



Agentes Móveis

■ Mobilidade

- propriedade ortogonal de um agente
- nem todos os agentes necessitam ser móveis

■ Agentes Estacionários

- um agente pode simplesmente permanecer em seu site original e se comunicar com o resto do mundo por meio da rede
- um agente estacionário executa somente no sistema onde ele foi originalmente executado
- se um agente estacionário necessita de informações que não estão no sistema, ele utiliza algum mecanismo de comunicação para acessar outros agentes ou recursos
- agentes estacionários podem ser extremamente úteis e poderosos - a falta de mobilidade não é necessariamente uma fraqueza do agente



Agentes Móveis

■ Agente Móvel

- não está ligado ao sistema que o iniciou
- livre para viajar por entre hosts em uma rede
- uma vez criado em uma máquina/sistema hospedeiros, ele pode transportar seu estado e código consigo para outras máquinas/sistemas na rede, onde continua sua execução

■ Estado

- valores dos atributos do agente necessários para que ele continue sua execução após o transporte

■ Código

- dentro de um contexto orientado a objetos, o código das classes necessárias para que o agente possa executar



Agentes Móveis

■ Definição de um Agente Móvel

- um agente móvel é um agente que não está ligado permanentemente ao sistema que o iniciou enquanto processo
- ele possui a habilidade única de se transportar de um sistema a outro em uma rede, mantendo seu contexto original antes do transporte
- esta habilidade para viajar, permite ao agente móvel se mover para o sistema que contém o recurso com o qual o agente deseja interagir
- pode portanto tirar vantagem disso para estar na mesma máquina ou rede onde busca informações, e com isso ser muito mais eficiente em sua tarefa



Boas Razões para Usar Agentes Móveis

- Agentes Móveis reduzem o tráfego na rede
 - sistemas distribuídos demandam protocolos de comunicação que envolvem múltiplas interações para concluir uma tarefa
 - isso é especialmente verdadeiro quando medidas de segurança são utilizadas
 - o resultado disso é um tráfego intenso na rede
 - agentes móveis permitem o deslocamento da interação entre a aplicação cliente e o servidor para a máquina onde está o servidor, onde estas interações podem então ocorrer localmente
 - quando grandes volumes de dados estão armazenados remotamente e necessitam ser processados, ao invés de trazer os dados para serem processados localmente, enviamos o agente para processá-los remotamente



Boas Razões para Usar Agentes Móveis

- Agentes podem resolver problemas de latência na rede
 - sistemas críticos de tempo-real (tais como robôs e processos de manufatura) precisam responder imediatamente a mudanças em seu ambiente
 - controlar estes sistemas por meio de redes em uma fábrica de tamanho substancial pode introduzir tempos de latência significativos
 - para sistemas de tempo real críticos, essas latências podem não ser aceitáveis
 - agentes móveis podem solucionar o problema, pois estes podem ser despachados de um controlador central para agir localmente e executar diretamente as diretivas de controle



Boas Razões para Usar Agentes Móveis

- Agentes podem encapsular protocolos
 - quando trocamos dados em um sistema distribuído, cada host implementa localmente os protocolos de comunicação com o qual pode receber e enviar dados
 - entretanto, como estes protocolos estão sendo sempre aperfeiçoados e/ou modificados por razões de segurança e eficiência, os softwares que implementam estes protocolos necessitam estar em constante manutenção
 - em uma rede com um número grande máquinas, a manutenção destes softwares pode ser custosa e até mesmo impossível
 - protocolos normalmente tornam-se uma sina de sistemas mais antigos
 - agentes móveis podem carregar consigo estes protocolos, e portanto podem implementar esta manutenção de uma maneira muito eficaz



Boas Razões para Usar Agentes Móveis

- Agentes Móveis executam de maneira assíncrona e autônoma
 - frequentemente, dispositivos móveis necessitam dispor de conexões de rede caras e frágeis
 - tarefas que demandam de conexões continuamente abertas entre um dispositivo móvel e uma rede fixa podem não ser economicamente viáveis
 - estas tarefas podem ser implementadas por meio de agentes móveis, que podem então ser despachados para dentro da rede
 - após serem despachados, os agentes móveis tornam-se independentes do processo que os criou e podem atuar de maneira assíncrona e autônoma
 - o dispositivo móvel pode se reconectar posteriormente então para coletar o agente de volta



Boas Razões para Usar Agentes Móveis

- Agentes Móveis podem adaptar-se dinamicamente
 - agentes móveis têm a habilidade de sensorar seu ambiente de execução e reagir autonomamente a mudanças
 - estas mudanças podem envolver questões como:
 - | taxa de ocupação de memória e CPU de máquinas específicas
 - | tráfego na rede
 - | ociosidade de máquinas
 - | trechos e/ou segmentos de rede congestionados
 - | horários prioritários em máquinas específicas
 - múltiplos agentes móveis possuem a habilidade única de se distribuírem mutuamente entre os hosts disponíveis em uma rede, de maneira a manter uma configuração ótima para a resolução de um dado problema



Boas Razões para Usar Agentes Móveis

- Agentes Móveis são naturalmente heterogêneos
 - redes são sistemas fundamentalmente heterogêneos, tanto sob perspectivas de hardware como de software
 - múltiplos tipos de máquinas e sistemas operacionais podem estar presentes em redes de vasto alcance
 - promover a integração de máquinas e sistemas diferentes pode constituir uma tarefa razoavelmente complicada
 - esta integração normalmente demanda soluções específicas para cada máquina/sistema
 - como agentes móveis geralmente não dependem do computador ou da camada de transporte que os conecta, mas somente de seu ambiente de execução, eles provêm condições ótimas para a integração de sistemas



Boas Razões para Usar Agentes Móveis

- Agentes Móveis são Robustos e Tolerantes a Falhas
 - agentes móveis podem reagir dinamicamente diante de situações desfavoráveis e eventos imprevistos
 - esta habilidade dos agentes móveis facilita a construção de sistemas distribuídos mais robustos e tolerantes a falhas
 - se um determinado host passa a apresentar problemas e necessita ser desligado, todos os agentes executando nesta máquina são avisados e têm um dado tempo para transferirem-se para outro host
 - no novo host, podem então continuar sua operação normal, sem prejuízo para o cumprimento de suas tarefas
 - caso seja impossível essa transferência, grupos de agentes que comunicam-se constantemente podem notar a ausência de comunicações do agente subitamente desaparecido e tomar providências para compensar sua ausência



Um Novo Paradigma de Programação em Rede

■ Paradigmas de Programação em Rede

- Paradigma Cliente-Servidor
- Paradigma Código-sob-Demanda
- Paradigma Agentes Móveis

■ Agentes Móveis

- provêm um paradigma poderoso e uniforme para programação em rede
- podem revolucionar o design e desenvolvimento de sistemas distribuídos
- podem ser mais simples de conceber
- podem ser mais simples de se testar
- podem gerar comportamentos mais sofisticados que os sistemas distribuídos convencionais



Paradigmas de Programação em Rede

■ Paradigma Cliente-Servidor

- servidor oferece um conjunto de serviços que provêm acesso a um conjunto de recursos (por exemplo bases de dados)
- o código que implementa estes serviços está normalmente localizado no servidor (o servidor tem o know-how)
- o servidor por si mesmo executa o serviço e portanto tem a capacidade de processamento
- se o cliente está interessado em acessar algum recurso, ele simplesmente usa um dos serviços oferecidos pelo servidor
- o cliente precisa saber quais os serviços disponíveis e como acessá-los
- este paradigma está associado a uma vasta gama de tecnologias:
 - | RPC, CORBA, RMI



Paradigmas de Programação em Rede

■ Paradigma Código-sob-Demanda

- O cliente obtém o know-how quando o necessita
- um determinado host A deseja executar uma tarefa, mas não possui o código que a implementa (know-how)
- outro host B acessável via rede possui o código desejado
- o código é transmitido do host B para o host A
- o host A pode então executar o código
- Ao contrário do paradigma cliente-servidor, o host A não precisa de maiores dados de B sobre o serviço, uma vez que o código necessário será obtido via download
- Exemplos deste paradigma são:
 - Java applets e servlets, controles ActiveX



Paradigmas de Programação em Rede

■ Paradigma Agentes Móveis

- característica chave é que qualquer host da rede pode agregar qualquer mistura dentre know-how, recursos e processamento
- capacidade de processamento dos agentes móveis pode então ser combinada a recursos locais, sendo transportada de host a host na rede
- o know-how (na forma de agentes móveis), não está vinculado a nenhum host específico, mas está disponível a partir da rede
- comparando-se aos outros paradigmas, percebe-se a grande flexibilidade do paradigma agentes móveis:
 - l clientes e servidores foram unificados na idéia de "host"
 - l applets e servlets são combinados e têm suas características extendidas e aperfeiçoadas



Aplicações de Agentes Móveis

- **Comércio Eletrônico**
 - agentes podem vasculhar a rede e negociar em nome de quem os enviou
- **Assistente Pessoal**
 - assistentes podem efetuar tarefas na rede em nome de seus criadores. Estes podem desligar seus micros da rede
- **Negociação Segura**
 - agentes podem mover-se para sites seguros a fim de negociar
- **Busca de Informação Distribuída**
 - ao invés de concentrar busca em um só site, distribui agentes pela rede
- **Processamento Paralelo**
 - agentes podem gerenciar processos paralelos



Aplicações de Agentes Móveis

- **Serviços de Telecomunicação via Rede**
 - agentes podem reconfigurar dinamicamente redes de telecomunicações, mantendo-a coesa
- **Aplicações de Workflow e Groupware**
 - agentes móveis podem cooperar em nome de co-participantes de um grupo de trabalho
- **Monitoração e Notificação**
 - agente pode monitorar uma determinada fonte de informações de maneira independente de onde se originou
- **Disseminação de Informação**
 - agentes vêm até você com a informação desejada (tecnologia push - aplicada a agentes móveis)



Plataformas de Agentes Móveis em Java

■ Aglets

- desenvolvido pela IBM
- inspirado no modelo de applets: aglet = agent + applet

■ Odissey

- desenvolvido pela General Magic Inc.
- descendente do Telescript, primeiro sistema de agentes móveis comercial

■ Concordia

- desenvolvido pela Mitsubishi

■ Voyager

- desenvolvido pela ObjectSpace
- ORB completo, com capacidades para agentes móveis



Outras Plataformas de Agentes Móveis

■ Agent TCL

- desenvolvido no Dartmouth College
- sistema de agentes móveis desenvolvido para a linguagem TCL
- provê extensivos serviços de navegação e comunicação, mecanismos de segurança e debug, além de ferramentas de rastreamento de agentes

■ Ara

- desenvolvido na Universidade de Kaiserslautern
- desenvolvido para TCL, em redes heterogêneas

■ TACOMA

- projeto focaliza no desenvolvimento de suporte do sistema operacional a agentes móveis
- baseado em UNIX e TCP, suportando C, TCL/TK, Perl, Python e Scheme



Padronização de Agentes Móveis - MASIF

- Mobile Agent System Interoperability Facility (MASIF)
 - desenvolvido pelas empresas: Crystaliz, General Magic, GMD Fokus, IBM Corporation e Open Group
 - submetido ao Object Management Group (OMG)
- Foco
 - interface entre sistemas de agentes (não entre aplicações de agentes e sistemas de agentes)
- Tópicos Abordados
 - Gerenciamento de Agentes
 - Transferência de Agentes
 - Nomes de Agentes e de Sistemas de Agentes
 - Tipos de Sistemas de Agentes e Sintaxe de Localização