

P. A disponibilidade generalizada de serviços e a apetência demonstrada pelos sistemas computacionais paralelos para a execução cooperativa de aplicações levantam a questão de se saber quão seguros são os fluxos de informação trocados e quão segura é a informação armazenada nos recursos associados. Indique os diferentes níveis em que se pode colocar a segurança de informação e descrever medidas que podem ser usadas para reduzir os riscos.

R.: É política mais adequada considerar que a segurança não se baseia na segurança da rede ou dos sistemas e admitir um princípio de suspeição mútua em relação a todas as entidades. Técnicas fundamentais para garantir a segurança num ambiente distribuído: • Canais de comunicação seguros; • Autenticação fidedigna dos agentes; • Controlo de acesso (Autorização) no acesso aos objectos com base na identidade do agente remoto e nos direitos de acesso; • Autenticação de informação transmitida; • Integridade da informação transmitida. A segurança da informação pode ser vista em 3 níveis: • a confidencialidade – que é a protecção contra a sua divulgação a entidades não autorizadas; • a integridade – que é a protecção contra a sua alteração ou corrupção; • a disponibilidade - protecção contra interferências que perturbem os mecanismos de acesso. Podem ser tomadas algumas medidas de modo a reduzir os riscos da falta de segurança: • a introdução de firewalls que criam uma barreira, envolvendo uma rede, limitando o tráfego de entrada e de saída a tipos conhecidos e definidos; • encriptação das mensagens que mascara o conteúdo dos fluxos de informação; • utilização de assinaturas electrónicas de modo a certificar o autor da mensagem.

P. Explique o que é um sistema distribuído e indique dois dos aspectos que têm que ser ponderados na sua concepção.

R.: Um sistema [de operação] distribuído é um sistema de operação cujos componentes, que estão localizados nos diferentes elementos de um sistema computacional paralelo, coordenam e comunicam normalmente as suas acções por passagem de mensagens. A motivação principal que leva à construção e à utilização destes sistemas, é a partilha de recursos. A concepção destes sistemas distribuídos, e das aplicações executadas sobre eles, envolve algumas questões: • grau de abertura que o sistema apresenta; • segurança dos fluxos de informação trocados e da informação armazenada; • potencial apresentado de modo a aumentar a escala; • tratamento de falhas; • sincronização e a exclusão mútua no acesso a regiões críticas; • um grau de transparência; • heterogeneidade dos componentes que formam o sistema computacional paralelo;

P. Caracterize o modelo cliente-servidor e indique qual é o papel aí desempenhado por um servidor de proxie.

R.: O modelo cliente-servidor é constituído por um conjunto de PCs, possuindo o utilizador um PC que mantém os dados e um servidor que possui as aplicações. A comunicação neste modelo é feita sob a forma de um pedido do cliente ao servidor requerendo a execução duma tarefa. Por sua vez, o servidor executa a tarefa e responde ao cliente, enviando os resultados requeridos. – Os serviços são acedidos via uma interface bem definida; – A interface de um servidor é partilhada por todos os clientes; – Os termos cliente e servidor correspondem aos papéis desempenhados numa relação particular. – Uma entidade pode ser ao mesmo tempo cliente numa dada

relação e servidor noutra. O proxy é um intermediário na interacção com o servidor. Um servidor de proxies, por sua vez tem a função de analisar o pedido do cliente e, dependendo das permissões aí existentes, o servidor irá executar o pedido se este tiver permissão. – A aplicação mais comum é como cache permitindo diminuir a carga na rede e nos servidores. Também usado em firewalls.

Em que consiste o método de invocação remota? Será que dentro deste modelo é possível a migração de código? Como?

R.: O método de invocação remota permite que haja interacção entre programas, que estão a ser executados em diferentes nós de um sistema distribuído, através da invocação de procedimentos. Esta invocação de procedimentos remotos permite que sejam acedidos serviços remotos. Neste modelo é possível a migração de código. O objecto remoto é instanciado e registado. Utilizando o marshalling o código é copiado para o servidor e é feito um unmarshalling de modo a que o código seja executado.

O modelo cliente-servidor e o modelo de comunicação entre pares são dois modelos conceptuais de organização da interacção em sistemas distribuídos. O que os distingue? Quais são as vantagens e os inconvenientes de cada um?

R.: No modelo cliente-servidor os servidores (máquinas potentes ou processos que fazem a gestão de discos rígidos (file servers), impressoras (printer servers) ou de tráfego de rede (network servers)) oferecem serviços aos utilizadores e os clientes requerem os serviços. Os serviços são acedidos via uma interface bem definida. A interface de um servidor é partilhada por todos os clientes. Os clientes são PCs ou workstations nos quais os utilizadores correm aplicações – os clientes servem-se dos recursos dos servidores, tais como

ficheiros, dispositivos, ou mesmo poder de processamento. No modelo de comunicação entre pares todos os elementos da rede têm capacidades e responsabilidades equivalentes interagindo de forma cooperativa e sem distinção entre clientes e servidores. Os padrões de interacção podem variar em função das circunstâncias ou da tarefa. No modelo cliente-servidor há computadores que estão dedicados a servir outros (os clientes). As redes peer-to-peer são geralmente mais simples, mas não oferecem o mesmo desempenho face a sobrecargas.

Quais são as diferenças principais entres os protocolos TCP e UDP?

R.: O protocolo TCP assegura que os blocos de dados são transmitidos correctamente. O canal é estabelecido antes da fase de comunicação, onde há uma ligação prévia, que garante a entrega e preserva a sequência de dados. O UDP é um protocolo sem conexão que não garante a entrega da informação, preservação da sequência ou protecção contra duplicação. No UDP o canal é conceptualmente estabelecido apenas o envio de uma mensagem. Não garante entrega, não notifica em caso de falhas e não preserva a sequência de envio de pacotes. O protocolo TCP é orientado à ligação enquanto que o UDP não. O TCP/IP usa o modelo cliente/servidor, onde uma aplicação cliente pede um serviço a uma aplicação servidor (a aplicação cliente normalmente encontra-se num computador distinto da aplicação servidor). A comunicação TCP/IP é ponto-a-ponto, o que significa que quando o cliente e servidor estão em computadores distintos as mensagens envolveram apenas esses dois computadores.

A serialização de objectos em JAVA constitui uma abordagem possível ao problema da comunicação de dados em ambientes distribuídos. Explique neste contexto em que consiste o marshalling e o unmarshalling de informação.

R.: Na serialização de objectos em JAVA um objecto é uma instância de uma classe que possui um estado. O marshalling e o unmarshalling são usados de modo a “serializar” um objecto que é transformado numa sequência de bytes. A informação usada pelos programas encontra-se em estruturas de dados cuja manipulação é facilitada pelos mecanismos das linguagens de programação. De modo a que a informação seja transferida na rede, esta necessita de ser convertida numa sequência de bytes e depois reconstruída ao chegar ao destino. O marshalling é o processo de transformação de um conjunto de dados numa sequência de bytes adequada para a transmissão (flattening). O unmarshalling é o processo de transformação de uma sequência de bytes em estruturas de dados equivalentes às que foram enviadas.

Um atributo importante de um sistema distribuído é a sua transparência. Explique em que consiste e, neste contexto, distinga transparência de acesso de transparência de localização.

R.: O grau de transparência de um sistema computacional é uma característica que exprime o menor ou maior sucesso que foi conseguido a “mascarar” a complexidade subjacente. A funcionalidade da transparência é descrita de uma forma integrada, conceptualmente simples, em vez de resultar da interacção de um conjunto de componentes independentes. A transparência tem como objectivos esconder os recursos que não são directamente relevantes para a tarefa em curso, tornando-os anónimos ao utilizador e/ou ao programador de aplicações. Então, o grau de transparência com que os diferentes recursos podem ser manipulados e acedidos, reflecte directamente o grau de abstracção e de operacionalidade apresentados pela camada “middleware” do sistema em causa. Existem algumas formas de transparência: •transparência no acesso -quando se efectuam o mesmo tipo de operações para acesso a recursos locais ou remotos; •transparência de posição -quando se realiza o acesso aos recursos sem necessidade de conhecimento da sua localização precisa; •transparência de rede -quando existe simultaneamente transparência no acesso e de posição; •transparência de concorrência -quando o acesso a recursos partilhados se faz sem que ocorram interferências (a informação permanece consistente); •transparência de replicação -quando é possível instanciar múltiplas réplicas dos recursos sem que tal se torne notório;

Uma maneira de se efectuar a localização remota de objectos dentro do modelo de invocação remota de métodos (RMI) é através do rmiregistry. Explique em que consiste e descreva as operações que têm que ser realizadas pelo servidor e pelo cliente para que tal seja possível.

R.: O rmiregistry permite guardar e obter informação sobre objectos remotos existentes numa máquina. Para se poderem distinguir os objectos, é usado um URL com o seguinte formato: rmi://host:port/name. Cada objecto remoto regista-se no RMI Registry da máquina onde se encontra. Um objecto que deseje invocar os métodos de um objecto remoto pode entrar em contacto com o RMI Registry da máquina onde se encontra o objecto remoto indicando o nome do objecto pretendido e obter a respectiva referência.

Vantagens e desvantagens de um SD.

R.: **VANTAGENS**

Economia: aproveitar recursos ociosos; é mais barato ter vários processadores inter conectados do que um super computador; **Distribuição inerente:** algumas aplicações são distribuídas por natureza; **Tolerância a falhas:** em caso de falha de uma máquina, o sistema pode sobreviver, mesmo com desempenho degradado; **Crescimento incremental:** o poder computacional pode ser aumentado através da inclusão de novos equipamentos; **Flexibilidade:** Maior flexibilidade na alocação dos recursos, permitindo que usuários compartilhem dados, processamento e dispositivos.

DESVANTAGENS

Aplicações mais complexas: Pouco software de alto nível disponível para sistemas distribuídos; **Segurança:** Necessidade de construir mecanismos para controle de acesso às informações; **Dependência da rede;** **Falhas:** Capacidade de tráfego insuficiente

P. Explique como é que o RMI (Remote Method of Invocation) em Java lida com a implementação da transparência de acesso e da transparência de localização. Que limitações apresenta?

R.: RMI (Remote Method Invocation): • Estende o modelo de programação baseado em objectos de forma a suportar a invocação de métodos em objectos de outros processos. – Transparência na invocação de métodos locais/remotos; – Transparência total não é possível nem desejável; • Um objecto capaz de receber invocações remotas designa-se por objecto remoto: – Um objecto remoto implementa um determinado interface remoto; – Um interface remoto define a assinatura dos métodos que podem ser invocados remotamente. Referências para objectos remotos: – Invocar um objecto implica ter uma referência para esse objecto; – Invocação remota implica referências para objectos remotos; – Referências não são apontadores; – Identifica univocamente um objecto no âmbito de um sistemas distribuído; – Referências para objectos remotos podem ser passadas como argumentos em métodos remotos; – Objectos que invocam métodos remotos não precisam de distinguir objectos locais de remotos (Transparência de acesso).

P. O método de invocação remota em Java possibilita a migração de código entre sistemas computacionais. Explique como é que isso é conseguido.

R.: Isso é conseguido com o uso do modelo “Código Móvel”– Variação do modelo cliente-servidor em que o cliente começa por transferir de um servidor o código que depois irá utilizar para aceder às funcionalidades do serviço. Após a transferência, o cliente efectua os pedidos directamente no código transferido. – Exemplo: Java applets, jiniApplets: • Uma das potencialidades da linguagem Java é a criação de Applets – pequenos programas que podem correr num Browser. • A sua programação é algo restritiva devido às limitações impostas pelo sistema de execução – Sandbox environment.

P. Que papel desempenha o firewall no acesso a uma rede local?

R.: Uma firewall tem o objectivo de implementar segurança numa rede. Tem o papel de impedir recursos internos a utilizadores não autorizados; impede que internamente se acedam a certos sites. Tem o objectivo de criar uma barreira, em torno de uma rede parcelar, que limita o tráfego de entrada e de saída a tipos bem definidos.

P. Qual é o papel desempenhado por um firewall numa arquitectura distribuída que disponibiliza um conjunto de serviços na Internet? Faça um diagrama que ilustre a sua localização.

R.:

P. Descreva o modelo de programação providenciado pelo protocolo TCP, quer do lado do servidor, quer do lado do cliente.

R.:

P. Uma forma de implementação do modelo cliente-servidor é efectuar a replicação do servidor sempre que há uma solicitação de um novo cliente. Mostre que vantagens e inconvenientes é que esta arquitectura apresenta.

R.:

P. O modelo de invocação remota de métodos (RMI) constitui um modelo de comunicação com uma abstracção mais elevada do que o modelo puro de passagem de mensagens implementado com sockets. Mostre porquê.