ACH2147 - Desenvolvimento de Sistemas de Informação Distribuídos

Aula 04 – Tipos de Sistemas Distribuídos

Norton Trevisan Roman

29 de março de 2022

Tipos de sistemas distribuídos

- Sistemas para computação distribuída de alto desempenho
- Sistemas de informação distribuídos
- Sistemas distribuídos para computação pervasiva

Tipos de sistemas distribuídos

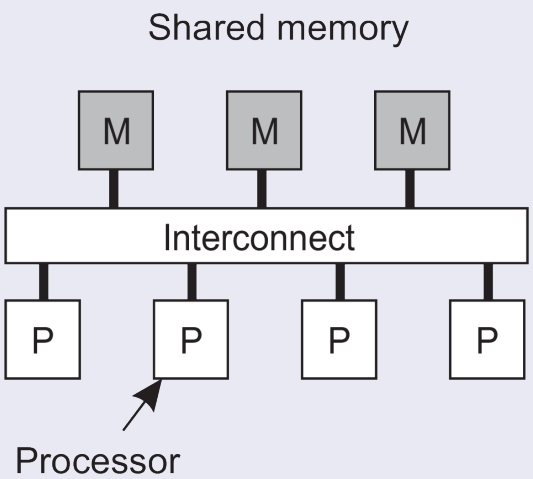
- **Sistemas para computação distribuída de alto desempenho**
- Sistemas de informação distribuídos
- Sistemas distribuídos para computação pervasiva

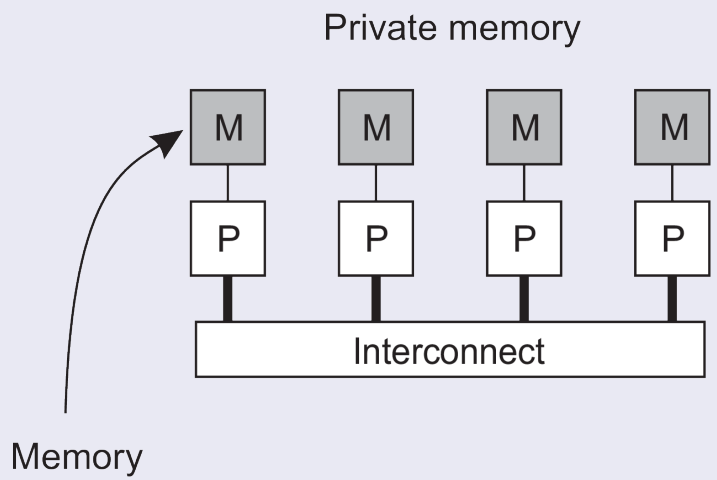
Computação paralela

- A computação distribuída de alto desempenho originou-se na computação paralela

Computação paralela

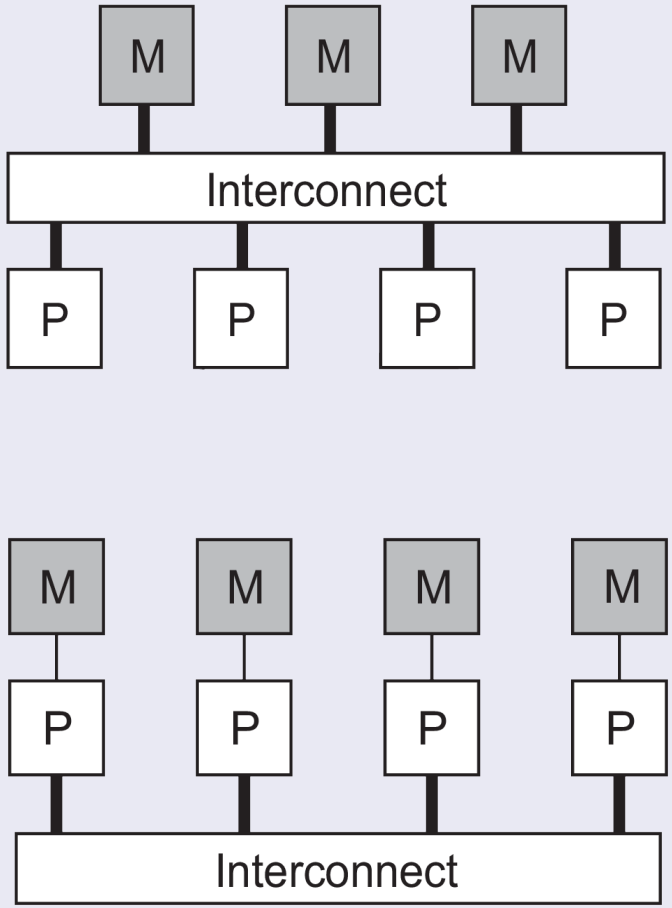
- A computação distribuída de alto desempenho originou-se na computação paralela
- Multiprocessadores $\times$ multicomputadores





Computação paralela

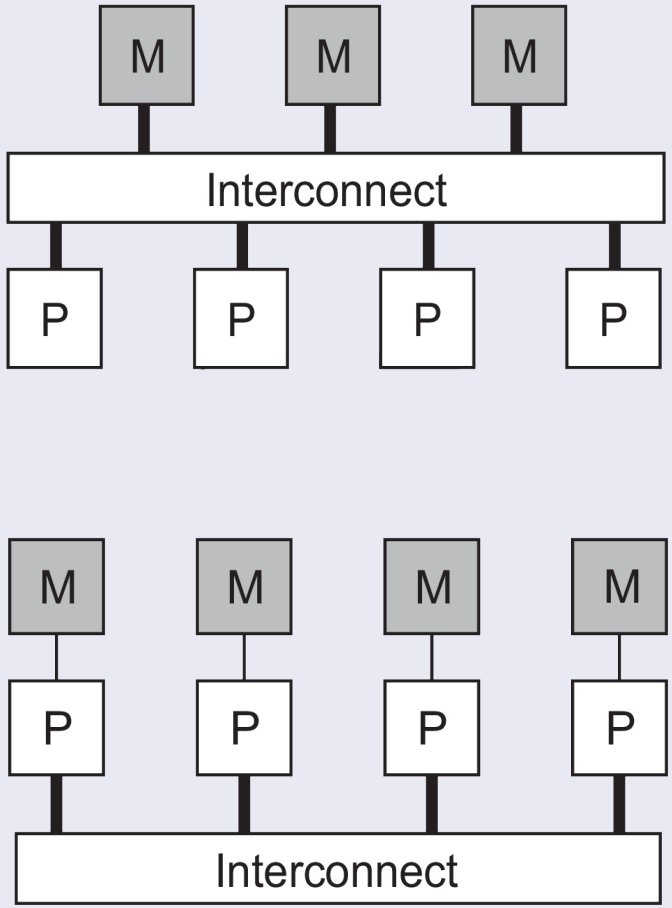
- Multiprocessadores são relativamente fáceis de programar se comparados a multicomputadores



Comput. distr. de alto desempenho

Computação paralela

- Multiprocessadores são relativamente fáceis de programar se comparados a multicomputadores
- Pena que é difícil de escalar
  - Não dispomos de um grande número de processadores (ou núcleos)

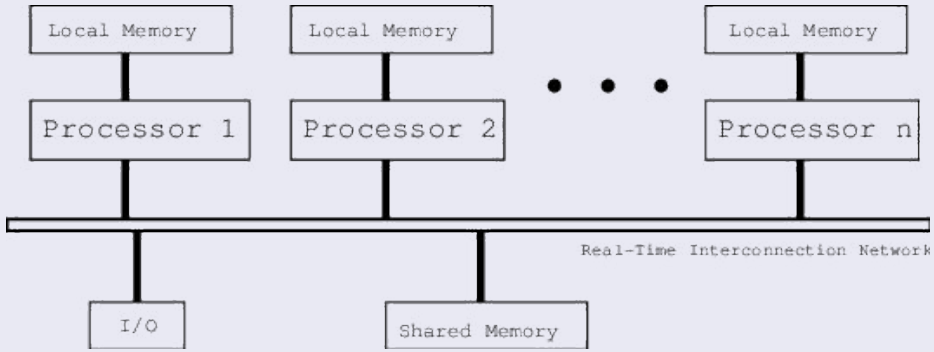


Computação paralela: Memória compartilhada

- Solução: tentar implementar um modelo de memória compartilhada para multicomputadores.

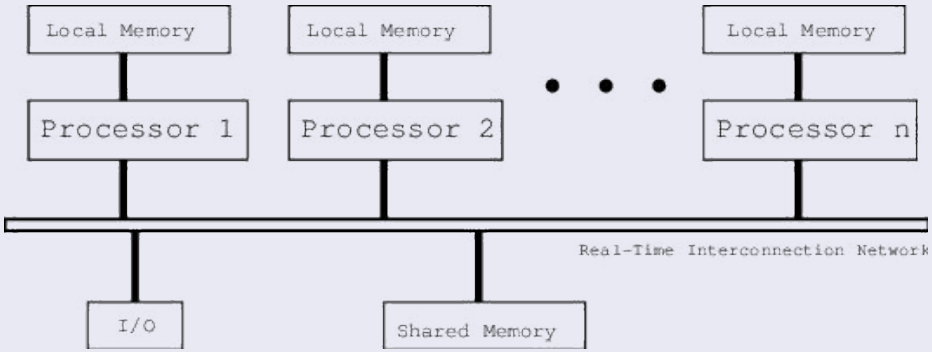
Computação paralela: Memória compartilhada

- Solução: tentar implementar um modelo de memória compartilhada para multicomputadores.
  - Mapear todas as páginas da memória principal (de todos os processadores) em um único espaço de endereçamento virtual.



Computação paralela: Memória compartilhada

- Solução: tentar implementar um modelo de memória compartilhada para multicomputadores.
  - Mapear todas as páginas da memória principal (de todos os processadores) em um único espaço de endereçamento virtual.



  - Se o processo no processador 1 referenciar uma página P localizada no processador 2, o SO em 1 lança uma interrupção e recupera P de 2, do mesmo modo que faria se fosse localmente.

Computação paralela: Memória compartilhada

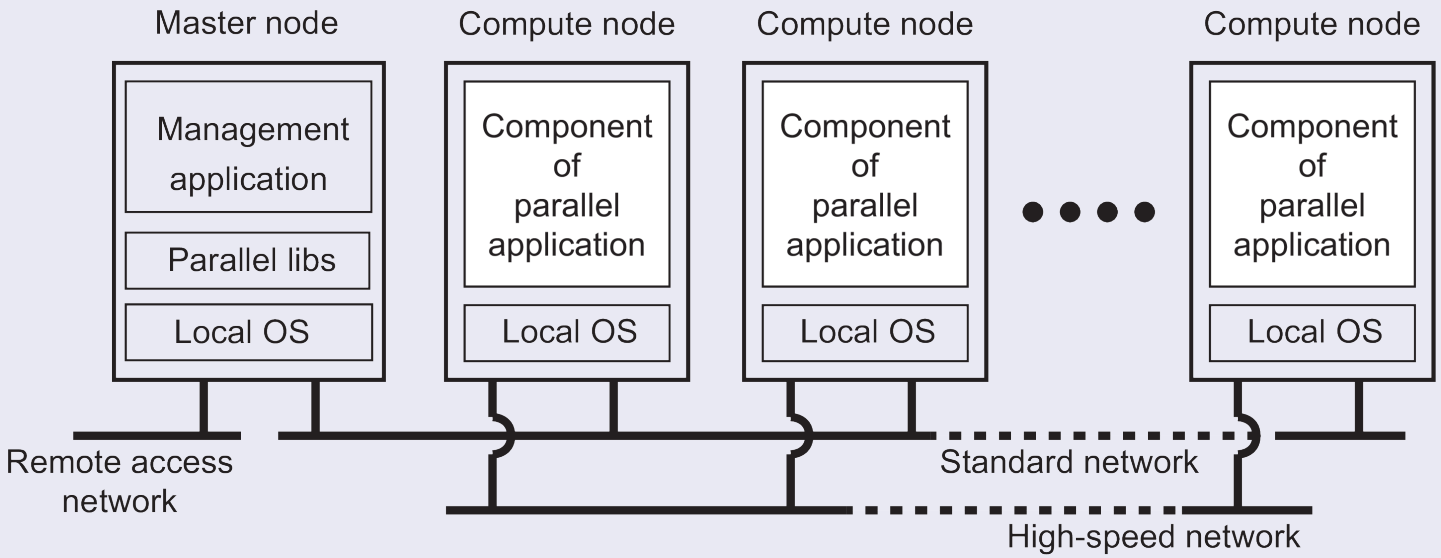
- O desempenho de um sistema de memória compartilhada distribuída nunca poderia competir com o desempenho de multiprocessadores
  - Por isso, a ideia foi abandonada

Computação paralela: Memória compartilhada

- O desempenho de um sistema de memória compartilhada distribuída nunca poderia competir com o desempenho de multiprocessadores
  - Por isso, a ideia foi abandonada (por enquanto...)

Cluster computing

- Essencialmente um grupo de computadores de boa qualidade conectados via LAN
  - Homogêneo: mesmo SO, hardware quase idêntico
  - Um único nó gerenciador



Computação em Grade (*Grid*)

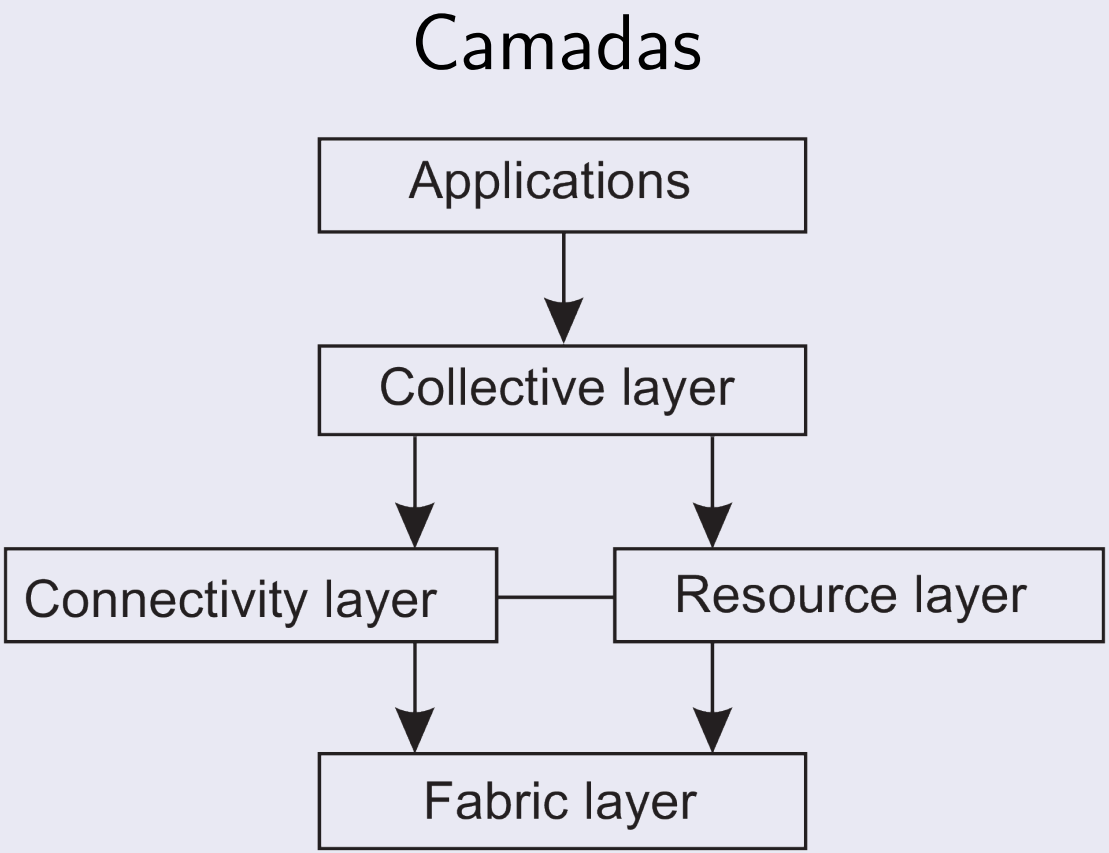
- O passo seguinte ao cluster: vários nós vindos de todos os cantos
  - Heterogêneos: nós são configurados para tarefas específicas
  - Espalhados entre diversas organizações
  - Normalmente formam uma rede de longa distância (*wide-area network*)

Comput. distr. de alto desempenho

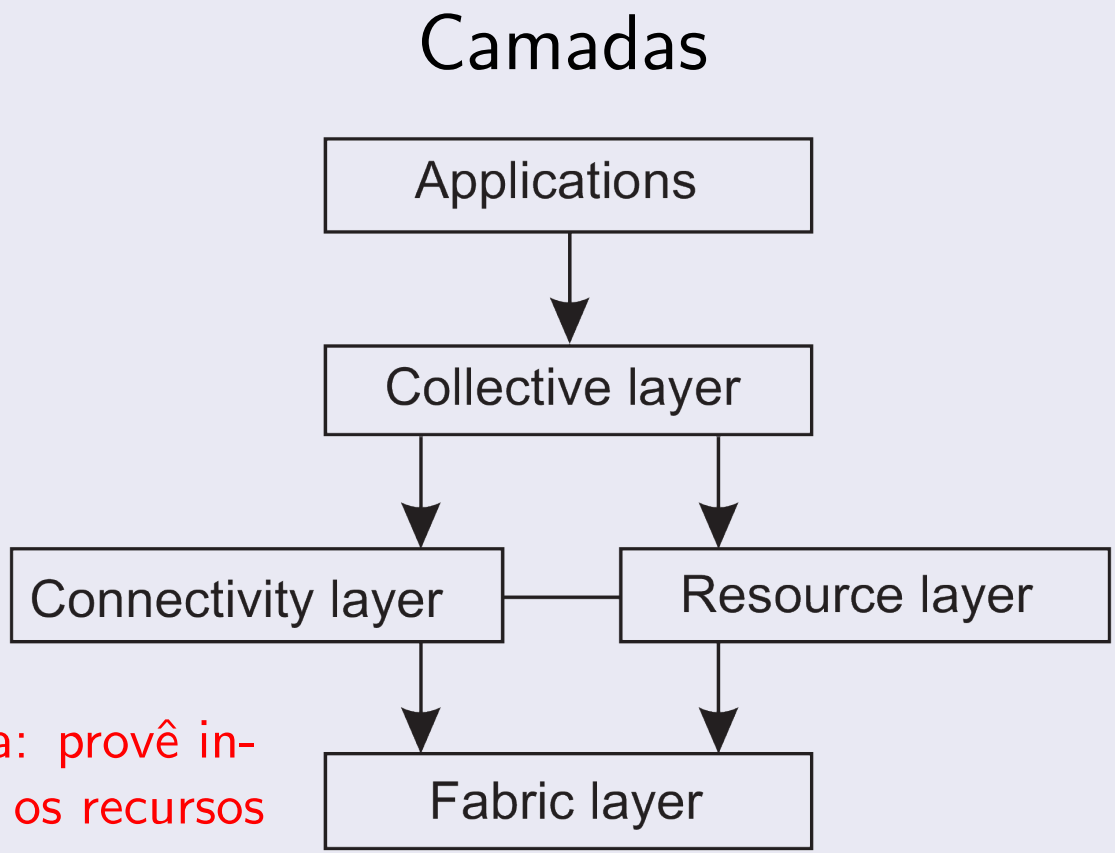
Computação em Grade (*Grid*)

- O passo seguinte ao cluster: vários nós vindos de todos os cantos
  - Heterogêneos: nós são configurados para tarefas específicas
  - Espalhados entre diversas organizações
  - Normalmente formam uma rede de longa distância (*wide-area network*)
- Para permitir colaborações, grades normalmente usam *organizações virtuais*
  - Essencialmente, os usuários (seus IDs) são organizados em grupos que possuem autorização para usar alguns recursos

Computação em Grade: arquitetura básica

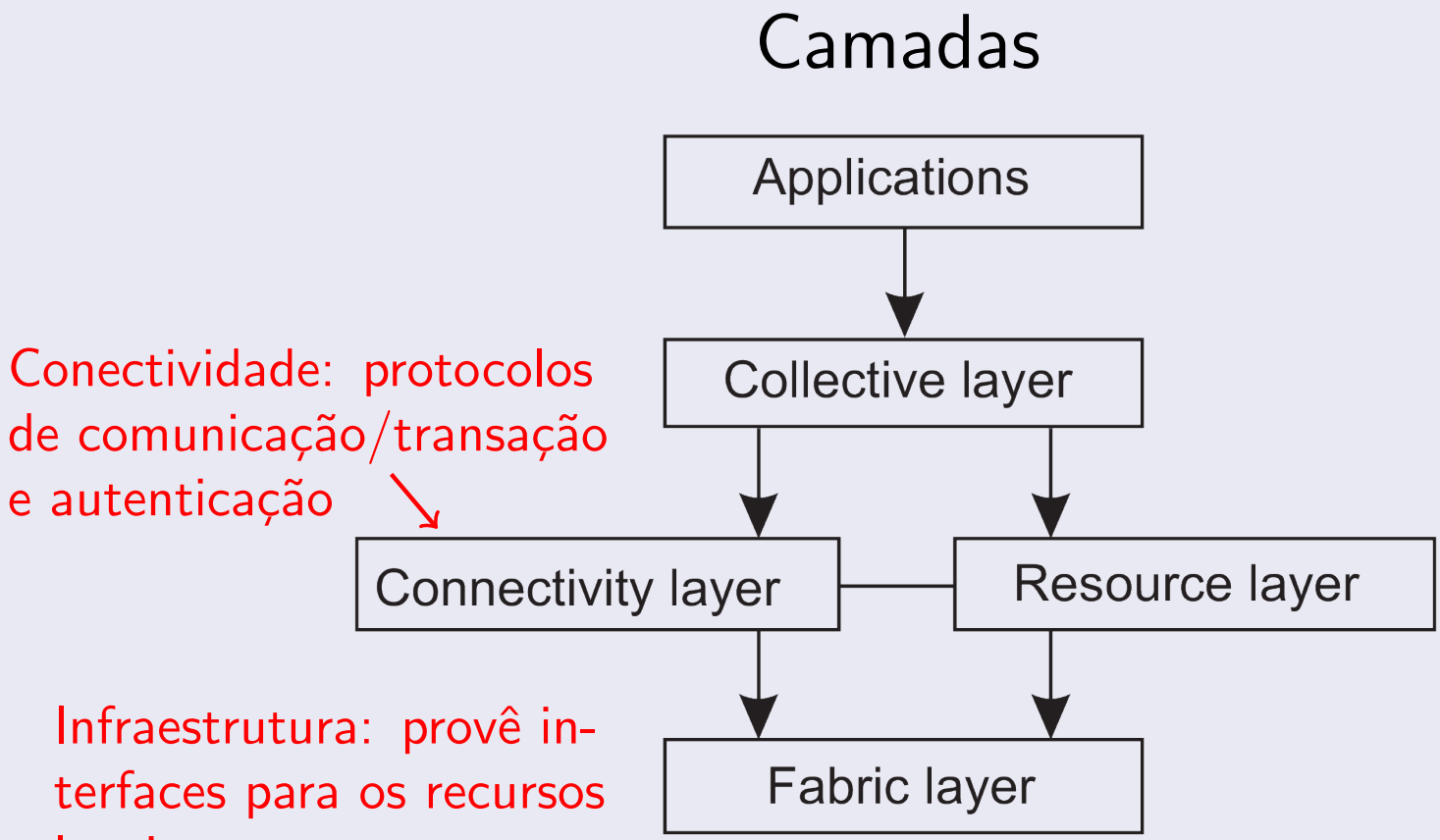


Computação em Grade: arquitetura básica

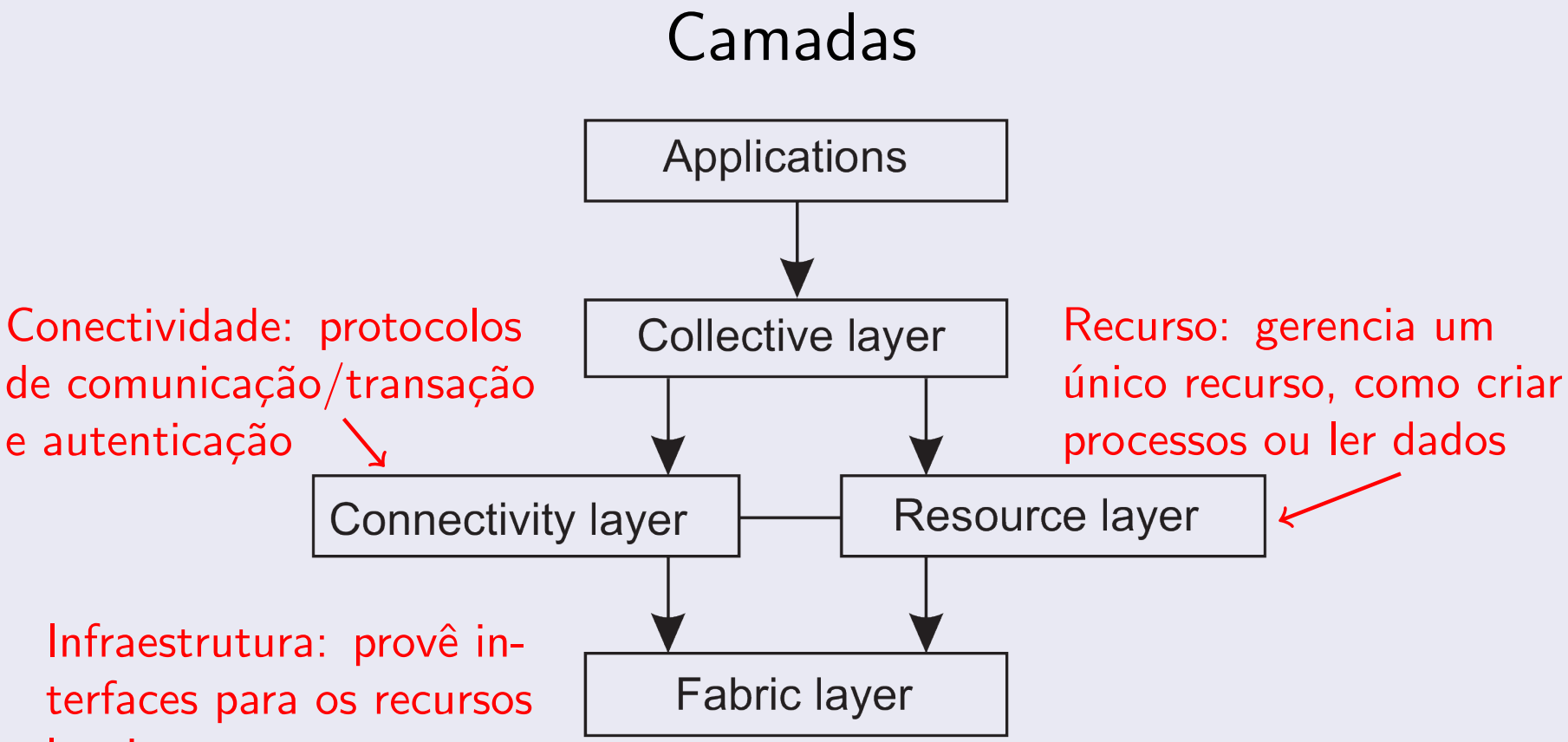


Infraestrutura: provê interfaces para os recursos locais

Computação em Grade: arquitetura básica



Computação em Grade: arquitetura básica



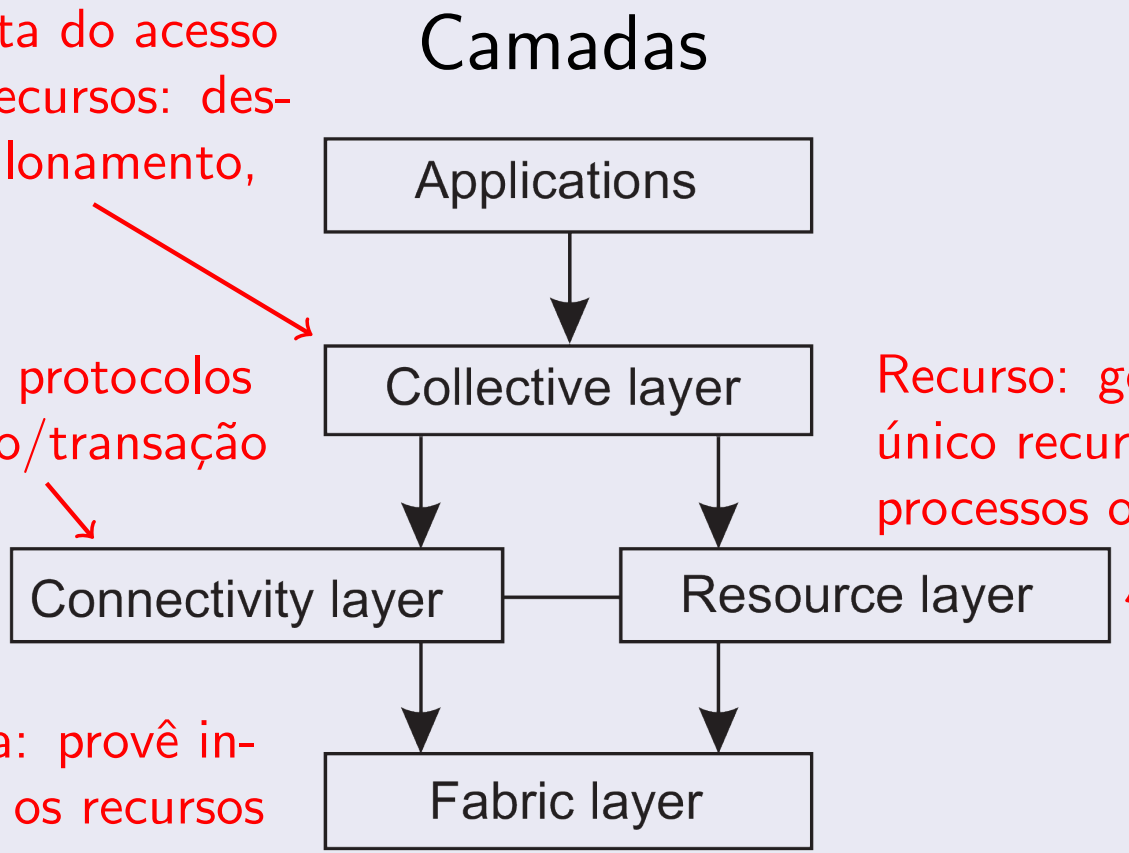
Computação em Grade: arquitetura básica

Coletiva: trata do acesso a múltiplos recursos: descoberta, escalonamento, replicação

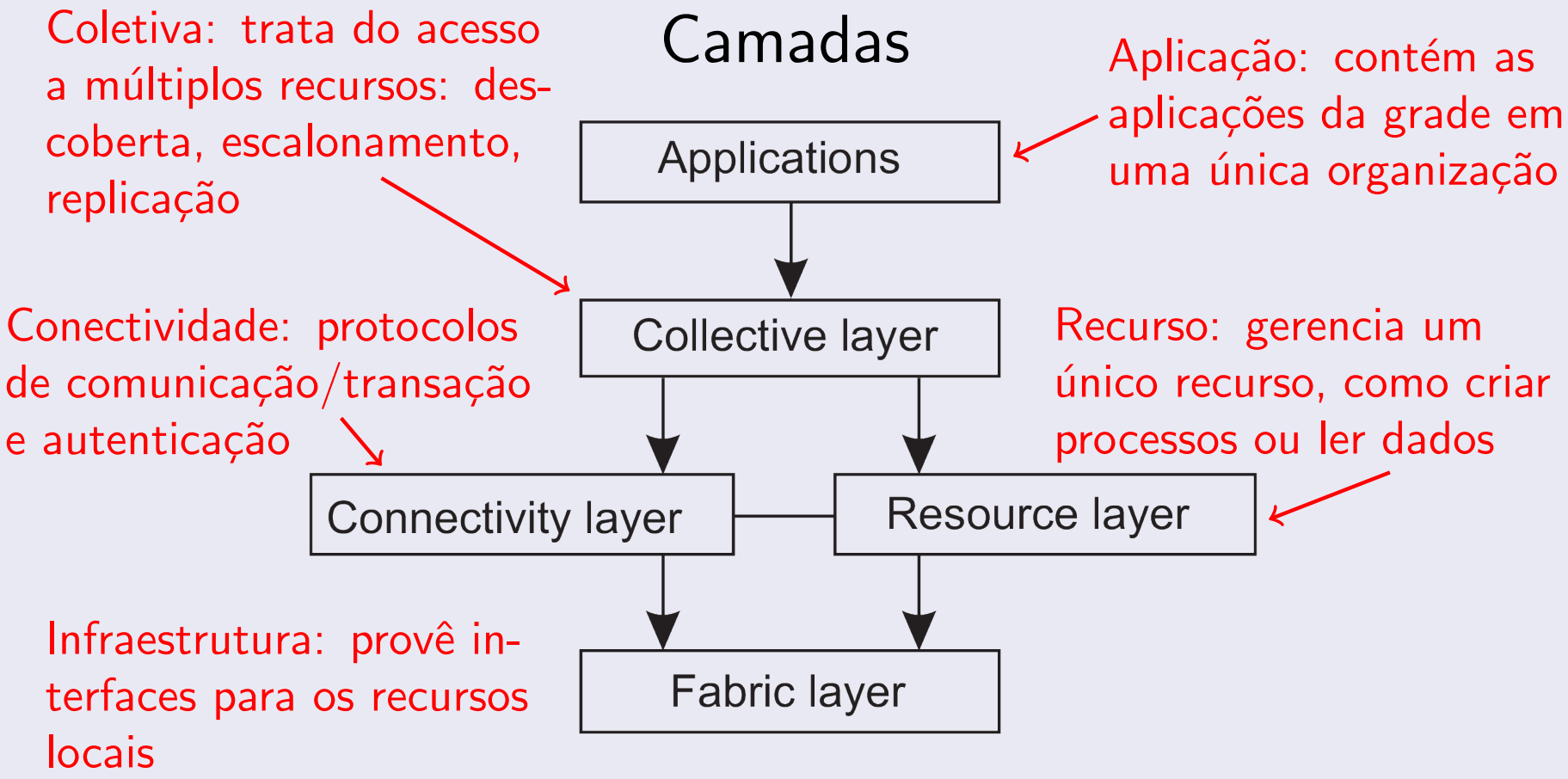
Conectividade: protocolos de comunicação/transação e autenticação

Recurso: gerencia um único recurso, como criar processos ou ler dados

Infraestrutura: provê interfaces para os recursos locais

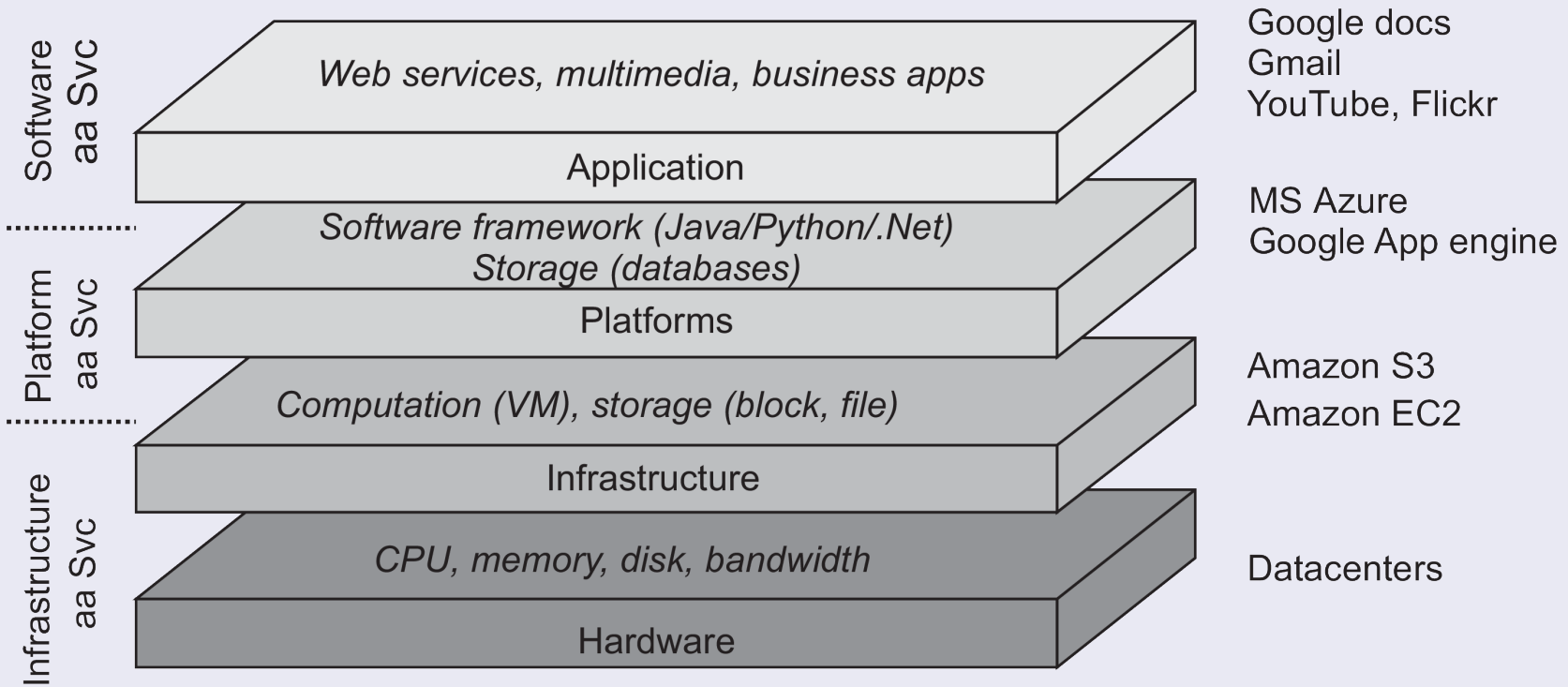


Computação em Grade: arquitetura básica



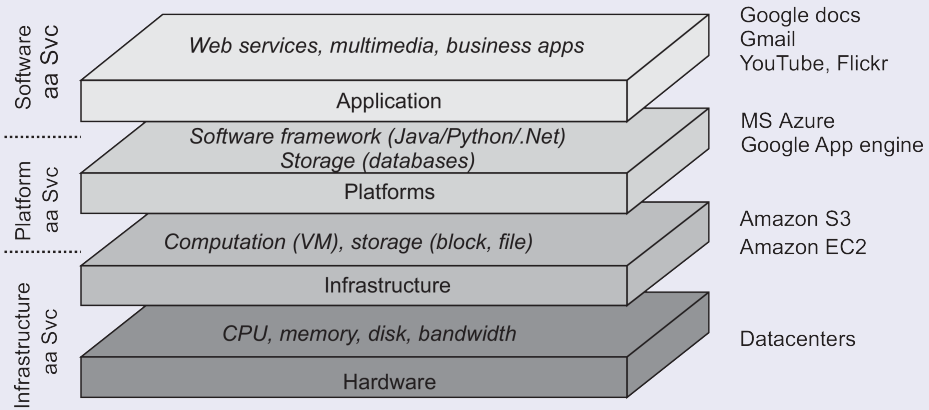
Computação em Nuvem

- Na prática, nuvens são organizadas em 4 camadas
  - Oferecidas ao usuário via interfaces para 3 tipos de serviços



Computação em Nuvem

- Infraestrutura
  - Utiliza técnicas de virtualização para alocação e gerenciamento de armazenamento e servidores virtuais



- Plataforma
  - Provê abstrações de alto nível para os serviços da plataforma
- Aplicação
  - As aplicações propriamente ditas

Tipos de sistemas distribuídos

- Sistemas para computação distribuída de alto desempenho
- **Sistemas de informação distribuídos**
- Sistemas distribuídos para computação pervasiva

Integrando aplicações

- Situação:
  - Organizações enfrentando muitas aplicações em rede, sendo que atingir interoperabilidade era problemático

Sistemas de informação distribuídos

Integrando aplicações

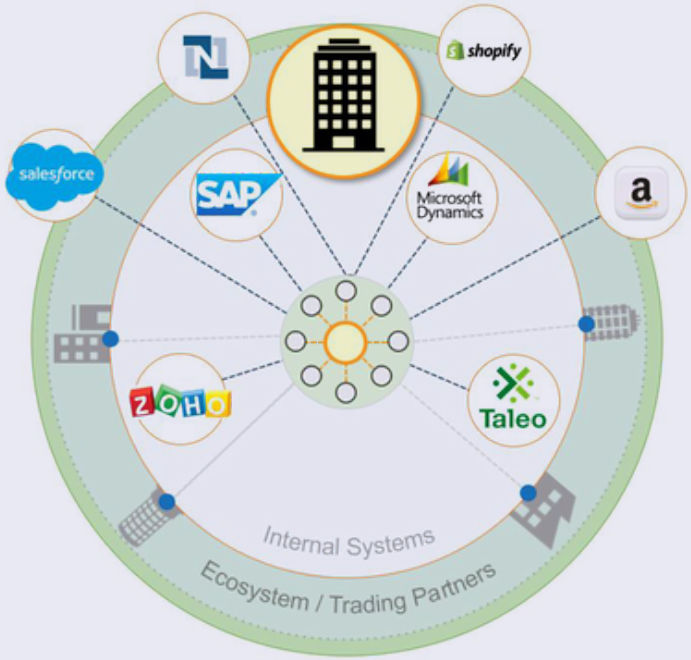
- Situação:
  - Organizações enfrentando muitas aplicações em rede, sendo que atingir interoperabilidade era problemático
- Abordagem básica:
  - Construir uma **aplicação em rede**: uma aplicação que roda em um servidor, disponibilizando seus serviços a clientes remotos
  - Integração simples: os cliente combinam requisições a diferentes aplicações, as enviam, obtêm as respostas, e apresentam um resultado coerente ao usuário

Integrando aplicações

- Próximo passo:
  - Permitir comunicação direta aplicação-aplicação

Integrando aplicações

- Próximo passo:
  - Permitir comunicação direta aplicação-aplicação
- Levando ao EAI (Enterprise Application Integration)
  - Meios computacionais e de arquitetura de sistemas usados no processo de Integração de Aplicações Corporativas



Integrando aplicações: transações (aninhadas)

- Transações são um exemplo de EAI
  - Uma quantidade enorme de sistemas distribuídos em uso hoje em dia são formas de sistemas de informação tradicionais, integrando sistemas legados

Integrando aplicações: transações (aninhadas)

- Transações são um exemplo de EAI
  - Uma quantidade enorme de sistemas distribuídos em uso hoje em dia são formas de sistemas de informação tradicionais, integrando sistemas legados
  - Tratados por meio de 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
```

Transações: Primitivas

- BEGIN_TRANSACTION
  - Marca o início de uma transação

```
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
```

Transações: Primitivas

- `BEGIN_TRANSACTION`
  - Marca o início de uma transação
- `END_TRANSACTION`
  - Termina uma transação e tenta fazer um *commit*

```
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
```

Transações: Primitivas

- `BEGIN_TRANSACTION`
  - Marca o início de uma transação
- `END_TRANSACTION`
  - Termina uma transação e tenta fazer um *commit*
- `ABORT_TRANSACTION`
  - Mata a transação e restaura os valores anteriores a ela

```
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
```

Transações: Primitivas

- READ
  - Lê dados de um arquivo, tabela etc

```
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
```

Transações: Primitivas

- READ
  - Lê dados de um arquivo, tabela etc
- WRITE
  - Escreve dados em um arquivo, tabela etc

```
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
```

Transações: Propriedades

- Uma transação é um conjunto de operações sobre o estado de um objeto (banco de dados, composição de objetos, etc.) que satisfazem as seguintes propriedades (ACID):
  - **Atomicidade**: ou todas as operações são bem sucedidas, ou todas falham. Quando uma transação falha, o estado do objeto permanecerá inalterado

Transações: Propriedades

- Uma transação é um conjunto de operações sobre o estado de um objeto (banco de dados, composição de objetos, etc.) que satisfazem as seguintes propriedades (ACID):
  - **Atomicidade**: ou todas as operações são bem sucedidas, ou todas falham. Quando uma transação falha, o estado do objeto permanecerá inalterado
  - **Consistência**: uma transação estabelece um estado de transição válido. Isto não exclui a existência de estados intermediários inválidos durante sua execução

Transações: Propriedades

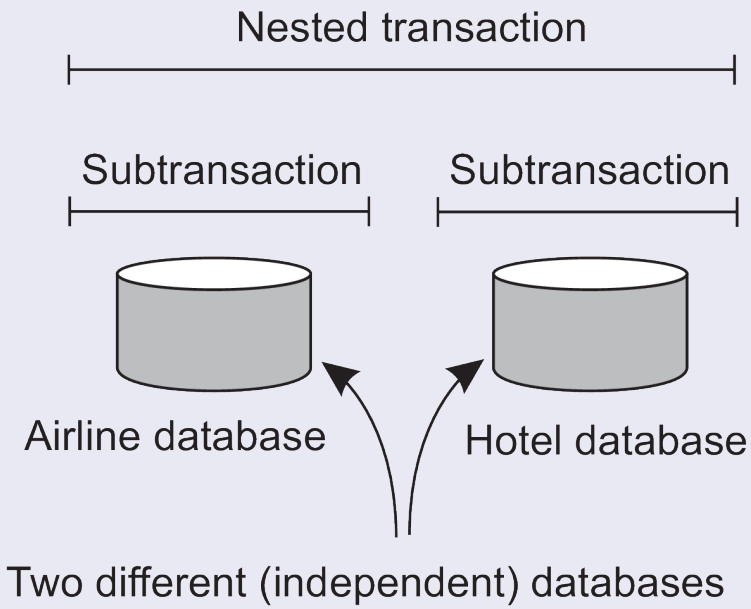
- (ACID):
  - **Isolamento**: transações concorrentes não interferem entre si. Para uma transação T, é como se as outras transações ocorressem ou *antes* de T ou *depois* de T

Transações: Propriedades

- (ACID):
  - **Isolamento**: transações concorrentes não interferem entre si. Para uma transação T, é como se as outras transações ocorressem ou *antes* de T ou *depois* de T
  - **Durabilidade**: Após o término de uma transação, seus efeitos são permanentes: mudanças de estado sobrevivem a falhas do sistema

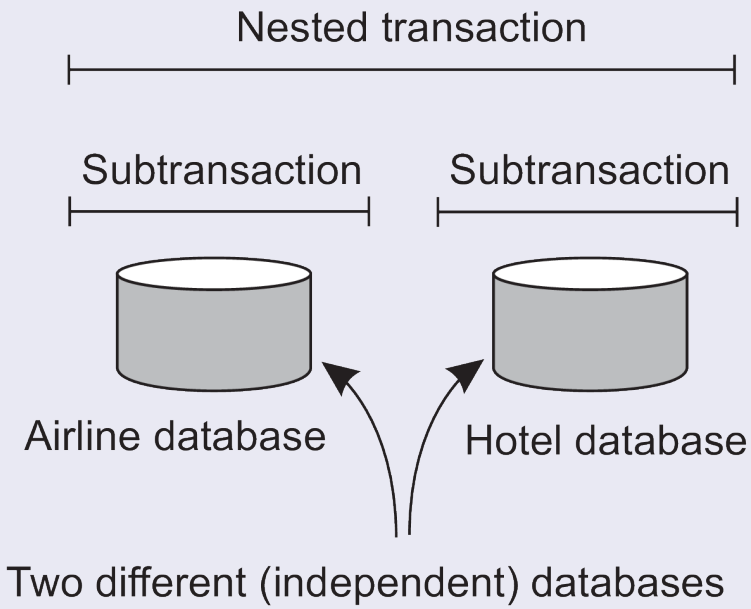
Transações: subtransações

- Transações podem ser construídas como um número de subtransações
  - Podendo rodar em paralelo, em máquinas distintas



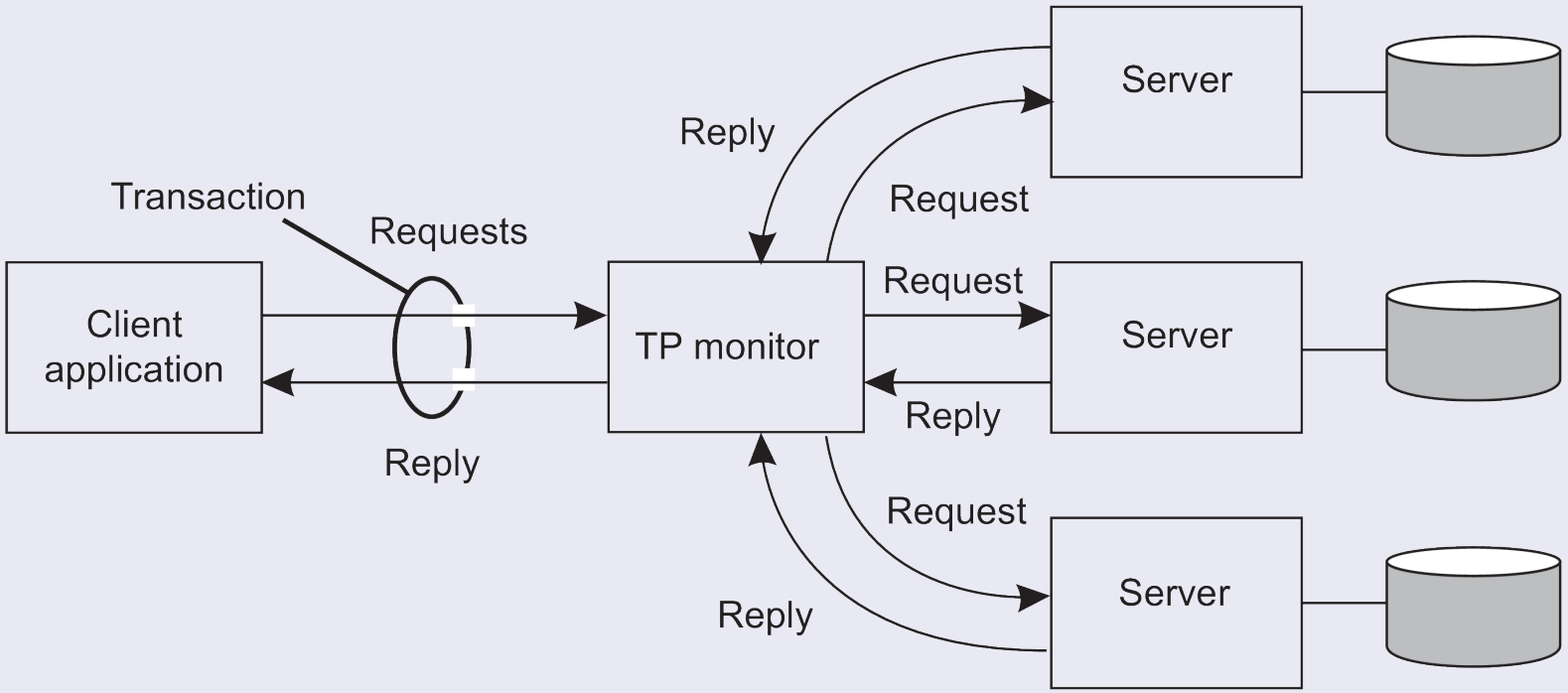
Transações: subtransações

- Transações podem ser construídas como um número de subtransações
  - Podendo rodar em paralelo, em máquinas distintas
- Juntas forma uma transição aninhada
  - Cada uma deve obedecer às propriedades das transações (ACID)



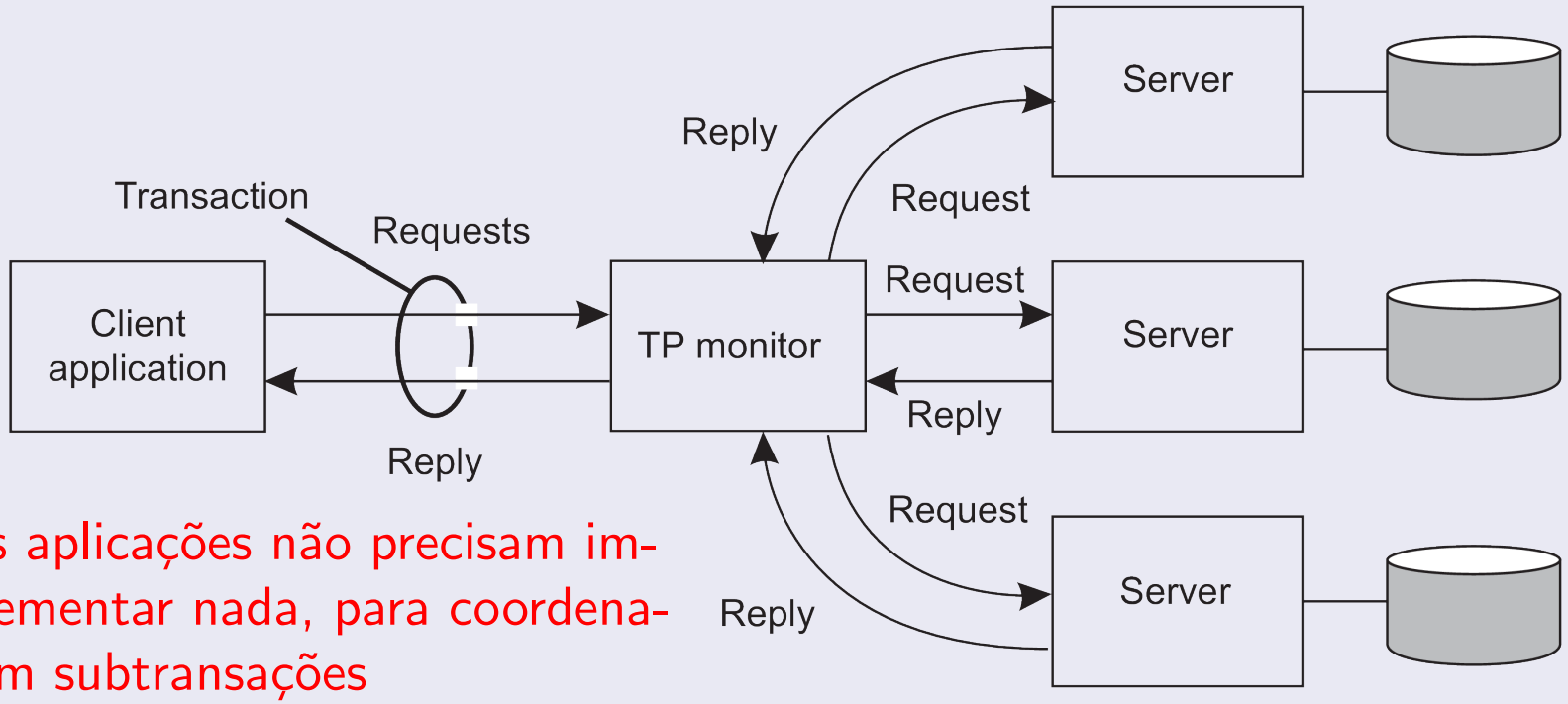
Transações: subtransações

Transações distribuídas (ou aninhadas) são coordenadas por um componente: o Transaction Processing Monitor



Transações: subtransações

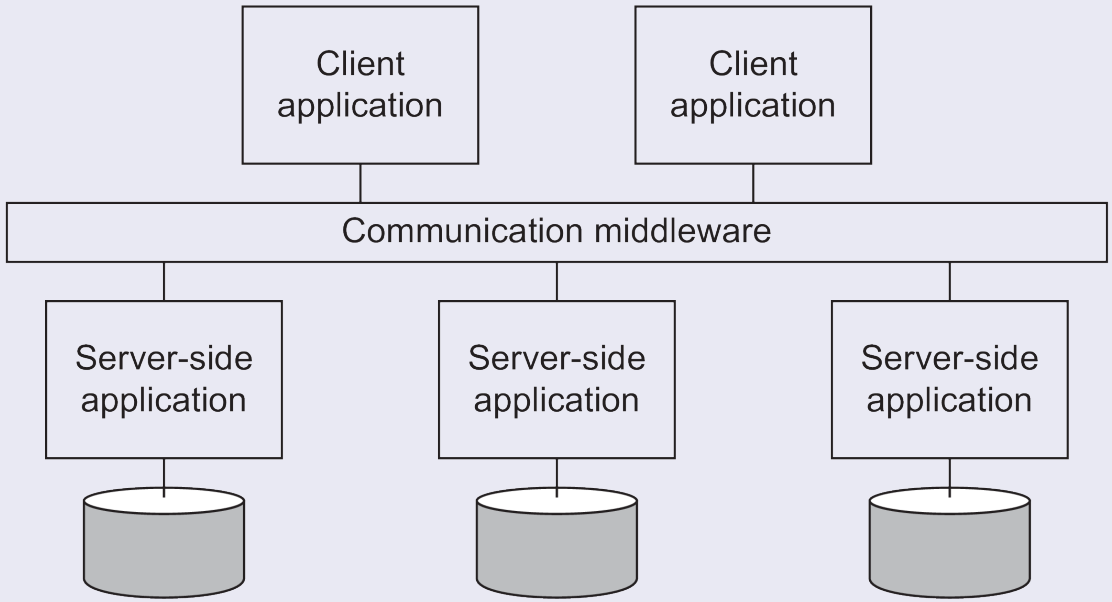
Transações distribuídas (ou aninhadas) são coordenadas por um componente: o Transaction Processing Monitor



As aplicações não precisam implementar nada, para coordenarem subtransações

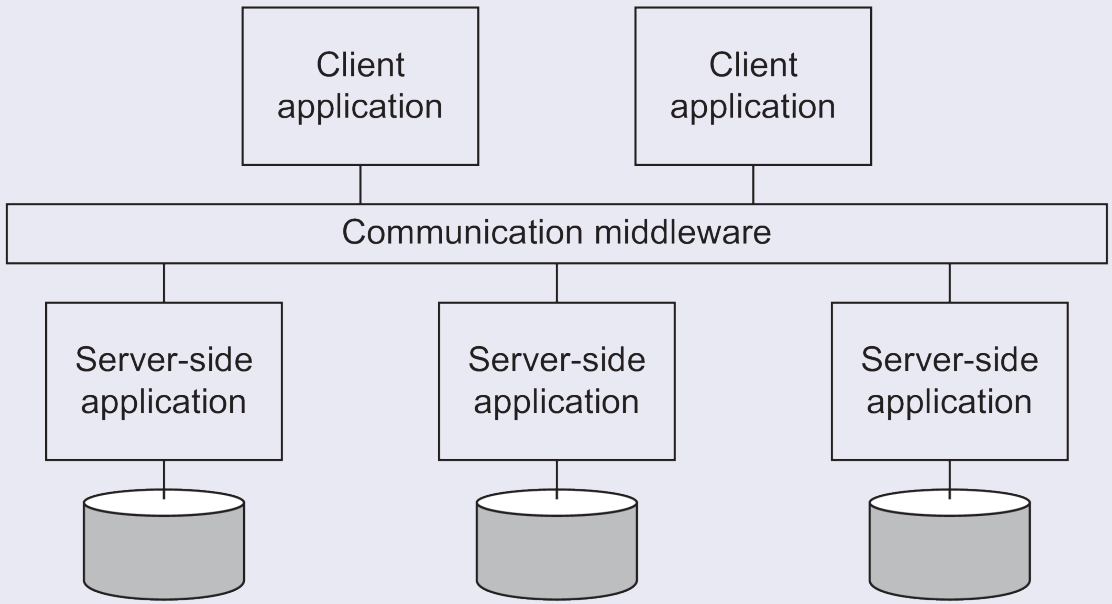
Middleware e Enterprise Application Integration

- Um TP Monitor não basta, também são necessários mecanismos para a comunicação direta entre aplicações
  - O middleware oferece facilidades de comunicação para a integração



Middleware: Mecanismos de comunicação

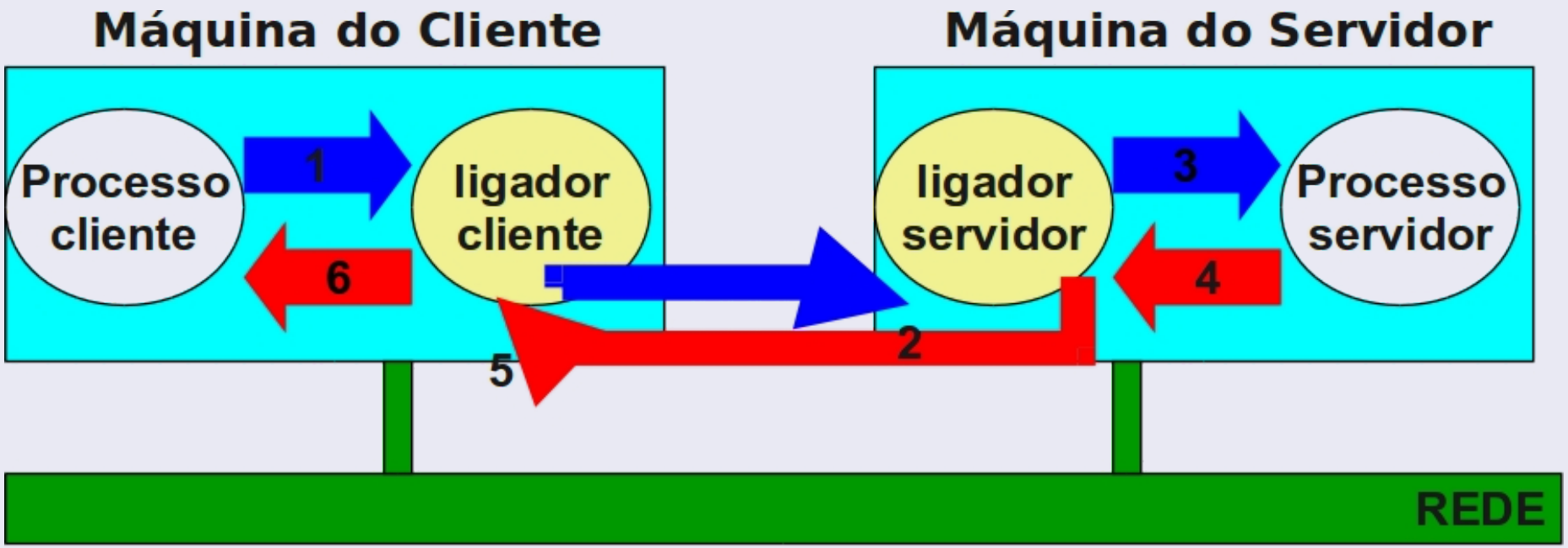
- Chamada de Procedimento Remoto (RPC)
- Middleware Orientado a Mensagens (MOM)



Middleware: Mecanismos de comunicação

- RPC – Remote Procedure Call
- Apresenta uma estrutura de comunicação na qual um processo pode comandar a execução de um procedimento situado em outra máquina
  - Requisições são feitas por meio de uma chamada de procedimento local
  - Esta são então empacotadas como mensagens, processadas no servidor, e respondidas como mensagens
  - O resultado é visto como o retorno da chamada ao procedimento local

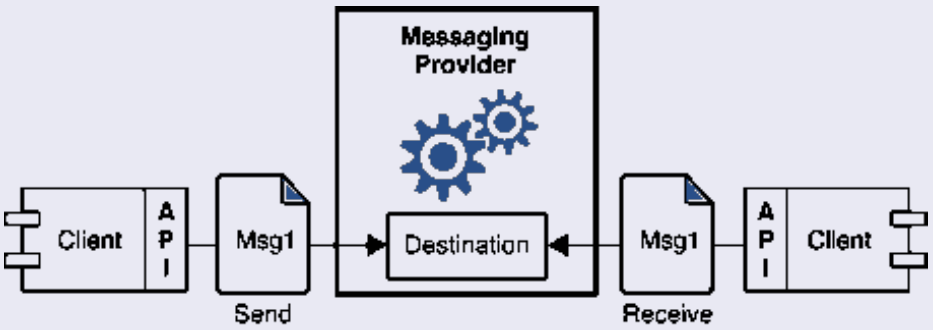
Middleware: RPC



(1) e (3) são chamadas de procedimento comuns
(2) e (5) são mensagens
(4) e (6) são retornos de procedimento comuns

Middleware: Mecanismos de comunicação

- MOM – Message Oriented Middleware
  - Mensagens são enviadas a um ponto de contato lógico (são publicadas)
  - Estas são então transmitidas a aplicações assinantes

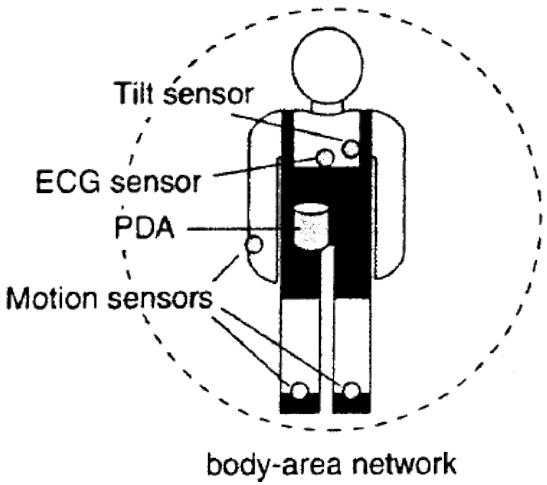


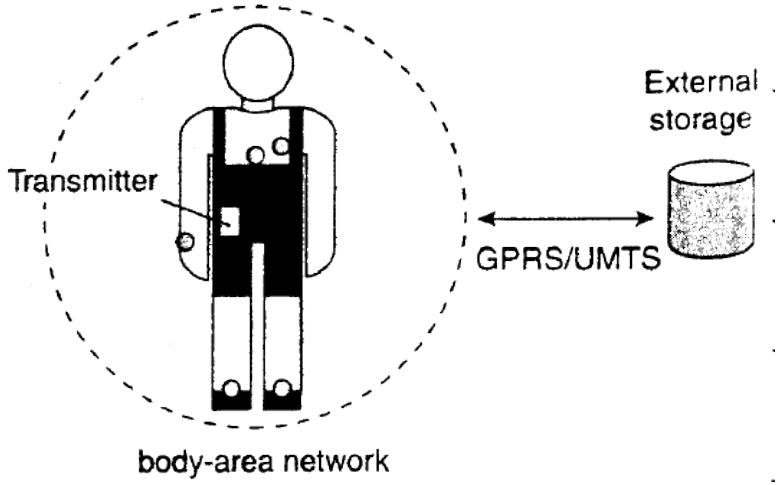
Tipos de sistemas distribuídos

- Sistemas para computação distribuída de alto desempenho
- Sistemas de informação distribuídos
- **Sistemas distribuídos para computação pervasiva**

Sistemas Distribuídos Pervasivos

- Tendência atual em sistemas distribuídos:
  - Nós pequenos, móveis e normalmente embutidos em um sistema muito maior
  - Caracterizados pelo fato do sistema naturalmente se misturar ao ambiente do usuário





Sistemas Distribuídos Pervasivos

Subtipos (há sobreposição)

- Sistemas Ubíquos:
  - Pervasivos e continuamente presentes
  - Há uma interação contínua entre o sistema e o usuário

Sistemas Distribuídos Pervasivos

Subtipos (há sobreposição)

- Sistemas Ubíquos:
  - Pervasivos e continuamente presentes
  - Há uma interação contínua entre o sistema e o usuário
- Móveis:
  - A ênfase está no fato dos dispositivos serem inerentemente móveis

Sistemas Distribuídos Pervasivos

Subtipos (há sobreposição)

- Sistemas Ubíquos:
  - Pervasivos e continuamente presentes
  - Há uma interação contínua entre o sistema e o usuário
- Móveis:
  - A ênfase está no fato dos dispositivos serem inerentemente móveis
- Redes de sensores:
  - Ênfase na leitura do e atuação no ambiente (colaborativas)

Sistemas Ubíquos: Elementos-chave

- Distribuição
  - Dispositivos em rede, distribuídos, e acessíveis de um modo transparente

Sistemas Ubíquos: Elementos-chave

- Distribuição
  - Dispositivos em rede, distribuídos, e acessíveis de um modo transparente
- Interação
  - A interação entre o usuário e o dispositivo é muito discreta

Sistemas Distribuídos Pervasivos

Sistemas Ubíquos: Elementos-chave

- Distribuição
  - Dispositivos em rede, distribuídos, e acessíveis de um modo transparente
- Interação
  - A interação entre o usuário e o dispositivo é muito discreta
- Noção de contexto
  - O sistema está ciente do contexto do usuário, de modo a otimizar a interação

Sistemas Ubíquos: Elementos-chave

- Autonomia
  - Os dispositivos operam autonomamente, sem intervenção humana

Sistemas Ubíquos: Elementos-chave

- Autonomia
  - Os dispositivos operam autonomamente, sem intervenção humana
- "Inteligência"
  - O sistema como um todo pode lidar com uma vasta gama de ações dinâmicas e interações

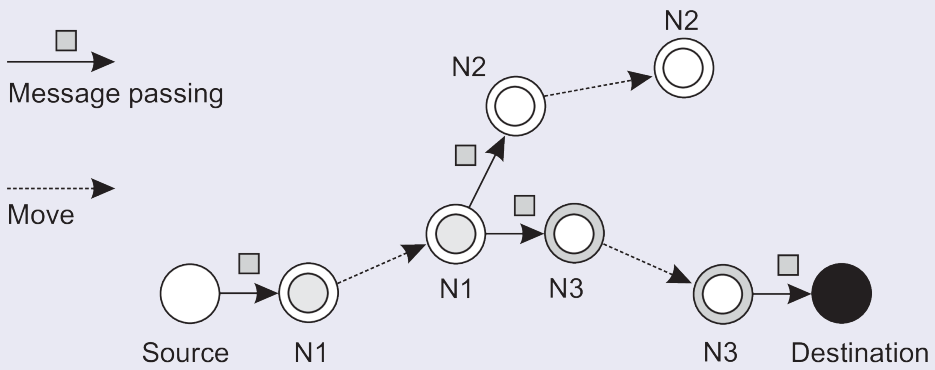
Sistemas Distribuídos Pervasivos

Coputação Móvel

- Dispositivos variam muito
  - Smartphones, tablets, dispositivos GPS etc
- Espera-se que a localização de um dispositivo varie com o tempo
  - Assim como os serviços disponíveis localmente
  - Serviços precisam ser descobertos dinamicamente
- A comunicação pode se tornar difícil
  - Sem rota estável entre dois dispositivos
  - Sem garantia de conexão

Coputação Móvel: Comunicação

- Disruption-tolerant networks
  - Redes em que a conectividade entre 2 nós não pode ser garantida
- Como enviar uma mensagem de um nó a outro?
  - Espalhando a mensagem pela rede (inundação)
  - Nós intermediários armazenam a mensagem até encontrar um nó ao qual passá-la



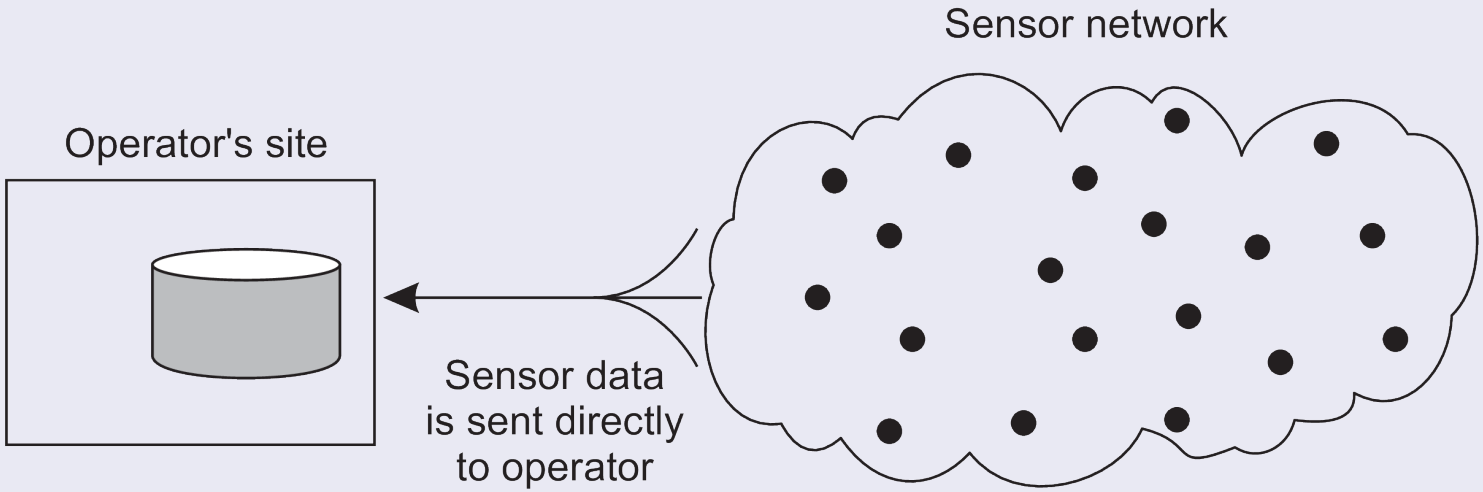
Sistemas Distribuídos Pervasivos

Redes de Sensores: Características

- Os nós aos quais os sensores estão presos são:
  - Muitos (10s–1000s)
  - Simples (pouca capacidade de memória/computação/comunicação)
  - Normalmente necessitam de uma bateria
- Frequentemente também são atuadores, além de sensores

Redes de Sensores: Organização

- Possibilidade 1:
  - Os sensores não cooperam. Simplesmente enviam os dados a uma base centralizada
  - Uso pesado e constante de rede



Redes de Sensores: Organização

- Possibilidade 2:
  - Requisições são enviadas aos sensores. Cada sensor computa uma resposta, cabendo ao operador agregá-las
  - Concentra a tarefa de agregação no operador