

1.1 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.

Hardware: discos, impressora, memória ram, processador, switch.

Software: arquivos, banco de dados, protocolos de comunicação, funções do sistema operacional, IP.

Exemplos:

- Compartilhamento de impressora entre vários computadores em um escritório;
- Memória principal de uma máquina que é compartilhada por todos os processadores através de uma barramento que os interliga;
- Compartilhamento de documentos no google docs.

1.2 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.

Já que o relógio é síncrono, é possível saber os limites da taxa de derivação dos relógios, o atraso máximo de transmissão de mensagens e o tempo que leva para executar cada etapa de um processo, através da técnica de relógios lógicos. O relógio lógico de Lamport é um contador de software que aumenta a contagem monotonicamente e cujo valor não precisa ter nenhum relacionamento em particular com qualquer relógio físico.

Não existe a noção de relógio global em um sistema distribuído; portanto, os relógios de diferentes computadores não fornecem necessariamente a mesma hora. Toda comunicação entre processos é obtida por meio de troca de mensagens. A comunicação por troca de mensagens em uma rede de computadores pode ser afetada por atrasos, sofrer uma variedade de falhas e ser vulnerável a ataques contra a segurança.

É impossível fazer a sincronização através de sistemas assíncronos.

1.3 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?

Vantagem: Utilizar uma arquitetura cliente-servidor na qual uma única cópia do estado do mundo é mantida em um servidor centralizado e acessada por programas clientes em execução nos consoles ou em outros equipamentos dos jogadores. A arquitetura centralizada ajuda significativamente no gerenciamento do mundo virtual e a cópia única

também diminui as preocupações com a coerência. Assim, o objetivo é garantir resposta rápida por meio da otimização de protocolos de rede e também para os eventos recebidos.

Desvantagem: Se o servidor central cair não teria outro para substituí-lo.

1.4 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?

Este usuário poderia se conectar a uma WIFI, e receber informações sobre este local devido ao provedor desta WIFI inserir tais informações na rede. Aplicativos de reconhecimento de localização também podem dar essas informações.

Desafios:

- O usuário deve executar um programa em seu dispositivo que escuta estas URLs, dando controle suficiente, sem que elas estejam inundadas por URLs indesejadas;
- Os meios de propagar a URL (por exemplo, infravermelho ou uma LAN sem fio 802.11) deve ter um alcance que corresponde à propagação física do próprio local.

1.5 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?

O modelo básico de funcionamento da internet é o chamado cliente/servidor, onde existe uma máquina central que tem armazenado os documentos que os usuários desejam ver e estes, por sua vez, são os chamados clientes.

A computação em nuvem é a distribuição de serviços de computação – servidores, armazenamento, bancos de dados, redes, software, análises, inteligência e muito mais pela Internet (“a nuvem”), proporcionando inovações mais rápidas, recursos flexíveis e economia na escala. Você normalmente paga apenas pelos serviços de nuvem que utiliza, ajudando a reduzir os custos operacionais, a executar sua infraestrutura com mais eficiência e a dimensionar conforme as necessidades da sua empresa.

1.6 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?

Uma página web é qualquer documento que faça parte de um sítio web e que costuma conter ligações (igualmente chamadas hiperligações ou *links*) para facilitar a navegação entre os conteúdos.

HTML: É uma linguagem relativamente simples para analisar e processar, está disponível em diversas plataformas e sua performance é ótima em PC's, dispositivos móveis e tablets, mas confunde apresentação com os dados subjacentes que estão sendo apresentados.

URL: O objetivo de um URL é identificar um recurso. Os navegadores examinam os URL's para acessar os recursos correspondentes. Às vezes, o usuário digita um URL no navegador. Mais comumente, o navegador pesquisa o URL correspondente quando o usuário clica em um link, quando seleciona um URL de sua lista de bookmarks ou quando o navegador busca um recurso incorporado em uma página Web, como uma imagem.

A desvantagem é que não são suficientemente ricos como links de recursos.

HTTP: São protocolos simples de serem implementados, mas apesar de serem dados sensíveis, eles são tratados de forma textual, já que o HTTP é um protocolo baseado em texto. Dessa forma, os dados que são enviados podem ser acessados no meio do tráfego, prejudicando muito a segurança da informação.

HTTP e URLs são convenientes como uma base para a computação cliente-servidor.

1.7 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.

A heterogeneidade se aplica aos seguintes aspectos:

- Redes;
- Hardware de computador;
- Sistemas Operacionais;
- Implementações de diferentes desenvolvedores.

Em **redes**, os protocolos já lidam com suas diferenças.

Se temos **hardware** diferente é necessário lidar com diferentes representações de dados entre cliente e servidor. Os tipos de dados, como os inteiros, podem ser representados de diversas maneiras em diferentes tipos de hardware; por exemplo, existem duas alternativas para a ordem em que os bytes de valores inteiros são armazenados: uma iniciando a partir do byte mais significativo e outra, a partir do byte menos significativo. Essas diferenças na representação devem ser consideradas, caso mensagens devam ser trocadas entre programas sendo executados em diferentes hardwares.

Se os computadores possuem **sistemas operacionais** diferentes, é necessário lidar com operações de envio e recepção de mensagens diferentes, mesmo que todos os computadores na internet precisem incluir uma implementação dos protocolos Internet,

nem todos fornecem, necessariamente, a mesma interface de programação de aplicativos para esses protocolos.

Os programas escritos por **diferentes desenvolvedores** não podem se comunicar, a menos que utilizem padrões comuns; por exemplo, para realizar a comunicação via rede e usar uma mesma representação de tipos de dados primitivos e estruturas de dados nas mensagens. Para que isso aconteça, padrões precisam ser estabelecidos e adotados. Assim é o caso dos protocolos Internet.

1.8 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.