

Sistemas Distribuídos

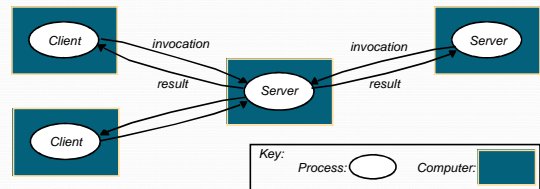
3 Arquiteturas de Sistemas Distribuídos

- **Cliente-servidor**
- **Código móvel**
- **Agentes de software**
- **Processos pares**

Profª Ana Cristina B. Kochem Vendramin
DAINF / UTFPR

Arquiteturas em Sistemas Distribuídos

• Cliente-Servidor

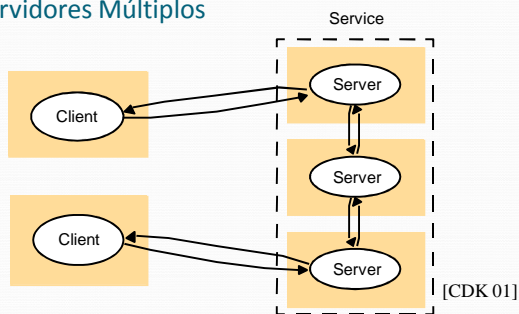


[CDK 01]

Profª Ana Cristina B. Kochem Vendramin,
DAINF/UTFPR

2

Servidores Múltiplos

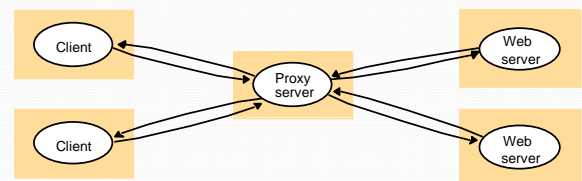


[CDK 01]

Profª Ana Cristina B. Kochem Vendramin,
DAINF/UTFPR

3

Servidor de Proxy Web



[CDK 01]

Profª Ana Cristina B. Kochem Vendramin,
DAINF/UTFPR

4

Código Móvel

- Código que pode ser enviado de um computador para outro e ser executado remotamente no destino;
 - Exemplo: Java Applets
- Diferentes máquinas = Diferentes instruções
- Máquina virtual:
 - Fornece uma maneira de executar um código em qualquer máquina.
 - O compilador para uma linguagem em particular gera código para uma máquina virtual ao invés de um código para um hardware particular.

Profª Ana Cristina B. Kochem Vendramin,
DAINF/UTFPR

5

Código Móvel

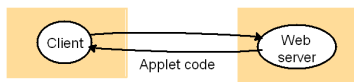
- Vantagem:
 - Código executado localmente fornece melhor tempo de resposta uma vez que não sofre atrasos ou variações de largura de banda associadas com a comunicação na rede.
- Desvantagem:
 - Ameaça à segurança dos recursos no computador destino.

Profª Ana Cristina B. Kochem Vendramin,
DAINF/UTFPR

6

Código Móvel

- O pedido do cliente resulta no *download* do código móvel

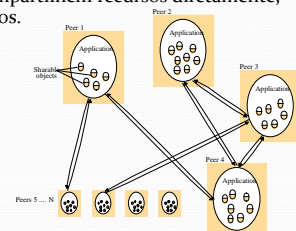


- O processo cliente interage com o código móvel



Redes Peer-to-Peer (P2P)

- São sistemas distribuídos nos quais os membros da rede são equivalentes em funcionalidade;
- Permitem que os pares compartilhem recursos diretamente, sem envolver intermediários.



Características

- Auto-organização:** não há um coordenador do grupo; toda a coordenação é distribuída;
- Adaptabilidade:** rede se ajusta ao ambiente, mesmo que ocorram falhas;
- Escalabilidade:** rede cresce em escala facilmente; não há ponto de estrangulamento;
- Comunicação direta entre os pares:** se opõe ao tradicional modelo cliente-servidor, já que cada nó pode fornecer ou obter recursos.

Utilização

- Compartilhamento de arquivos, imagens, músicas, vídeos, etc.
- Atualização de sistemas operacionais e de software aplicativo;
- Gerenciamento de redes e sistemas;
- Processamento distribuído;
- Sincronização de bancos de dados;
- Difusão de informações, entre outros.

Arquitetura

- Arquitetura Centralizada:** utiliza um servidor central para controle de acesso à rede e para publicação e pesquisa de conteúdo;
- Arquitetura Descentralizada:** todos os *peers* (pares) possuem funcionalidade equivalente;
- Arquitetura Híbrida:** alguns pares especiais, chamados supernós, possuem um papel diferenciado na rede.

Arquitetura Centralizada

- Um servidor central controla as entradas e saídas de pares da rede;
- Os pares registram no servidor central os recursos que compartilharão na rede;
- Pesquisas por recursos disponíveis nos pares são efetuadas pelo servidor central;
- O acesso aos recursos é feito diretamente entre pares.
- Exemplos: **Napster**, **eMule**

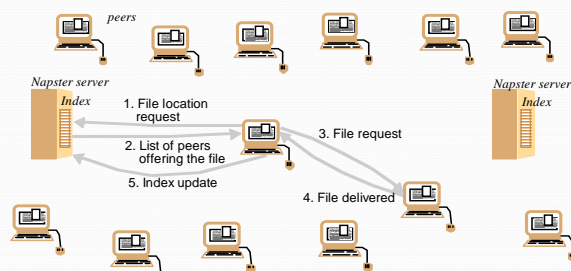
Arquitetura Híbrida

- Supernós permitem o ingresso dos nós na rede, podendo também exercer atividades de coordenação da rede, indexar os recursos compartilhados pelos nós e permitir a busca por estes recursos;
- Após localizado, um recurso pode ser obtido a partir da interação direta entre nós;
- Falha de um supernó pode ser tolerada elegendo dinamicamente outro supernó;
- Exemplos: **Kazaa**, **Skype**, **BitTorrent**.

Arquitetura Descentralizada

- Não há um elemento central;
- Todos os nós possuem papel equivalente;
- As pesquisas por recursos compartilhados são feitas por inundação (*flooding*);
- Gera um alto tráfego na rede;
- O desempenho das pesquisas é ruim devido à necessidade de contatar muitos nós e aguardar a resposta.
- Exemplos: **Gnutella** e **JXTA**

Exemplo de Redes P2P – Napster



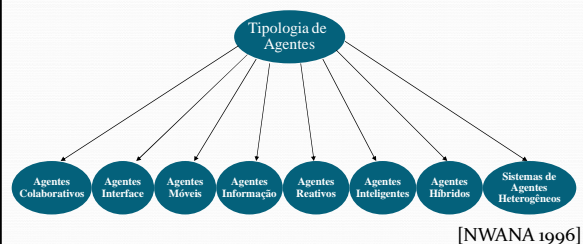
Agentes de Software

- O que difere um agente de um software convencional é:
 - Agentes são **autônomos**
 - Podem operar sem controle direto de humanos ou de outros agentes.
 - São capazes de se **comunicar** com outros agentes e humanos;
 - São **reativos** podendo reagir a mudanças no seu ambiente;
 - São **pró-ativos** podendo por si mesmo tomar decisões em busca de uma determinada meta.

Agentes de Software

- Um agente é um software que possui uma coleção de regras a verificar e ações a serem tomadas segundo uma lógica de execução.
- Um agente pode funcionar “*stand-alone*” ou em uma federação de agentes.
- Um agente pode ser estático ou móvel.

Classificação de Agentes



Classificação de Agentes

• Agentes Colaborativos

- Enfatizam **autonomia** e **cooperação** (com outros agentes) para realizar tarefas em nome de alguém;
- Solucionam problemas que são complexos para um único agente devido a limitações de recursos;
- Solucionam problemas distribuídos (ex. controle de tráfego aéreo, redes de sensores distribuídas);
- Melhoram modularidade (reduz complexidade), velocidade (devido ao paralelismo), confiabilidade (redundância), etc.

Classificação de Agentes

• Agentes de Informação ou Internet

- Ajudam a buscar, analisar e recuperar a grande quantidade de informações disponíveis nas fontes distribuídas em redes como a Internet.

Classificação de Agentes

• Agentes de Interface

- Enfatizam a **autonomia** e o **aprendizado** para realizar tarefas;
- Adquire conhecimento por meio das seguintes técnicas de aprendizagem:
 - Observando e imitando;
 - Recebendo informações explícitas dos usuários;
 - Pedindo conselhos a outros agentes.

Classificação de Agentes

• Agentes de Interface

- Usado para implementar assistentes pessoais;
- Ex.: agente calendário que auxilia o usuário no gerenciamento de reuniões (aceitar, rejeitar e agendar).
- Auxiliar um usuário no aprendizado de uma determinada aplicação. O agente observa e monitora as ações dos usuários na interface, aprende novos atalhos e sugere melhores maneiras de executar uma tarefa.

Classificação de Agentes

• Agentes de Interface

- Auxiliares de memorização
 - Exemplo: ao escrever um artigo o agente pode, através de palavras-chave digitadas pelo usuário, recomendar outros pesquisadores que deveriam ser consultados.

Classificação de Agentes

• Agentes Móveis

- **Mobilidade** - programas que percorrem uma rede, interagindo com máquinas externas, realizando uma tarefa em nome de seu usuário.
- Podem fazer invocações aos recursos locais em cada *site* que visitam.
- Exemplos de uso:
 - Instalação de software em computadores de uma organização;
 - Pesquisa de preços de produtos de vendedores visitando o *site* de cada um e executando operações em base de dados.

Classificação de Agentes

• Agentes Móveis

- Vantagens:
 - Redução do custo de comunicação;
 - Substituição de chamadas remotas por locais.
- Desvantagens:
 - Ameaça à segurança dos recursos dos computadores que um agente móvel visita (vírus, consumo de CPU, etc.)
 - O ambiente que recebe o agente deve decidir quais recursos locais podem ser acessados com base na identidade do usuário (autenticação) para qual o agente está atuando.

Classificação de Agentes

• Agentes Reativos

- Modelo de funcionamento formado pelo par: **Estímulo-Resposta** (Ação-Reação)
- Não há uma representação interna explícita do ambiente onde residem.
- O comportamento (resposta) de cada agente é baseado no que ele percebe (estímulo) no seu ambiente a cada instante.
- Exemplos: jogos

Classificação de Agentes

• Agentes Inteligentes

- Dispõe de algum grau de inteligência artificial;
- Capacidade de **raciocinar** sobre os elementos percebidos e escolher a melhor ação segundo as circunstâncias externas a ele.

• Agente Híbrido

- Coleções de duas ou mais filosofias de agentes em um único agente.

• Sistemas de Agentes Heterogêneos

- Integração de dois ou mais agentes que pertencem a duas ou mais classes diferentes de agentes → federação de agentes.
- Pode conter um ou mais agentes híbridos.

Referências

- Coulouris, George; Dollimore, Jean; Kindberg, Tim. Distributed Systems: concepts and design. Third Edition. Addison-Wesley 2001.
- Coulouris, George; Dollimore, Jean; Kindberg, Tim; tradução João Tortello. Sistemas Distribuídos: conceitos e projeto. 4. ed. Bookman 2007.
- Nwana, H.S. (1996). Software Agents: An Overview. *Knowledge Engineering Review*, Vol.11, n.3, 1-40. Cambridge University Press. Disponível em: <http://tcc.itc.it/people/valitutti/home/archivio/collegato/tesi/stesura/archivio/Nwana1996.htm>.