

1. Qual é o papel do middleware em um sistema de informação distribuído? (valor da questão: 2,0 pontos)

O papel do middleware em um sistema de informação distribuído é conectar e fazer a interação entre as camadas de um sistema distribuído. Um exemplo seria um sistema de três camadas onde o middleware faz a comunicação, entre outros, dos requests do cliente e os dados do servidor. Também é quem garante a ligação entre os dispositivos do sistema.

4,0
0,5
et

2. O que é uma arquitetura cliente-servidor de três camadas? Discuta os pros e os contras desta arquitetura (valor da questão: 2,0 pontos)

É uma arquitetura de tipo lasanha, onde existem três camadas: usuário (o front-end); servidor (back-end ou aplicação) e a base de dados (SGBD, file server, etc.). Como pontos negativos, podemos destacar o fato de que todas as camadas dependem umas das outras (ex: se o usuário não estiver disponível, o sistema como um todo fica indisponível. Como ponto positivo, podemos destacar que o sistema fica modularizado, possibilitando o desenvolvimento/atualização de diferentes camadas ao mesmo tempo, bem como facilitando a manutenção.

2,0

3. Um problema importante relacionado com a natureza do cliente-servidor Web é que é fácil um servidor ficar sobrecarregado. Quais são as possíveis soluções para contornar este problema? Discuta-as. (valor da questão: 2,0 pontos)

Existem 4 estratégias para auxiliar na distribuição de carga:

- * DNS cíclico: consiste em vários servidores com IPs diferentes mapeados para um mesmo domínio. O servidor DNS é configurado para alternar qual IP é retornado a cada request.

- * Front-end: consiste em usar um servidor arquitado que recebe todas as requests e, através de um algoritmo, seleciona para qual servidor enviá-la.

- * Distribuição das requests em função do conteúdo: similar ao front-end, mas o arquitado escolhe para qual servidor enviar a request em função do conteúdo que está sendo acessado.

- * Vários servidores com o mesmo IP: a request é enviada para todos os servidores e eles executam um algoritmo para decidir quem vai responder.

2,0
e

4. Você está projetando um sistema de informação distribuído e uma característica importante é desempenho (isto é, o tempo de resposta deverá ser adequado). Discuta três decisões de projeto você tomaria para garantir esta propriedade. (valor da questão: 2,0 pontos)

Medida	Discussão
Esconder a latência pelo front-end	Usar mecanismos na aplicação de modo que requêsts só serão realizadas se os dados estão validados no front-end. Um exemplo seria validar os campos de um formulário antes do envio ao servidor.
Armazenamento local dos dados	Através de uma "cache" os dados são armazenados no dispositivo do cliente ou em um servidor intermediário. Desta forma, requêsts de mesmo conteúdo são respondidos rapidamente.
Distribuição geográfica dos dados	Posicionando servidores geograficamente mais próximos dos clientes, o tempo de resposta diminui. Um exemplo seria a World Wide Web, onde existem servidores de uma mesma aplicação espalhados pelo globo.

paralelismo

5. E se a propriedade mais importante fosse disponibilidade, discuta quais três medidas seriam tomadas. (valor da questão: 2,0 pontos)

Medida	Discussão
Desenvolvimento de sistemas distribuídos abertos	Usar práticas de desenvolvimento que, por meio de interfaces modularizadas, aumentem a interoperabilidade e escalabilidade, consequentemente melhorando a disponibilidade e acesso do sistema.
Aplicar cópias do sistema	Espalhar cópias completas do sistema e independentes em locais geográficos diferentes. Consequentemente, a falha de um ou mais pontos não prejudica o funcionamento dos demais.
Controle dinâmico da quantidade de servidores	Implementar sistemas que monitorem os servidores e em caso de sobrecarga ativem uma quantidade maior de servidores impedindo a indisponibilidade do sistema. Um exemplo seriam os momentos de pico não esperados em um sistema como uma notícia global que aumenta intensamente o fluxo de um website.

20

20