

1. Cite cinco tipos de recurso de hardware e cinco tipos de recursos de dados ou de software que possam ser compartilhados com sucesso. Dê exemplos práticos de seu compartilhamento em sistemas distribuídos.

Answer:

Recursos de Hardware	Recursos de Dados/Software
Processador	Bancos de Dados
Memória	Sistemas de Arquivos
Armazenamento	Serviços Web
Rede	Bibliotecas e Frameworks
Dispositivos de E/S	Sistemas de Mensagens

2. Como os relógios de dois computadores ligados por uma rede local podem ser sincronizados sem referência a uma fonte de hora externa? Quais fatores limitam a precisão do procedimento que você descreveu? Como os relógios de um grande número de computadores conectados pela Internet poderiam ser sincronizados? Discuta a precisão desse procedimento.

Answer:

Os relógios de computadores em uma rede local podem ser sincronizados usando protocolos como o NTP. No entanto, a precisão é limitada por atrasos na rede, jitter, latência do sistema e precisão dos relógios locais. Sincronizar os relógios de muitos computadores pela Internet envolve hierarquia de servidores NTP, mas a precisão depende da qualidade das fontes de tempo e da infraestrutura de rede. Geralmente, é suficiente para a maioria dos casos, mas não tão preciso quanto relógios atômicos.

3. Considere as estratégias de implementação de MMOG (massively multiplayer online games) discutidas na Seção 1.2.2. Em particular, quais vantagens você vê em adotar a estratégia de servidor único para representar o estado do jogo para vários jogadores? Quais problemas você consegue identificar e como eles poderiam ser resolvidos?

Answer:

A estratégia de servidor único para MMOGs oferece simplicidade, coerência do mundo virtual e segurança, mas pode enfrentar problemas de escalabilidade, falha do servidor único e latência. Soluções incluem a distribuição da carga entre vários servidores, redundância para falhas e otimização da latência por meio de servidores distribuídos em diferentes regiões geográficas. A escolha depende das necessidades específicas do jogo e das prioridades de desenvolvimento.

4. Um usuário chega a uma estação de trem que nunca havia visitado, portando um PDA capaz de interligação em rede sem fio. Sugira como o usuário poderia receber informações sobre serviços locais e comodidades dessa estação, sem digitar o nome ou os atributos da estação. Quais desafios técnicos devem ser superados?

Answer:

O usuário pode usar um aplicativo de reconhecimento de localização para identificar a estação e, em seguida, usar um aplicativo de mapeamento para obter informações sobre serviços locais e comodidades. Os desafios incluem a precisão do reconhecimento de localização, a disponibilidade de mapas e a precisão das informações sobre serviços e comodidades.

5. Compare e contraste a computação em nuvem com a computação cliente-servidor mais tradicional. O que há de novo em relação à computação em nuvem como conceito?

Answer:

A computação em nuvem é um modelo de computação que permite acesso sob demanda a um conjunto compartilhado de recursos de computação configuráveis (por exemplo, redes, servidores, armazenamento, aplicativos e serviços) que podem ser rapidamente provisionados e liberados com o mínimo de esforço de gerenciamento ou interação com o provedor de serviços. A computação em nuvem é um modelo de computação que permite acesso sob demanda a um conjunto compartilhado de recursos de computação configuráveis (por exemplo, redes, servidores, armazenamento, aplicativos e serviços) que podem ser rapidamente provisionados e liberados com o mínimo de esforço de gerenciamento ou interação com o provedor de serviços.

6. Use a World Wide Web como exemplo para ilustrar o conceito de compartilhamento de recursos, cliente e servidor. Quais são as vantagens e desvantagens das tecnologias básicas HTML, URLs e HTTP para navegação em informações? Alguma dessas tecnologias é conveniente como base para a computação cliente-servidor em geral?

Answer:

A World Wide Web (WWW) exemplifica o compartilhamento de recursos cliente-servidor:

Cliente: Navegadores da web, como Chrome e Firefox, são usados pelos usuários para acessar recursos na web.

Servidor: Os recursos, como páginas da web e imagens, são armazenados em servidores web, que respondem às solicitações dos clientes.

HTML (HyperText Markup Language): O HTML é usado para criar e formatar conteúdo na web, mas não é interativo por si só.

URLs (Uniform Resource Locators): URLs identificam recursos na web, tornando a navegação mais conveniente.

HTTP (HyperText Transfer Protocol): O HTTP permite a transferência de dados entre cliente e servidor.

Vantagens: Simplicidade e interoperabilidade. Acessibilidade e URLs amigáveis.

Desvantagens: Limitações de interatividade. Questões de segurança. Escalabilidade limitada.

Vantagens	Desvantagens
Simplicidade e interoperabilidade	Limitações de interatividade
Acessibilidade e URLs amigáveis	Questões de segurança

Vantagens	Desvantagens
	Escalabilidade limitada
<p>7. Um programa servidor escrito em uma linguagem (por exemplo, C++) fornece a implementação de um objeto BLOB destinado a ser acessado por clientes que podem estar escritos em outra linguagem (por exemplo, Java). Os computadores cliente e servidor podem ter hardware diferente, mas todos eles estão ligados em uma rede. Descreva os problemas devidos a cada um dos cinco aspectos da heterogeneidade que precisam ser resolvidos para que seja possível um objeto cliente invocar um método no objeto servidor.</p>	
<p>Answer:</p>	
<p>Para permitir que um objeto cliente, escrito em uma linguagem diferente (por exemplo, Java), invoque um método em um objeto servidor (por exemplo, C++) em uma rede com computadores de hardware diferente, os principais desafios são:</p>	
Redes Heterogêneas	Garantir que os protocolos de rede sejam compatíveis.
Hardware Heterogêneo	Lidar com diferenças na representação de dados.
Sistemas Operacionais Heterogêneos	Fornecer uma camada de abstração para abordar as variações nos sistemas operacionais.
Linguagens de Programação Heterogêneas	Criar interfaces para traduzir chamadas de função e dados entre as linguagens.
Implementações de Diferentes Desenvolvedores	Assegurar compatibilidade entre bibliotecas e implementações.

O middleware desempenha um papel fundamental na resolução desses desafios.

8. Um sistema distribuído aberto permite que novos serviços de compartilhamento de recursos (como o objeto BLOB do Exercício 1.7) sejam adicionados e acessados por diversos programas clientes. Discuta, no contexto desse exemplo, até que ponto as necessidades de abertura do sistema diferem das necessidades da heterogeneidade.

Answer:

A abertura de um sistema distribuído refere-se à capacidade de adicionar facilmente novos serviços, enquanto a heterogeneidade diz respeito à integração entre diferentes tecnologias e plataformas. Embora ambas estejam relacionadas à interoperabilidade, a abertura se concentra na extensibilidade do sistema, permitindo a adição flexível de serviços, enquanto a heterogeneidade lida com a compatibilidade entre tecnologias diversas. Ambos os conceitos são importantes para sistemas distribuídos, mas têm focos distintos.

9. Suponha que as operações do objeto BLOB sejam separadas em duas categorias – operações públicas que estão disponíveis para todos os usuários e operações protegidas, que estão disponíveis somente para certos usuários nomeados. Indique todos os problemas envolvidos para se garantir que somente os usuários nomeados possam usar uma operação protegida. Supondo que o acesso a

uma operação protegida forneça informações que não devem ser reveladas para todos os usuários, quais outros problemas surgem?

Answer:

Para garantir que somente usuários nomeados acessem operações protegidas em um objeto BLOB, é preciso resolver problemas de autenticação, autorização e controle de acesso. Além disso, para proteger informações sensíveis, é necessário criptografar os dados, gerenciar chaves, controlar exceções e implementar auditorias detalhadas.

10. O serviço INFO gerencia um conjunto potencialmente muito grande de recursos, cada um dos quais podendo ser acessado por usuários de toda a Internet por intermédio de uma chave (um nome de string). Discuta uma estratégia para o projeto dos nomes dos recursos que cause a mínima perda de desempenho à medida que o número de recursos no serviço aumenta. Sugira como o serviço INFO pode ser implementado de modo a evitar gargalos de desempenho quando o número de usuários se torna muito grande.

Answer:

Para projetar um serviço INFO escalável que gerencia uma grande quantidade de recursos com nomes de strings, é essencial criar uma estrutura hierárquica de diretórios para facilitar a busca eficiente e usar indexação, como árvores B ou tabelas de hash. Dividir o espaço de nomes em partições gerenciáveis e implementar cache e redundância ajudará a aliviar a carga no serviço. Priorize a escalabilidade horizontal, adicionando servidores conforme necessário e usando balanceadores de carga. Adote algoritmos de busca eficazes e caches distribuídos. Monitore o desempenho regularmente, otimizando o sistema quando necessário e planejando dimensionamento futuro com base no crescimento projetado. Com essas estratégias, o serviço INFO pode lidar com eficiência com um grande número de recursos e usuários.

11. Liste os três principais componentes de software que podem falhar quando um processo cliente chama um método em um objeto servidor, dando um exemplo de falha em cada caso. Sugira como os componentes podem ser feitos de modo a tolerar as falhas uns dos outros.

Answer:

Em sistemas distribuídos, ao chamar um método de um objeto servidor, três principais componentes de software podem falhar: o processo do cliente, sujeito a erros de programação ou falhas de hardware; a comunicação entre cliente e servidor, suscetível a problemas de rede e perda de pacotes; e o processo do servidor, propenso a exceções não tratadas ou falta de recursos. Para torná-los tolerantes a falhas, o cliente pode implementar recuperação e relatórios de erros, a comunicação pode empregar retransmissão e protocolos confiáveis, e o servidor pode capturar exceções, registrar falhas e permitir replicação para garantir a continuidade do serviço em caso de falhas.

12. Um processo servidor mantém um objeto de informação compartilhada, como o objeto BLOB do Exercício 1.7. Dê argumentos contra permitir que os pedidos do cliente sejam executados de forma concorrente pelo servidor e a favor disso. No caso de serem executados de forma concorrente, dê um exemplo de uma possível “interferência” que pode ocorrer entre as operações de diferentes clientes. Sugira como essa interferência pode ser evitada.

Answer

Permitir a execução concorrente de pedidos de clientes por um processo servidor pode melhorar o desempenho e a capacidade de resposta, mas também introduzir desafios, como a possibilidade de interferência de dados quando operações de diferentes clientes ocorrem simultaneamente em um objeto compartilhado, o que pode levar a inconsistências. Para evitar essa interferência, o servidor pode usar mecanismos como travas, transações e semáforos para sincronizar e controlar o acesso concorrente, garantindo a consistência dos dados. A decisão de permitir a execução concorrente deve levar em consideração as necessidades específicas do sistema, o volume de clientes e a importância da integridade dos dados. Uma abordagem equilibrada que permite a concorrência controlada muitas vezes é a mais adequada.

13. Um serviço é implementado por vários servidores. Explique por que recursos poderiam ser transferidos entre eles. Seria satisfatório para os clientes fazer multicast (difusão seletiva) de todos os pedidos para o grupo de servidores, como uma maneira de proporcionar transparência de mobilidade para os clientes?

Answer

A transferência de recursos entre servidores em sistemas distribuídos pode ocorrer por várias razões, incluindo balanceamento de carga, escalabilidade e recuperação de falhas. No entanto, utilizar multicast como uma forma de proporcionar transparência de mobilidade para clientes pode não ser a solução ideal devido a desafios como sobrecarga de rede, questões de segurança, eficiência e complexidade. Em vez disso, estratégias de mobilidade mais controladas, como a atualização das informações de localização dos recursos, podem ser mais apropriadas para garantir que os clientes acessem recursos de maneira eficiente, independentemente de sua localização física, adaptando-se às necessidades específicas do sistema e evitando problemas associados ao multicast em larga escala.

14. Os recursos na World Wide Web e outros serviços são nomeados por URLs. O que denotam as iniciais URL? Dê exemplos de três diferentes tipos de recursos da Web que podem ser nomeados por URLs.

Answer

Os URLs, que significam "Uniform Resource Locators," são identificadores uniformes de recursos na World Wide Web. Eles apontam para diferentes tipos de recursos, como páginas da Web estáticas (por exemplo, "http://www.example.com/index.html"), imagens (como "https://www.example.com/images/pic.jpg"), e serviços da Web (por exemplo, "https://api.example.com/getdata"). Esses URLs permitem aos usuários e sistemas localizar e acessar recursos específicos na Internet de forma única e universal, tornando-os fundamentais para a navegação e a interação na Web.

15. Cite um exemplo de URL HTTP. Liste os principais componentes de um URL HTTP, dizendo como seus limites são denotados e ilustrando cada um, a partir de seu exemplo. Até que ponto um URL HTTP tem transparência de localização?

Answer

Um exemplo de URL HTTP, como "http://www.example.com:8080/path/to/resource?param1=value1¶m2=value2#section1", possui vários componentes essenciais, incluindo o esquema (HTTP), o nome do servidor (www.example.com), a porta (8080, opcional), o caminho (/path/to/resource), a consulta (?param1=value1¶m2=value2, opcional) e o fragmento (#section1, opcional). Esses elementos definem com precisão a localização do recurso na web e permitem uma identificação direta e acessível, tornando a web altamente interconectada e facilmente acessível em todo o mundo.

