2.1 Dê três exemplos específicos e contrastantes dos níveis de heterogeneidade cada vez maiores experimentados nos sistemas distribuídos atuais, conforme definido na Seção 2.2. página 39

R:

A heterogeneidade pode ser aplicada aos seguintes aspectos:

1. Linguagens de programação: Algumas linguagens de programação costumam variar a forma como representam caracteres e estruturas de dados, como vetores e registros. Essas diferenças devem ser consideradas, caso programas escritos em diferentes linguagens precisem se comunicar.

2. Hardware de computadores: Tipos de dados, como os inteiros, podem ser representados de diversas maneiras em diferentes tipos de hardware. Por exemplo, os bytes de valores inteiros podem ser armazenados considerando a ordem a partir do byte mais significativo ou a partir do byte menos significativo. Portanto, caso ocorra troca de mensagens entre programas executados em diferentes hardwares, é necessário ter atenção em relação a esse ponto.

3. Implementações de diferentes desenvolvedores: Programas escritos por diferentes desenvolvedores não podem se comunicar, a menos que utilizem padrões comuns, por exemplo, uma mesma representação de tipos de dados primitivos e estruturas de dados nas mensagens. Outro exemplo de padrão comum é o caso dos protocolos de Internet.

2.2 Quais problemas você antevê no acoplamento direto entre entidades que se comunicam, que está implícito nas estratégias de invocação remota? Consequentemente, quais vantagens você prevê a partir de um nível de desacoplamento, conforme o oferecido pelo não acoplamento espacial e temporal? Nota: talvez você queira rever sua resposta depois de ler os Capítulos 5 e 6. página 43

R:

Problemas relacionados ao acoplamento direto entre entidades que se comunicam:

1. Dependência. Quando duas entidades estão acopladas diretamente, qualquer mudança em uma delas pode afetar a outra, criando uma forte dependência entre elas e dificultando a manutenção e a evolução do sistema.

2. Escalabilidade. O acoplamento direto pode afetar a escalabilidade de um sistema pois, conforme o número de entidades aumenta, torna-se cada vez mais difícil gerenciar as dependências entre elas.

3. Desempenho. A comunicação entre as entidades pode criar sobrecarga de rede e diminuir a eficiência da aplicação.

Vantagens previstas a partir de um nível de desacoplamento:

1. Flexibilidade. O desacoplamento permite que as entidades evoluam independentemente uma da outra, tornando o sistema mais flexível e adaptável a mudanças.

2. Reutilização. Entidades desacopladas podem ser reutilizadas em diferentes contextos por não estarem ligadas intimamente a outras entidades específicas

3. Escalabilidade. Com o desacoplamento, as entidades podem ser facilmente distribuídas e escalonadas independentemente umas das outras.

4. Manutenção. Com o desacoplamento, as mudanças feitas em uma entidade não afetam diretamente outras entidades do sistema, tornando-o mais fácil de se realizar manutenções.

2.3 Descreva e ilustre a arquitetura cliente-servidor de um ou mais aplicativos de Internet importantes (por exemplo, Web, correio eletrônico ou news). página 46

R:

1. Arquitetura Cliente-Servidor da Web

A arquitetura cliente-servidor da Web é amplamente utilizada na Internet para fornecer acesso a páginas da Web. Nesta arquitetura, o cliente envia uma solicitação HTTP para o servidor, que retorna uma resposta contendo a página da Web solicitada.
Na arquitetura da Web, o cliente é geralmente um navegador, enquanto o servidor é um aplicativo Web que recebe solicitações HTTP e retorna páginas HTML, juntamente com outros recursos (como imagens, scripts etc.) para o navegador exibir. O servidor Web pode ser configurado para lidar com várias solicitações simultaneamente, o que permite que muitos usuários acessem o aplicativo Web ao mesmo tempo.

2. Arquitetura Cliente-Servidor de E-mail

A arquitetura cliente-servidor de e-mail é usada para permitir o envio e recebimento de e-mails em todo o mundo. Nesta arquitetura, o cliente (um aplicativo de e-mail) envia uma solicitação para o servidor de e-mail, que verifica se há novas mensagens e as entrega ao cliente.
Na arquitetura de e-mail, o cliente pode ser um aplicativo de e-mail em um computador, telefone ou outro dispositivo, enquanto o servidor é responsável por armazenar e gerenciar as mensagens de e-mail. Os protocolos de e-mail como o SMTP (Simple Mail Transfer Protocol) e o POP (Post Office Protocol) são usados para permitir que o cliente e o servidor se comuniquem entre si.

2.5 Um mecanismo de busca é um servidor Web que responde aos pedidos do cliente para pesquisar em seus índices armazenados e (concomitantemente) executa várias tarefas de Web crawling para construir e atualizar esses índices. Quais são os requisitos de sincronização entre essas atividades concomitantes? página 46

R:

A sincronização entre as atividades concomitantes em um mecanismo de busca é fundamental para garantir que os resultados da pesquisa sejam precisos e atualizados. Para isso, devem ser utilizadas técnicas avançadas de gerenciamento de concorrência, sincronização de índices, gerenciamento de cache e controle de qualidade para atender aos requisitos de sincronização dessas atividades.

Detalhamento sobre cada uma dessas técnicas:

1. Gerenciamento de Concorrência: O mecanismo de busca deve ser capaz de gerenciar a concorrência entre as várias tarefas de Web crawling para evitar que elas se sobrecarreguem e prejudiquem o desempenho do servidor. Isso pode ser feito usando técnicas de bloqueio e exclusão mútua.

2. Sincronização de Índices: O mecanismo de busca deve garantir que seus índices sejam atualizados com precisão à medida que as tarefas de Web crawling são executadas. Isso pode ser alcançado usando um sistema de armazenamento distribuído e garantindo que os índices sejam atualizados em tempo real conforme novos dados são coletados.

3. Gerenciamento de Cache: O mecanismo de busca deve ser capaz de gerenciar o cache de suas páginas da Web indexadas para garantir que os resultados da pesquisa sejam entregues o mais rápido possível. Isso pode ser feito usando técnicas de armazenamento em cache e monitoramento do desempenho do cache.

4. Controle de Qualidade: O mecanismo de busca deve ter um sistema de controle de qualidade em vigor para garantir que os resultados da pesquisa sejam precisos e relevantes. Isso pode ser feito usando técnicas de análise de dados e mineração para avaliar a qualidade dos resultados e ajustar os algoritmos de pesquisa conforme necessário.

2.6 Frequentemente, os computadores usados nos sistemas peer-to-peer são computadores desktop dos escritórios ou das casas dos usuários. Quais são as implicações disso na disponibilidade e na segurança dos objetos de dados compartilhados que eles contêm e até que ponto qualquer vulnerabilidade pode ser superada por meio da replicação? páginas 47, 48

R:

Em relação à disponibilidade, os computadores podem ser desligados, desconectados ou desativados a qualquer momento, o que pode afetar a disponibilidade dos objetos de dados compartilhados que eles contêm. Isso significa que os objetos de dados podem não estar disponíveis quando outros usuários os buscam, resultando em uma experiência de usuário pobre.

Em relação à segurança, o uso de computadores desktop em um sistema peer-to-peer também pode representar um risco de segurança, pois esses computadores podem ser menos seguros do que os servidores dedicados. Os usuários podem não ter as medidas de segurança adequadas em vigor, como firewalls, antivírus e autenticação forte, o que pode permitir que hackers mal-intencionados acessem os objetos de dados compartilhados ou os dados pessoais dos usuários.

No entanto, a replicação de dados pode ajudar a superar algumas dessas vulnerabilidades de segurança. Quando os dados são replicados em vários nós em um sistema peer-to-peer, se um nó for comprometido, os outros nós ainda terão uma cópia dos dados. Isso aumenta a disponibilidade e a resiliência do sistema.

Além disso, os sistemas peer-to-peer geralmente usam técnicas avançadas de criptografia para proteger os dados compartilhados. Por exemplo, o protocolo BitTorrent usa criptografia para proteger o tráfego de dados entre os nós do sistema. Isso ajuda a proteger a privacidade dos usuários e dificulta que hackers mal-intencionados acessem os dados.

2.11 Considere um servidor simples que executa pedidos do cliente sem acessar outros servidores. Explique por que geralmente não é possível estabelecer um limite para o tempo gasto por tal servidor para responder ao pedido de um cliente. O que precisaria ser feito para tornar o servidor capaz de executar pedidos dentro de um tempo limitado? Essa é uma opção prática? página 62

R:

Não é possível estabelecer um limite para o tempo gasto por tal servidor para responder ao pedido de um cliente, porque existem muitos fatores imprevisíveis que afetam o tempo de processamento de um pedido, tais como:

1. A complexidade do pedido: O tempo necessário para processar um pedido pode variar amplamente, dependendo da complexidade do pedido. Alguns pedidos podem ser processados em alguns milissegundos, enquanto outros podem levar vários segundos ou até minutos para serem processados.

2. O tráfego de rede: O tempo de resposta do servidor também é afetado pelo tráfego de rede entre o cliente e o servidor. Se o tráfego de rede estiver congestionado, o tempo de resposta do servidor será afetado.

3. A carga do servidor: O tempo de resposta do servidor também depende da carga que o servidor está enfrentando. Se muitos clientes estiverem fazendo pedidos ao mesmo tempo, o servidor pode ficar sobrecarregado e levar mais tempo para processar os pedidos.

Para tornar o servidor capaz de executar pedidos dentro de um tempo limitado, é necessário implementar técnicas de gerenciamento de carga e escalabilidade. Por exemplo, o servidor pode ser dividido em vários servidores menores e independentes, cada um responsável por uma parte do processamento do pedido. Isso ajuda a distribuir a carga e a reduzir o tempo de resposta do servidor.

Outra técnica é implementar um cache no servidor. Quando um pedido é recebido, o servidor verifica se os dados necessários já estão armazenados em cache. Se estiverem, o servidor pode retornar os dados armazenados em cache em vez de executar o processamento novamente, o que reduz o tempo de resposta.

No entanto, essas técnicas podem não ser práticas em todos os casos. A implementação de técnicas de gerenciamento de carga e escalabilidade pode ser cara e complexa, e pode não ser viável para servidores pequenos ou com poucos recursos. Além disso, a implementação de um cache pode não ser eficaz se os dados estiverem mudando com frequência. Portanto, é importante avaliar cuidadosamente as necessidades do servidor e escolher as técnicas apropriadas para garantir que o servidor possa executar pedidos dentro de um tempo limitado.

2.15 Considere dois processos, X e Y, que utilizam o serviço de comunicação B do Exercício 2.14 para se comunicar entre si. Suponha que X seja um cliente e que Y seja um servidor e que uma invocação consiste em uma mensagem de requisição de X para Y, seguida de Y executando a requisição, seguida de uma mensagem de resposta de Y para X. Descreva as classes de falha que podem ser exibidas por uma invocação. página 67

R:

Existem várias classes de falhas que podem ocorrer durante uma invocação do processo X para o processo Y usando o serviço de comunicação B. Algumas das classes de falhas possíveis são:

1. Falha na comunicação: Pode haver falhas de comunicação entre os processos X e Y, resultando em perda de mensagens ou em um tempo excessivo de espera para uma resposta. Isso pode ocorrer devido a problemas de rede, erros no código do serviço de comunicação ou outras falhas.

2. Falha no servidor: O servidor Y pode falhar ao executar a requisição recebida de X. Isso pode ocorrer devido a erros no código do servidor, falta de recursos do sistema, problemas de acesso a dados ou outras falhas.

3. Falha no cliente: O cliente X pode falhar ao enviar a mensagem de requisição para o servidor Y ou ao processar a resposta recebida. Isso pode ocorrer devido a erros no código do cliente, problemas de rede ou outras falhas.

4. Falha de segurança: O serviço de comunicação pode ser comprometido por um ataque malicioso, resultando em perda ou roubo de dados, violação de privacidade ou outras consequências indesejáveis.

5. Condições de corrida: Condições de corrida podem ocorrer se múltiplos processos tentarem acessar o serviço de comunicação B ao mesmo tempo. Isso pode resultar em conflitos de dados, bloqueios de recursos ou outros problemas.

2.18 Descreva as possíveis ocorrências de cada um dos principais tipos de ameaça à segurança (ameaças aos processos, ameaças aos canais de comunicação, negação de serviço) que poderiam ocorrer na Internet.

R:

1. Ameaças aos processos: Essas ameaças visam os processos de um sistema, com o objetivo de obter acesso não autorizado ou causar danos ao sistema. Alguns exemplos de ameaças aos processos incluem:

- Malware: Softwares maliciosos, como vírus, cavalos de Tróia e worms, que se infiltram em sistemas para causar danos ou roubar informações.

- Ataques de força bruta: Tentativas repetidas de adivinhar senhas ou chaves de criptografia para acessar sistemas ou dados.

- Engenharia social: Ataques que exploram a confiança dos usuários para obter acesso não autorizado a sistemas ou informações.

2. Ameaças aos canais de comunicação: Essas ameaças visam os canais de comunicação, como redes e servidores, com o objetivo de interceptar ou manipular informações que são transmitidas através deles. Alguns exemplos de ameaças aos canais de comunicação incluem:

- Intercepção: Ataques que visam interceptar informações que são transmitidas através de uma rede. Isso pode permitir que um atacante acesse informações sensíveis, como senhas ou dados de cartão de crédito.

- Modificação: Ataques que visam alterar informações que são transmitidas através de uma rede. Isso pode resultar em informações incorretas ou danificadas.

- Spoofing: Ataques que envolvem a criação de identidades falsas ou manipulação de endereços IP para enganar usuários ou servidores.

3. Negação de serviço (DoS): Essas ameaças visam interromper o funcionamento normal de sistemas ou redes, tornando-os inacessíveis aos usuários legítimos. Alguns exemplos de ameaças de DoS incluem:

- Ataques de sobrecarga: Ataques que visam sobrecarregar um servidor ou rede com tráfego falso ou excessivo, tornando-o inacessível aos usuários legítimos.

- Ataques de inundação: Ataques que visam inundar um servidor ou rede com uma grande quantidade de solicitações falsas ou inúteis, tornando-o inacessível aos usuários legítimos.

- Ataques de exaustão de recursos: Ataques que visam esgotar os recursos de um servidor ou rede, como CPU, memória ou largura de banda, tornando-o inacessível aos usuários legítimos.