

ACH 2147 — Desenvolvimento de Sistemas de Informação Distribuídos

Aula 02: Introdução

Prof. Renan Alves

Escola de Artes, Ciências e Humanidades — EACH — USP

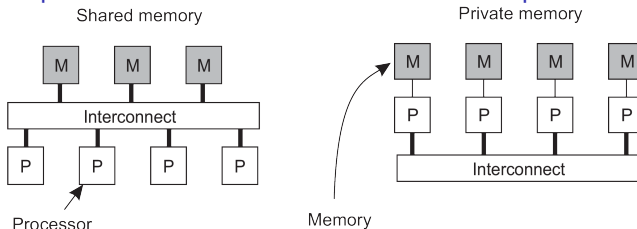
Uma classificação simples de sistemas distribuídos

Computação paralela

Observação

A computação distribuída de alto desempenho começou com a computação paralela.

Multiprocessador e multicore versus multicomputador



Sistemas distribuídos de memória compartilhada

Observação

Os multiprocessadores são relativamente fáceis de programar em comparação com os multicomputadores, mas têm problemas ao aumentar o número de processadores (ou núcleos). **Solução:** Tentar implementar um **modelo de memória compartilhada** em cima de um multicomputador.

Exemplo através de técnicas de memória virtual

Mapeie todas as páginas de memória principal (de diferentes processadores) em um **único espaço de endereço virtual**. Se um processo no processador *A* acessa uma página *P* localizada no processador *B*, o SO em *A* **captura e busca *P*** de *B*, da mesma forma como faria se *P* estivesse localizada no disco local.

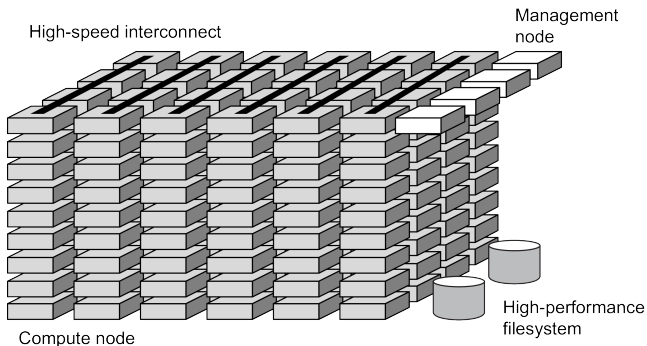
Problema

O desempenho da memória compartilhada distribuída nunca pôde competir com o dos multiprocessadores e não atendeu às expectativas dos programadores. Esta estratégia foi amplamente abandonada atualmente.

Computação em cluster

Basicamente um grupo de sistemas de alto nível conectados em LAN

- Homogêneo: mesmo SO, hardware quase idêntico
- Poucos nós de gerenciamento



Computação em grade (grid)

O próximo passo: muitos nós em vários lugares

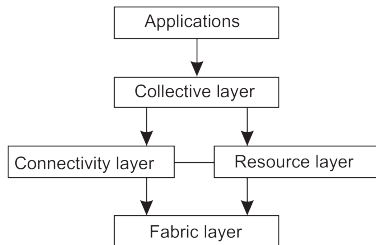
- Heterogêneo
- Disperso entre várias organizações
- Pode abranger uma rede de longa distância

Nota

Para permitir colaborações, os grids geralmente usam **organizações virtuais**. Em essência, este é um agrupamento de usuários (ou melhor: seus IDs) que permite controlar o acesso (autorização) durante a alocação de recursos.

Arquitetura para computação em grade (grid)

As camadas



- **Base:** Fornece interfaces para recursos locais (para consulta de estado e capacidades, travas, etc.)
- **Conectividade:** Protocolos de comunicação/transação, por exemplo, para mover dados entre recursos. Também inclui protocolos de autenticação.
- **Recurso:** Gerencia um único recurso, como criar processos ou ler dados.
- **Coletiva:** Lida com acesso a vários recursos: descoberta, agendamento, replicação.
- **Aplicação:** Contém as aplicações em si de uma dada organização.

Sistemas de informação distribuídos: Integrando aplicações

Situação

Organizações confrontadas com muitas **aplicações em rede**, mas alcançar interoperabilidade era doloroso.

Abordagem básica

Uma aplicação em rede é aquela que roda em um **servidor** tornando seus serviços disponíveis para **clientes** remotos. Integração simples: clientes combinam requisições para (diferentes) aplicações; enviam essas requisições; coletam as respostas e apresentam um resultado coerente ao usuário.

Próximo passo

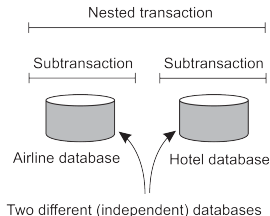
Permitir comunicação direta entre aplicações, levando à **Integração de Aplicações Empresariais**.

Exemplo Integração de Aplicações Empresariais: transações (aninhadas)

Transação

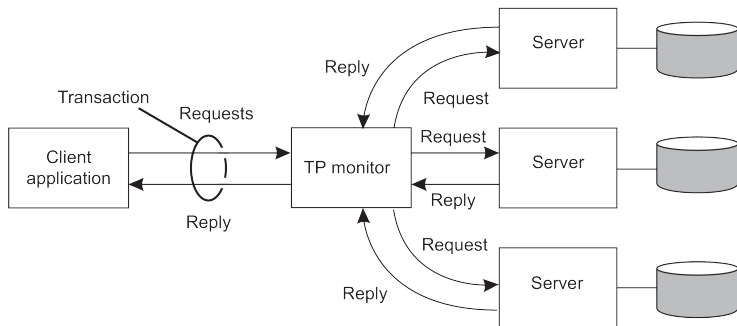
Primitiva	Descrição
<i>INICIAR_TRANSACAO</i>	Marca o início de uma transação
<i>TERMINAR_TRANSACAO</i>	Encerra a transação e tenta confirmar alterações
<i>ABORTAR_TRANSACAO</i>	Aborta a transação e restaura os valores antigos
<i>LER</i>	Lê dados (de um arquivo, uma tabela, etc)
<i>ESCREVER</i>	Escreve dados (em um arquivo, uma tabela, etc)

Questão: tudo-ou-nada



- **ACID:**
- **Atômico:** ocorre indivisivelmente (aparentemente)
- **Consistente:** não viola invariantes do sistema
- **Isolado:** sem interferência mútua
- **Durável:** confirmação significa que as mudanças são permanentes

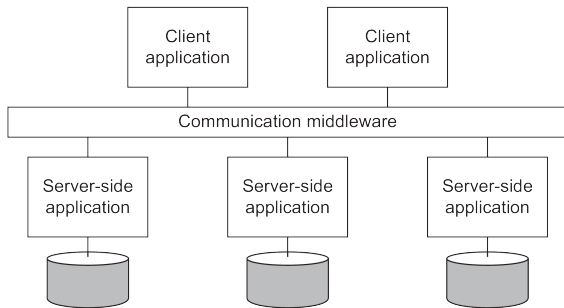
TPM: Transaction Processing Monitor (Monitor de Processamento de Transações)



Observação

Muitas vezes, os dados envolvidos em uma transação estão distribuídos em vários servidores. Um **Monitor de Processamento de Transações** é responsável por coordenar a execução de uma transação.

Middleware e Integração de Aplicações Empresariais



Middleware oferece facilidades de comunicação para integração

Chamada de Procedimento Remoto (RPC): Requisições são enviadas através de uma chamada de procedimento local, encapsuladas como mensagem, processadas, respondidas através de mensagem e o resultado retornado como retorno da chamada.

Middleware Orientado a Mensagens (MOM): Mensagens são enviadas para um ponto de contato lógico e encaminhadas para aplicações interessadas (modelo **produtor/consumidor**).

Como integrar aplicações

Transferência de arquivos: Tecnicamente simples, mas não flexível:

- Formato e o layout do arquivo
- Gerenciamento de arquivos
- Propagação e notificação de atualizações.

Banco de dados compartilhado: Muito mais flexível, mas ainda requer um esquema de dados comum ao lado do risco de ser um gargalo.

Chamada de procedimento remoto (RPC): Efetivo quando a execução de uma série de ações é necessária.

Orientado a mensagens: As chamadas de procedimento remoto requerem que o chamador e o chamado estejam em funcionamento ao mesmo tempo. Usar mensagens permite o desacoplamento no tempo e no espaço.

Sistemas pervasivos distribuídos

Observação

Próxima geração emergente de sistemas distribuídos em que os nós são pequenos, móveis e muitas vezes incorporados em um sistema maior, caracterizado pelo fato de que o sistema **se integra naturalmente ao ambiente do usuário**.

Três subtipos (sobrepostos)

Sistemas pervasivos distribuídos

Observação

Próxima geração emergente de sistemas distribuídos em que os nós são pequenos, móveis e muitas vezes incorporados em um sistema maior, caracterizado pelo fato de que o sistema **se integra naturalmente ao ambiente do usuário**.

Três subtipos (sobrepostos)

- **Sistemas de computação ubíqua**: pervasivos e **continuamente presentes**, ou seja, há uma interação contínua entre o sistema e o usuário.

Sistemas pervasivos distribuídos

Observação

Próxima geração emergente de sistemas distribuídos em que os nós são pequenos, móveis e muitas vezes incorporados em um sistema maior, caracterizado pelo fato de que o sistema **se integra naturalmente ao ambiente do usuário**.

Três subtipos (sobrepostos)

- **Sistemas de computação ubíqua**: pervasivos e **continuamente presentes**, ou seja, há uma interação contínua entre o sistema e o usuário.
- **Sistemas de computação móvel**: pervasivos, mas ênfase no fato de que os dispositivos são **inerentemente móveis**.

Sistemas pervasivos distribuídos

Observação

Próxima geração emergente de sistemas distribuídos em que os nós são pequenos, móveis e muitas vezes incorporados em um sistema maior, caracterizado pelo fato de que o sistema **se integra naturalmente ao ambiente do usuário**.

Três subtipos (sobrepostos)

- **Sistemas de computação ubíqua**: pervasivos e **continuamente presentes**, ou seja, há uma interação contínua entre o sistema e o usuário.
- **Sistemas de computação móvel**: pervasivos, mas ênfase no fato de que os dispositivos são **inerentemente móveis**.
- **Redes de sensores (e atuadores)**: pervasivas, com ênfase no **sensoriamento** e **atuação** colaborativas no ambiente.

Sistemas ubíquos

Elementos principais

1. (**Distribuição**) Dispositivos são conectados em rede, distribuídos e acessíveis de forma transparente
2. (**Interação**) A interação entre usuários e dispositivos é altamente discreta
3. (**Consciência de contexto**) O sistema está ciente do contexto do usuário para otimizar a interação (where, who, when, what)
4. (**Autonomia**) Dispositivos operam autonomamente sem intervenção humana e, portanto, são altamente autogerenciados
5. (**Inteligência**) O sistema como um todo pode lidar com uma ampla variedade de ações e interações dinâmicas

Computação móvel

Características distintivas

- Uma infinidade de diferentes dispositivos móveis (smartphones, tablets, dispositivos GPS, controles remotos).
- Móvel implica que a localização de um dispositivo é esperada mudar ao longo do tempo \Rightarrow mudança de serviços locais, acessibilidade, etc.
Palavra-chave: **descoberta**.
- Manter a comunicação estável pode introduzir desafios.
- Por muito tempo, a pesquisa se concentrou no compartilhamento direto de recursos entre dispositivos móveis. Nunca se tornou popular e agora é considerado um caminho de pesquisa infrutífero.

Computação móvel

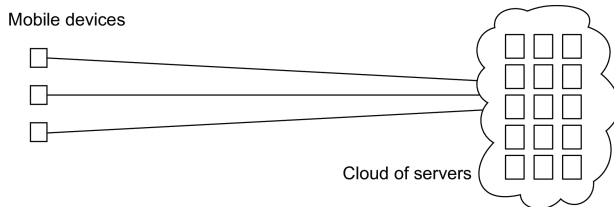
Características distintivas

- Uma infinidade de diferentes dispositivos móveis (smartphones, tablets, dispositivos GPS, controles remotos).
- Móvel implica que a localização de um dispositivo é esperada mudar ao longo do tempo \Rightarrow mudança de serviços locais, acessibilidade, etc.
Palavra-chave: **descoberta**.
- Manter a comunicação estável pode introduzir desafios.
- Por muito tempo, a pesquisa se concentrou no compartilhamento direto de recursos entre dispositivos móveis. Nunca se tornou popular e agora é considerado um caminho de pesquisa infrutífero.

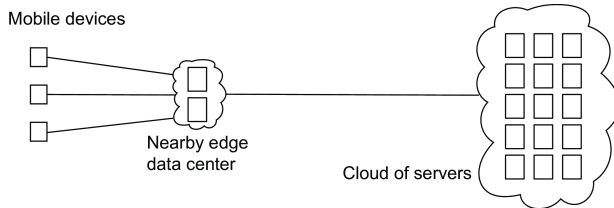
Resumindo

Dispositivos móveis estabelecem conexões com servidores estacionários, essencialmente transformando os usuários de computação móvel em clientes de serviços baseados em nuvem.

Computação móvel



Computação em nuvem móvel



Computação de borda móvel

Redes de sensores

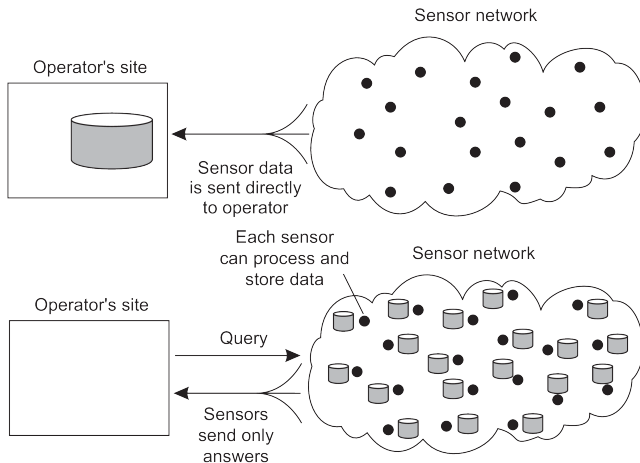
Características

Os nós aos quais sensores estão conectados são:

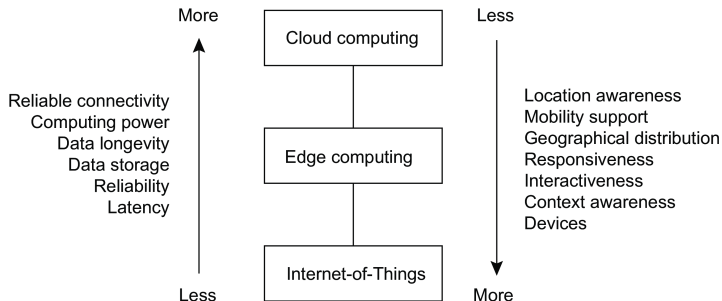
- Muitos (10s-1000s)
- Simples (pequena capacidade de memória/computação/comunicação)
- Muitas vezes alimentados por bateria (ou até mesmo sem bateria)

Redes de sensores como bancos de dados distribuídos

Dois extremos



O continuum nuvem-borda



Desenvolvendo sistemas distribuídos: Armadilhas

Observação

Muitos sistemas distribuídos acabam sendo desnecessariamente complexos, devido a erros que exigiram correções posteriormente. Muitas **falsas suposições** são frequentemente feitas.

Desenvolvendo sistemas distribuídos: Armadilhas

Observação

Muitos sistemas distribuídos acabam sendo desnecessariamente complexos, devido a erros que exigiram correções posteriormente. Muitas falsas suposições são frequentemente feitas.

Suposições falsas (e frequentemente ocultas)

Desenvolvendo sistemas distribuídos: Armadilhas

Observação

Muitos sistemas distribuídos acabam sendo desnecessariamente complexos, devido a erros que exigiram correções posteriormente. Muitas **falsas suposições** são frequentemente feitas.

Suposições falsas (e frequentemente ocultas)

- A rede é confiável

Desenvolvendo sistemas distribuídos: Armadilhas

Observação

Muitos sistemas distribuídos acabam sendo desnecessariamente complexos, devido a erros que exigiram correções posteriormente. Muitas **falsas suposições** são frequentemente feitas.

Suposições falsas (e frequentemente ocultas)

- A rede é confiável
- A rede é segura

Desenvolvendo sistemas distribuídos: Armadilhas

Observação

Muitos sistemas distribuídos acabam sendo desnecessariamente complexos, devido a erros que exigiram correções posteriormente. Muitas **falsas suposições** são frequentemente feitas.

Suposições falsas (e frequentemente ocultas)

- A rede é confiável
- A rede é segura
- A rede é homogênea

Desenvolvendo sistemas distribuídos: Armadilhas

Observação

Muitos sistemas distribuídos acabam sendo desnecessariamente complexos, devido a erros que exigiram correções posteriormente. Muitas **falsas suposições** são frequentemente feitas.

Suposições falsas (e frequentemente ocultas)

- A rede é confiável
- A rede é segura
- A rede é homogênea
- A topologia da rede não muda

Desenvolvendo sistemas distribuídos: Armadilhas

Observação

Muitos sistemas distribuídos acabam sendo desnecessariamente complexos, devido a erros que exigiram correções posteriormente. Muitas **falsas suposições** são frequentemente feitas.

Suposições falsas (e frequentemente ocultas)

- A rede é confiável
- A rede é segura
- A rede é homogênea
- A topologia da rede não muda
- A latência é zero

Desenvolvendo sistemas distribuídos: Armadilhas

Observação

Muitos sistemas distribuídos acabam sendo desnecessariamente complexos, devido a erros que exigiram correções posteriormente. Muitas **falsas suposições** são frequentemente feitas.

Suposições falsas (e frequentemente ocultas)

- A rede é confiável
- A rede é segura
- A rede é homogênea
- A topologia da rede não muda
- A latência é zero
- A largura de banda é infinita

Desenvolvendo sistemas distribuídos: Armadilhas

Observação

Muitos sistemas distribuídos acabam sendo desnecessariamente complexos, devido a erros que exigiram correções posteriormente. Muitas **falsas suposições** são frequentemente feitas.

Suposições falsas (e frequentemente ocultas)

- A rede é confiável
- A rede é segura
- A rede é homogênea
- A topologia da rede não muda
- A latência é zero
- A largura de banda é infinita
- O custo de transporte é zero

Desenvolvendo sistemas distribuídos: Armadilhas

Observação

Muitos sistemas distribuídos acabam sendo desnecessariamente complexos, devido a erros que exigiram correções posteriormente. Muitas **falsas suposições** são frequentemente feitas.

Suposições falsas (e frequentemente ocultas)

- A rede é confiável
- A rede é segura
- A rede é homogênea
- A topologia da rede não muda
- A latência é zero
- A largura de banda é infinita
- O custo de transporte é zero
- Há um único administrador