

1) Jogos online, sistemas de pesquisas, redes sociais, plataformas idênticas, sistemas distribuídos ubíqua, móveis

2) O principal problema decorrente do acoplamento direto entre as entidades comunicantes que resulta do uso de modelos de interação baseadas em chamada remota (de métodos ou procedimentos) é que para que a comunicação ocorra os componentes têm que estar acoplados temporalmente e referencialmente, tornando a comunicação mais complexa. Em contrapartida, com a adoção de modelos de interação com desacoplamento no tempo e no espaço essa complexidade na comunicação fim-a-fim acaba melhorando o desempenho.

3) - Mudar o servidor de local porque a distância interfere nesse aspecto,

- Usar uma CDN(Content Delivery Network). Com ele, o conteúdo é programado para ser entregue sempre no local mais próximo geograficamente. Assim, o requisitante recebe os dados com mais velocidade. Seja qual for o local, ao invés de atravessar o mundo para acessar o servidor original, o usuário é encaminhado para uma CDN próxima que tenha uma cópia dos arquivos em questão.

- Utilizar outros servidores DNS

- Revise os canais de comunicação: Compartilhamento de mensagens, áudios, imagens, documentos, transações financeiras, dentre outros, exigem canais de comunicação rápidos e seguros

- Use a escalabilidade automática: O emprego da nuvem para realizar escalabilidade automática é outra dica para reduzir a latência no servidor de seus clientes. Dessa forma, o desempenho e a largura da banda são ampliados, diminuindo a demora.

4) Um wrapper ou adaptador oferece uma interface aceitável para uma aplicação cliente.

Transforma as funções do componente legado nas funções esperadas pela aplicação. Exemplos: As funções de wrapper podem ser usadas para escrever rotinas de verificação de erro para funções de sistema pré-existentes sem aumentar o comprimento de um código, não necessitando repetir a mesma verificação de erro para cada chamada para a função, outro exemplo seria em uma linguagem de programação que não suporta múltiplas heranças de outras classes, as funções de wrapper podem ser usadas para simulá-la.

5) Distribuição vertical - Surge da divisão das aplicações distribuídas em camadas lógicas complementares, sendo que os componentes de cada camada executam em um servidor (máquina) diferente.

Distribuição horizontal - Um cliente ou servidor pode ser fisicamente dividido em partes lógicas equivalentes, mas cada parte opera sobre sua própria fatia do conjunto de dados completo.

Exemplos de vertical: arquitetura cliente-servidor multidividida, Exemplo de horizontal:

Arquiteturas peer-to-peer.

6) As três camadas são:

- Camada de interface contém unidades para interfaceamento com o usuário ou com aplicações externas

- Camada de processamento contém as funções de uma aplicação, mas sem dados específicos

- Camada de dados contém os dados que os clientes desejam manipular através das funções da aplicação components

7) Os principais problemas são:

- Principalmente a dependência porque o P_i acaba dependendo de P_{i+1} para executar e ao mesmo tempo P_{i-1} depende de P_i , se um desses processos falhar por qualquer motivo, toda a execução para de funcionar já que cada um depende de outro processo para executar

- Perda de desempenho porque além de ter muitos processos executando, essa dependência pode gerar muita perda de desempenho porque caso um processo falhe no meio, todos os processos que foram executados não adiantaram de nada, ou seja, desgaste atoa

8) Desvantagens:

- Legalidade: em muitas ocasiões, o conteúdo com direitos autorais é compartilhado, o que acaba sendo ilegal. A lei vale essa prática em muitos países, entre os quais a Espanha.
- Risco: Muitos desses programas que permitem a transferência de arquivos ponto a ponto são uma porta para malware, especialmente spyware. Portanto, é necessário que você tenha proteção e, se possível, procure um serviço com visualização, comentários do usuário e outros métodos para detectar conteúdo inapropriado e arriscado.

9) Inundação (flooding): A busca por determinado arquivo é feita enviando uma mensagem a todos os nós adjacentes. Esses nós adjacentes também enviam uma mensagem aos demais nós conectados a eles e assim por diante até obterem uma resposta positiva ou um número máximo de saltos (TTL) Caminho aleatório: o nó que inicia a busca u passa a requisição pelo dado d para um vizinho escolhido aleatoriamente, v. Se v não tem d, ele encaminha a requisição para um de seus vizinhos, escolhido aleatoriamente, e assim por diante.

O algoritmo de inundação gera uma vasta quantidade de pacotes duplicados, na verdade um número infinito, a menos que algumas medidas sejam tomadas para tornar mais lento o processo. O algoritmo de caminho aleatório pode reduzir significativamente a sobrecarga de pesquisa de busca em redes não estruturadas e, assim, aumentar a escalabilidade dos algoritmos.

10) Os requisitos são possuírem maior taxa de transmissão, maior espaço em disco, capacidade de processamento, conectividade ou fiabilidade.

11) É uma maneira de compartilhar arquivos entre usuários em forma de protocolo de rede que permite ao utilizador realizar downloads (descarga) de arquivos sem que o arquivo em si precise estar em um servidor. É mais um exemplo de rede peer-to-peer de transferência de arquivos entre dois ou mais usuários. Esse protocolo introduziu o conceito de partilhar o que já foi descarregado, maximizando o desempenho e possibilitando altos índices de transferência, mesmo com um enorme número de usuários realizando descargas (downloads) de um mesmo arquivo simultaneamente. Enquanto nas outras redes você deve procurar pelo próprio programa o arquivo e baixá-lo, nos arquivos de protocolo BitTorrent a coisa é diferente. Acontece que, todos os usuários que fazem o download de um arquivo, tornam-se também fornecedores deste arquivo; quando alguém for baixá-lo, todos os computadores que já o possuem através do torrent e que estiverem online no momento vão atuar como "semeadores", cada um fornecendo uma parte do arquivo, portanto quanto mais usuários fizerem o download, mais rápido será feito o compartilhamento, pois a parte enviada do arquivo por cada um será menor.

12) O interceptor é a construção em software que intercepta um fluxo usual de controle e aciona um outro código para ser executado. Exemplos: podemos colocar limites de banda num cliente, para não acessar o arquivo ou um dado varias vezes. Podemos também ter interceptador que fazem encriptação das mensagens quando essa sai do cliente ou do servidor

13) Os aspectos são ele pode ser um modificador especial em alguns momentos então mesmo quando sai do middleware, antes de entregar pro servidor ou sair do cliente, pode ter esse interceptador para pegar essa imagem e fazer algo com ela pra depois entregar pro consumidor

14) Basicamente o middleware funciona como um meio de comunicação e gerenciamento de dados, e nesse caso estamos tratando com um espaço de dados compartilhado entao todos as tarefas criadas

pelo master vao ser passados por lá. Os workers vão basicamente pegar algumas dessas tarefas e executar.