Engenharia de Telecomunicações
STD29006 – Sistemas Distribuídos

Prof. Emerson Ribeiro de Mello
http://docente.ifsc.edu.br/mello/std

Lista de exercícios 2

Francin Barcelos

23/02/2022

1. Por qual motivo você projetaria um sistema distribuído? Liste algumas vantagens dos sistemas distribuídos.

A implementação de um sistema distribuído traz inúmeros benefícios, e as principais motivações são: **acessibilidade**, no qual permite dar acesso a recurso ao usuário de forma eficiente e controlada; **transparência**, esconde os recursos que estão distribuídos fisicamente por vários computadores, como, localização, reconfiguração de recursos, mobilidade, reaplicação do sistema, concorrência e recuperação do sistema; **ser aberto**, permite que o sistema seja estendido ou implementado de outra maneira; **escalável**, pois mesmo com o crescimento de usuários e recursos o sistema continua efetivo; **e concorrente**, podem ser acesso por diversos usuários, garantindo consistência dos dados nas múltiplas réplicas.

2. Liste algumas desvantagens ou dificuldades dos sistemas distribuídos que sistemas locais não apresentam ou que não impactam tanto esses sistemas.

As principais desvantagens de um sistema distribuído estão relacionadas ao seu **desenvolvimento**,visto que um sistema distribuído é algo bem complexo, e **custo**, pois se torna muito caro implementação de aplicações colaborativas, devido ao fato de os recursos estarem fisicamente separados.

3. Em uma rede Peer-to-Peer não estruturada, como os nós conseguem encontrar os recursos disponibilizados pelos demais nós da rede?

A localização de um recurso em uma rede Peer-to-Peer não estruturada, funciona da seguinte maneira, os arquivos são mantidos em cada nó da rede, e a lista contendo os arquivos é enviada para um servidor central do Napster, e o servidor retorna a lista de IPs que possuem o arquivo para sua transferência entre dois clientes.

4. Qual o problema em manter o estado para um cliente na aplicação servidora?

As aplicações e os processos stateful são aquelas que as informações sobre a interação com o cliente se mantêm. Portanto, é possível saber quais arquivos foram abertos e quais dados o cliente possui em cache. Este serviço oferece um excelente desempenho, por manter cópias locais das informações, mas, a sua confiabilidade acaba sendo problemática.

5. O que são stub e skeleton e por que são necessários em chamada de procedimento remoto?

O **stub** opera de forma semelhante a um proxy para as funções remotas do servidor. Quando um objeto local invoca um método remoto, o **stub** fica responsável por enviar a chamada ao método para o objeto remoto.

Quando invocado o stub segue os seguinte passos:

- Faz a serialização dos parâmetros
- Envia a mensagem pela rede
- Aguarda pela resposta do servidor
- Faz a deserialização e retorna a resposta para o cliente
- Diante de problemas, dispara exceções

O **skeleton**, é responsável por aguardar os pedidos dos clientes, e ao recebê-lo invoca a função do servidor.

- Faz a deserialização dos parâmetros
- Invoca o método no objeto remoto
- Escreve e transmite o resultado

Essa funcionalidade permite que o código da aplicação seja atualizado dinamicamente, sem a necessidade de compilar novamente o código.

6. Adicionar linhas no final de um arquivo pode ser considerada como uma operação idempotentes?

Não, pois uma operação idempotente sempre irá reproduzir o mesmo resultado, independente da quantidade que for executada.

7. Qual o objetivo do portmapper?

O portmapper mapeia de forma dinâmica as portas TCP e UDP para serviços do tipo RPC, de modo que este serviço é uma forma de comunicação utilizado entre processos de diferentes máquinas em uma rede. Portanto, o portmapper pode monitorar um grande números de portas e direcioná-las a outros serviços.

Este serviço é indicado que não esteja disponível ao acesso público, uma vez que podem ser realizados ataques do tipo DDoS e DRDoS.

8. Qual o objetivo de uma linguagem de definição de interface?

Uma Linguagem de descrição de interfaces (IDL), é uma linguagem utilizada para caracterizar a interface dos componentes de software, sendo independente de qualquer linguagem de programação, por este motivo, proporciona a comunicação entre componentes escritos em linguagens de programação diferentes.

Este recurso é utilizado em software para realizarem chamadas de procedimento remoto, estabelecendo uma ligação entre o sistema implementado e o sistema operacional ou linguagem de programação.

9. Quais serviços de nome você conhece? Por que os sistemas distribuídos precisam de serviços de nomes?

O DNS ou Domain Name System, é um serviço de nome. Este serviço é responsável por converter o domínio legível, no caso a URL, em endereços IPs.

Em um sistema distribuído é preciso utilizar o serviço de nomes para referenciar vários recursos, como, computadores, serviços, objetos remotos, arquivos e usuários.

10. Quais são as diferenças entre URI, URL e URN?

A Uniform Resource Locator (URL), trata-se de uma URI que localiza um recurso na rede. Como exemplo temos o site do IFSC-SJ: http://www.sj.ifsc.edu.br.

Já a Uniform Resource Name ou URN, é utilizada para permitir a separação estrita entre identificação (nome único) e localização (endereços URL que podem oferecer o recurso identificado). Como exemplo: urn:isbn:978-8-57-605142-8

A Uniform Resource Identifier, mas conhecida como URI é utilizada para identificar os recursos na Internet, como por exemplo: file:///home/professor/std/aula.tex. Entretanto, pode ser classificado como URN ou URL.

11. Os Serviços Web possuem alguma coisa semelhante com um serviço de nomes e alguma semelhante com uma linguagem de definição de interface?

Sim, pois possui serviço identificado por uma URL, de forma que é semelhante ao serviço de nome, visto que o DNS converte a URL em endereço IP. E também é semelhante ao IDL, já que os dados são representados por XML definidos por tags.

12. Serviços Web baseados no protocolo SOAP trocam pedidos e respostas usando a linguagem XML. Quais são as vantagens e desvantagens de usar XML em chamada de procedimento remoto?

A mensagem SOAP funciona como um pacote para transportar documentos encapsulados no formato XML. E consiste nos seguintes componentes: envelope, cabeçalho e corpo.

O envelope contém toda a mensagem SOAP. Já o cabeçalho (header) contém as informações de controle e processamento, e pode ser opcional. O corpo (body) é a mensagem de fato (payload). Contém informação a ser transportada para o seu destino final.

A principal vantagem do XML é por ser um padrão aberto e pode ser usado em independentes plataformas e linguagens de programação. Uma desvantagem é na questão da quantidade de caracteres, no qual pode acarretar em uma quantidade maior de informações a serem transmitidas pela rede.