
Lista de exercícios 2

Curso: Engenharia de Telecomunicações
Disciplina: STD29006 – Sistemas Distribuídos
Professor: Prof. Emerson Ribeiro de Mello

Aluno
Arthur Anastopulos dos Santos

1 Lista de exercícios 2

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

Resposta: Pois projetar um sistema distribuído, pode fazer com que uma tarefa qualquer a ser executada, se for feita por um conjunto de computadores a computação é distribuída por todos e não apenas por um, não sendo necessária sobrecarga de processamento.

Algumas vantagens de utilizar Sistemas Distribuídos: Permitir computadores baratos e de baixo processamento, formar um supercomputador que normalmente seria muito caro; O compartilhamento de recursos; a transparência; concorrência; tolerância a falhas; escalabilidade; aumento da capacidade de computação; aumento da segurança/estabilidade global; e capacidade de processamento.

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

Resposta: Algumas das Desvantagens/difícildades de um Sistema Distribuído em relação a um sistema local são: A Segurança, onde a facilidade oferecida pelos Sistemas Distribuídos no acesso aos dados, pode criar uma dificuldade na garantia da segurança dos dados existentes e a privacidade dos dados secretos; No quesito de Desenvolvimento, pois um sistema distribuído é mais complexo se comparado a um sistema local; Na questão de custo, pois existe um alto custo para implementação de aplicações distribuídas, devido ao fato de os recursos estarem fisicamente separados; na Rede, que compõem a estrutura de um sistema distribuído podem ocasionar atrasos, perda de mensagens e falhas no sistema. Algumas dessas falhas são ocasionadas pela sobrecarga da rede.

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

Resposta: Uma rede P2P não estruturados geralmente se baseiam em algoritmos aleatorizados para a construção de uma rede sobreposta. A ideia principal é que cada nó mantenha uma lista de vizinhos, que é construída mais ou menos de forma aleatória. Da mesma forma, se assume que os dados são colocados de forma aleatória nos nodos. Assim, quando um nó necessita localizar um item específico, a única coisa que pode fazer é inundar a rede com uma busca, isto pode ser uma desvantagem na busca em que consultas podem não ser respondidas caso o cliente e o hospedeiro estejam muito afastados na rede. Outro problema desta busca é que mecanismo de inundação geralmente causam grande tráfego de sinalização, o que, muitas vezes, torna esse tipo de busca lenta.

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

Resposta: Armazenar no lado do servidor significa que ele será armazenado em algum outro servidor. Isso torna mais fácil compartilhar esses dados e manipulá-los sem a aprovação do próprio dispositivo. A principal problema é que qualquer pessoa que tenha acesso a esse servidor pode visualizar todos os dados dos usuários, o que pode ser uma grande preocupação com em relação a privacidade.

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

Resposta: O Java RMI utiliza o mecanismo de stub e skeletons para se comunicar com objetos remotos. Portanto esta camada é a responsável por receber as chamadas de métodos feitas pelo cliente através da interface e encaminhá-las para o objeto remoto. O stub é uma classe utilizada pela aplicação do cliente e funciona como um proxy entre a aplicação cliente e o objeto remoto. Os parâmetros dos métodos são passados para a classe skeleton, que é utilizada para a aplicação do servidor. A classe skeleton recebe esses parâmetros enviados pelo stub e executa as chamadas no objeto remoto.

6. Adicionar linhas no final de um arquivo pode ser considerada como uma operação idempotentes?
- Resposta:** A operação para anexar linhas a um arquivo não é idempotente, pois o mesmo é estendido toda vez que essa operação é executada.
7. Qual o objetivo do portmapper?
- Resposta:** Um portmapper é o protocolo que mapeia o número ou a versão de um programa Open Network Computing Remote Procedure Call (ONC RPC) para uma porta usada para rede. Após a inicialização, o servidor ONC RPC solicita ao mapeador de portas que atribua um número de porta para cada programa usando o protocolo de transporte para fins de transferência de arquivos ou comunicação. Portanto, os programas usam um portmapper para determinar qual porta é atribuída a eles antes de usá-lo.
8. Qual o objetivo de uma linguagem de definição de interface?
- Resposta:** Sua finalidade é definir os recursos de um serviço distribuído juntamente com um conjunto comum de tipos de dados para interagir com esses serviços distribuídos.
9. Quais serviços de nome você conhece? Por que os sistemas distribuídos precisam de serviços de nomes?
- Resposta:** Em um Sistema Distribuído, um Naming Service é um serviço específico cujo objetivo é fornecer uma nomenclatura consistente e uniforme de recursos, permitindo assim que outros programas ou serviços os encontrem e obtenham os metadados necessários para interagir com eles. Alguns destes serviços são: Nomes e endereços de máquina (host); Nomes de usuário; senhas; Permissões de acesso.
10. Quais são as diferenças entre URI, URL e URN?
- Resposta:** URI: (identificador uniforme de recursos) identifica um recurso (documento de texto, arquivo de imagem, etc.);
URL: (localizador uniforme de recursos) é um subconjunto dos URIs que incluem um local de rede;
URN: (nome uniforme do recurso) é um subconjunto de URIs que inclui um nome dentro de um determinado espaço, mas nenhum local
11. Os Serviços Web possuem alguma coisa semelhante com um serviço de nomes e alguma semelhante com uma linguagem de definição de interface?
- Resposta:** Não há semelhanças entre serviços de nomes e serviços Web, pois os Serviços Web são componentes que permitem às aplicações enviar e receber dados, para que aplicações possam interagir com aquelas que já existem e que sistemas desenvolvidos em plataformas diferentes sejam compatíveis.
Já o serviço de nome ajuda a direcionar o tráfego numa aplicação, conectando nomes de domínio a servidores da Web reais.
Para uma linguagem de definição de interface existe uma semelhança entre ambas, pois estas possibilitam a comunicação entre componentes escritos em linguagens de programação diferentes.
12. Serviços Web baseados no protocolo SOAP trocam pedidos e respostas usando a linguagem XML. Quais são as vantagens e as desvantagens de usar XML em chamada de procedimento remoto?
- Resposta:** O XML quando usado em RPC tem algumas vantagens como: XML separa dados de HTML; simplifica o compartilhamento de dados; simplifica o transporte de dados; simplifica a mudança de plataforma; aumenta a disponibilidade de dados.
Já algumas desvantagens em seu uso seriam: A sintaxe XML é detalhada e redundante em comparação com outros formatos de transmissão de dados baseados em texto, como por exemplo, o JSON; A redundância na sintaxe do XML causa maior custo de armazenamento e transporte quando o volume de dados é grande; O documento XML é menos legível em comparação com outros formatos de transmissão de dados baseados em texto, como JSON;

o XML não suporta array; Os tamanhos dos arquivos XML geralmente são muito grandes devido à sua natureza detalhada, é totalmente dependente de quem está escrevendo.

Referências