração de alto nível - “enterprise application integration”

Introdução – 1.3 Tipos de Sistemas Distribuídos

Sistema de Processamento de Transações

- A vasta maioria de sistemas distribuídos em uso hoje em dia são formas de sistemas de informação tradicionais, que agora integram sistemas legados.
- ... e.g., sistemas de processamento de transações.

```
BEGIN TRANSACTION(server, transaction)
READ(transaction, file-1, data)
WRITE(transaction, file-2, data)
newData := MODIFIED(data)
IF WRONG(newData) THEN
  ABORT TRANSACTION(transaction)
ELSE
  WRITE(transaction, file-2, newData)
END TRANSACTION(transaction)
END IF
```

Introdução – 1.3 Tipos de Sistemas Distribuídos
... Sistema de Processamento de Transações

- Transações contemplam as seguintes propriedades:
 - “atomic” - para o mundo exterior, a transação é indivisível;
 - “consistent” - a transação não viola invariantes do sistema;
 - “isolated” - transações concorrentes não interferem umas nas outras;
 - “durable” - transação completada => as alterações são permanentes.
- Estas propriedades são comumente referenciadas como ACID.

Introdução – 1.3 Tipos de Sistemas Distribuídos

Enterprise Application Integration

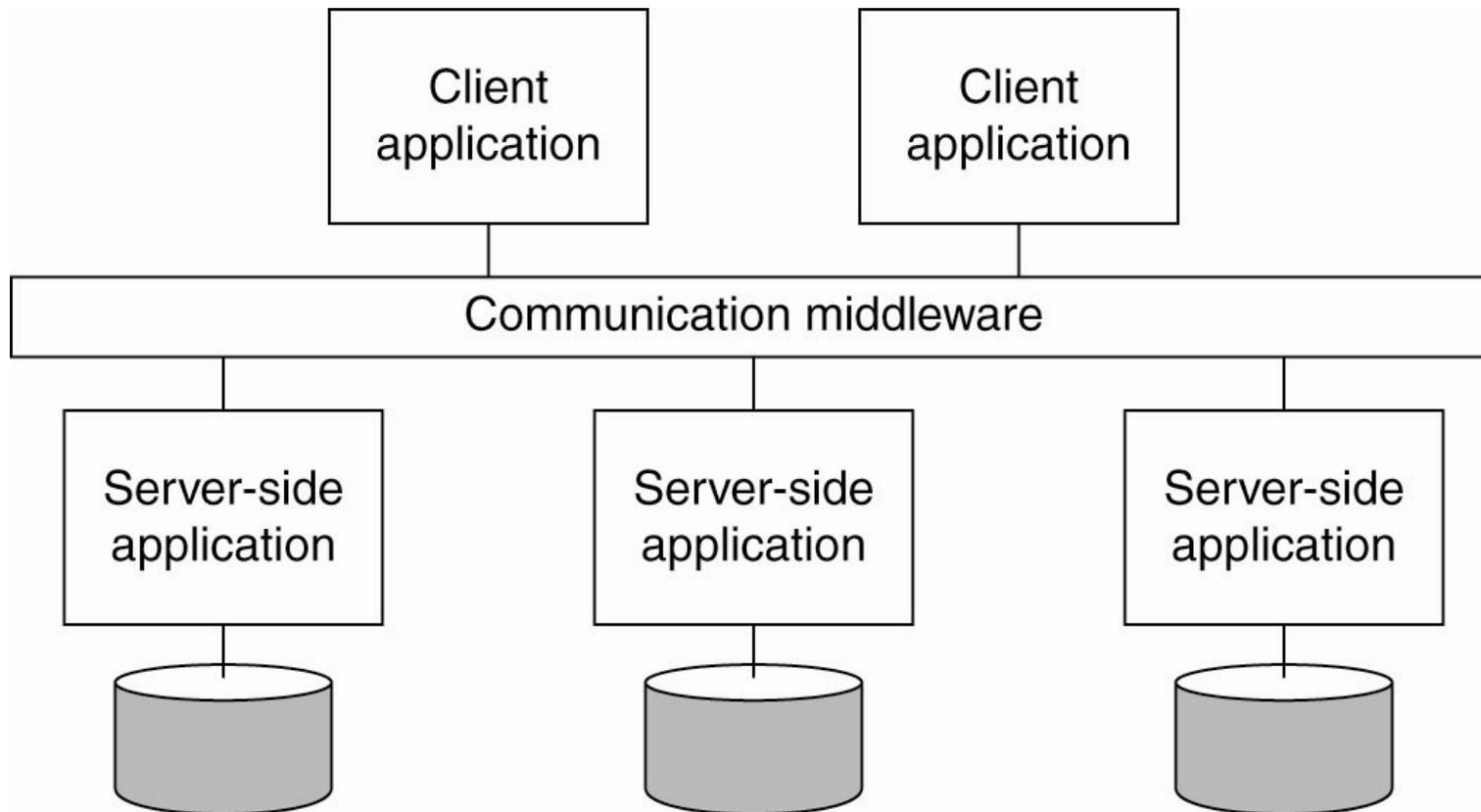
- A medida que aplicações se desacoplam de banco de dados, elas passam a ser construídas sobre as outras, pois a integração de aplicações passa a depender das suas bases de dados.
- ... a necessidade de comunicação entre as aplicações conduz a diferentes modelos de comunicação:
 - RPC (Remote Procedure Call) – uma aplicação envia uma requisição para um outro componente de aplicação invocando um chamada local, que resulta em uma requisição empacotada como uma mensagem.
 - ... o resultado será enviado de volta e retornado para a aplicação como o resultado da chamada do procedimento.

Introdução – 1.3 Tipos de Sistemas Distribuídos ... Interprise Application Integration

- ... a necessidade de comunicação entre as aplicações conduz a diferentes modelos de comunicação:
 - RMI (Remote Method Invocation) – é essencialmente o mesmo que o RPC, exceto que opera sobre objetos em vez de aplicações.
- RPC e RMI apresentam como desvantagens:
 - ambos (solicitante e receptor) estão executando no momento da comunicação, senão, as mensagens serão perdidas;
 - ambos necessitam conhecer como devem fazer para se comunicarem.
- Como mencionado, este forte acoplamento é uma desvantagem que conduzir/permitiu o que se conhece por: “Message-Oriented Middleware – MOM”
 - tais sistemas são conhecidos como “publish/subscriber system”.

Introdução – 1.3 Tipos de Sistemas Distribuídos ... **Interprise Application Integration**

- ... a necessidade de comunicação entre as aplicações conduz a diferentes modelos de comunicação.



1.3.3 – Sistemas Distribuídos Pervasivos

- Sistemas Distribuídos em que nós são pequenos, mantidos por bateria, móveis, passíveis de conexão através de rede sem fio e geralmente embutidos em um sistema maior.
- Requisitos para Aplicações Pervasivas (Difusas):
 - troca contextual – o sistema é parte de um ambiente em que mudanças devem ser rapidamente percebidas.
 - composição ad hoc – cada nó pode ser usado de diferentes formas por diferentes usuários, no entanto, requer facilidade de configuração;
 - compartilhamento é o padrão – nós entram e saem provendo serviço compartilhado e informação.
- Pervasão (?!) e Transparência de Distribuição – bons parceiros ?

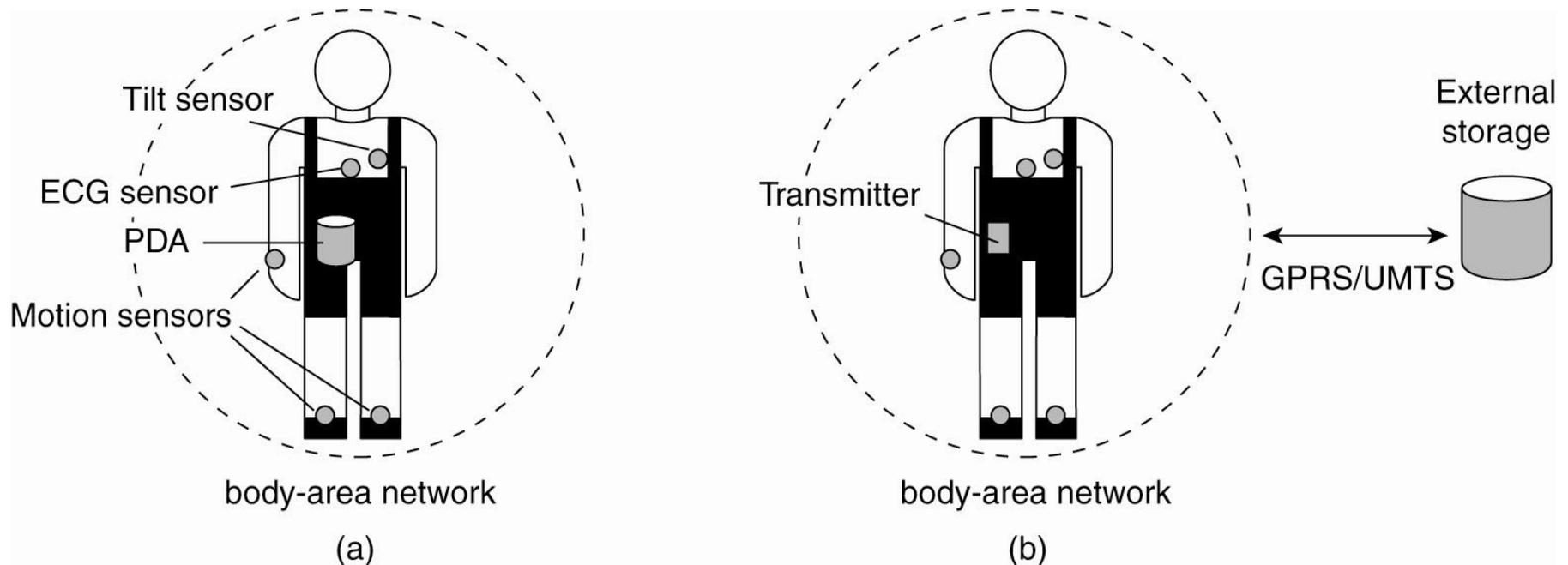
... 1.3.3 – Sistemas Distribuídos Pervasivos

- Home Systems – devem contemplar a auto configuração e o auto gerenciamento, ou seja, não deve ser necessário a existência de um administrador e o sistema ao mesmo tempo deve prover espaço pessoal para cada usuário.
- Health Care Systems – são sistemas equipados com sensores organizados sobre o corpo humano (BAN – Body Area Network).
- ... um requisito importante é o de não impedir ou limitar os movimentos da pessoa, ou seja, ausência de fios ligados aos dispositivos/sensores da rede.
- ... estes requisitos conduzem a 02 cenários: hub central com parte da rede; todos os componentes/sensores ligados preferencialmente via rede sem fio a uma rede externa.

Introdução – 1.3 Tipos de Sistemas Distribuídos

... 1.3.3 – Sistemas Distribuídos Pervasivos

- ... estes requisitos conduzem a 02 cenários: hub central com parte da rede; todos os componentes/sensores ligados preferencialmente via rede sem fio a uma rede externa.

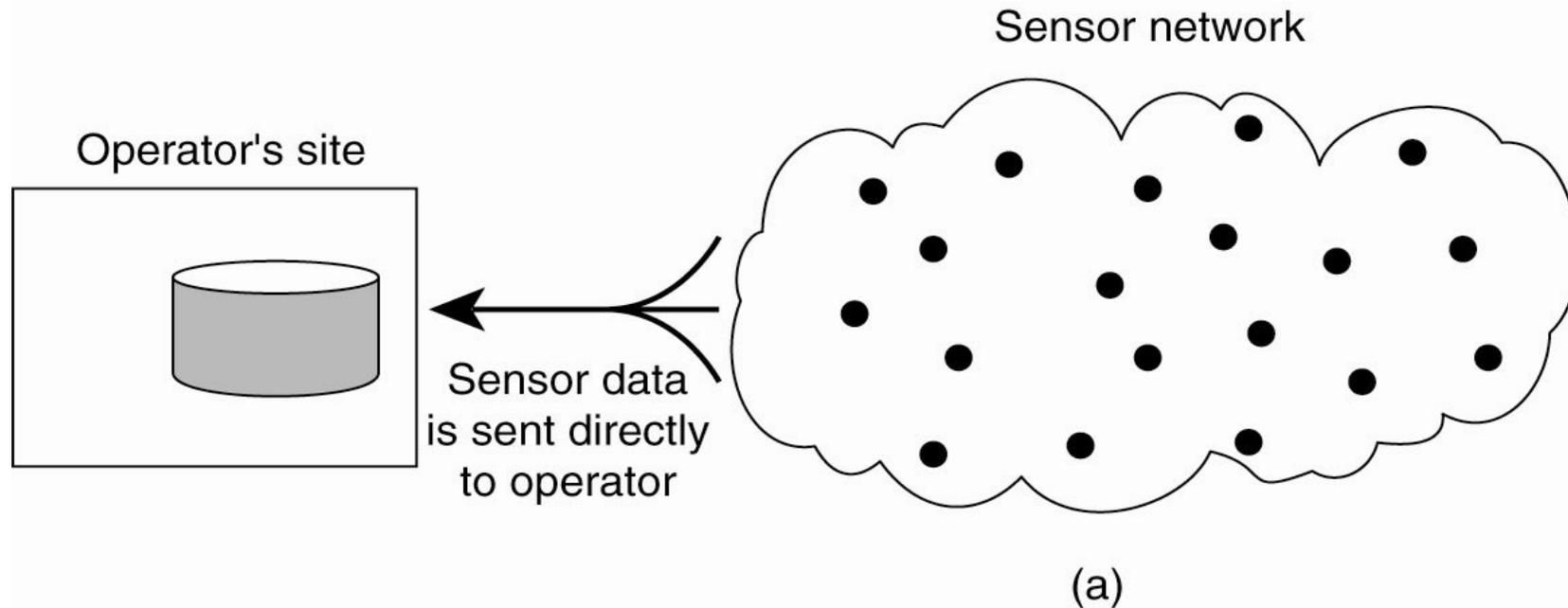


... 1.3.3 – Sistemas Distribuídos Pervasivos

- ... da perspectiva dos sistemas distribuídos nos deparamos com algumas questões tais como:
 - onde e como devem ser armazenados os dados monitorados ?
 - como prevenir perda de dados cruciais ?
 - como propagar alertas ?
 - como garantir segurança ?
 - como médicos podem prover feedback online?

... 1.3.3 – Sistemas Distribuídos Pervasivos

- Sensor Networks – tipicamente consiste de 10s a 1000s de pequenos nós frequentemente mantidos por bateria e com suporte a comunicação via rede sem fio.
- ... cada nó está equipado com um dispositivo sensor.



... 1.3.3 – Sistemas Distribuídos Pervasivos

- Sensor Networks – tipicamente consiste de 10s a 1000s de pequenos nós frequentemente mantidos por bateria e com suporte a comunicação via rede sem fio.
- ... cada nó está equipado com um dispositivo sensor.

