

A lista é opcional. Portanto, não entregue, pois não tem nota e não será corrigida (sem gabarito).

A lista é uma guia e não necessariamente representa o que será pedido na prova.

CAPÍTULO 1

1. Uma definição alternativa para um sistema distribuído é que ele é um conjunto de computadores independentes que dá a impressão de ser um *sistema único*, isto é, o fato de haver vários computadores fica completamente oculto para os usuários. Dê um exemplo para o qual essa visão viria muito a calhar.
2. Qual é o papel do middleware em um sistema distribuído?
4. Explique o que quer dizer *transparência* (de distribuição) e dê exemplos de diferentes tipos de transparência.
5. Por que às vezes é tão difícil ocultar a ocorrência e a recuperação de falhas em um sistema distribuído?
6. Por que nem sempre é uma boa idéia visar à implementação do mais alto grau de transparência possível?
7. O que é um sistema distribuído aberto e quais são os benefícios que a abertura proporciona?
8. Descreva, com exatidão, o que quer dizer *sistema escalável*.
9. Pode-se conseguir escalabilidade pela aplicação de diferentes técnicas. Quais são essas técnicas?
11. Dissemos que, quando uma transação é abortada, o mundo é restaurado a seu estado anterior, como se a transação nunca tivesse acontecido. Mentimos. Dê um exemplo no qual restaurar o mundo é impossível.
12. Executar transações aninhadas requer certo tipo de coordenação. Explique o que um coordenador deveria realmente fazer.
13. Argumentamos que a transparência de distribuição pode não estar presente em sistemas pervasivos. Essa declaração não vale para todos os tipos de transparências. Dê um exemplo.
14. Já demos alguns exemplos de sistemas distribuídos pervasivos: sistemas domésticos, sistemas eletrônicos para tratamento de saúde e redes de sensores. Amplie essa lista com mais exemplos.

CAPÍTULO 2

1. Se um cliente e um servidor forem colocados longe um do outro, podemos ver a latência de rede dominar o desempenho global. Como podemos atacar esse problema?
2. O que é uma arquitetura cliente-servidor de três divisões?
3. Qual é a diferença entre uma distribuição vertical e uma distribuição horizontal?
4. Considere uma cadeia de processos P_1, P_2, \dots, P_n implementando uma arquitetura cliente-servidor multidivida. O processo P_i é cliente do processo P_{i+1} , e P_i retornará uma resposta a P_{i-1} somente após receber uma resposta de P_{i+1} . Quais são os principais problemas dessa organização quando se examina o desempenho de requisição-resposta no processo P_1 ?
5. Em uma rede de sobreposição estruturada, mensagens são roteadas de acordo com a topologia da sobreposição. Cite uma importante desvantagem dessa abordagem.

CAPÍTULO 3

1. Nesse problema você deverá fazer uma comparação entre ler um arquivo usando um servidor de arquivos monothread ou um servidor multithread. Obter uma requisição para trabalho, despachá-la e fazer o resto do processamento necessário demora 15 ms, considerando que os dados necessários estejam em uma cache na memória principal. Se for preciso uma operação de disco, como acontece em um terço das vezes, serão necessários mais 75 ms, durante os quais o thread dorme. Quantas requisições por segundo o servidor pode manipular se for monothread? E se for multithread?
2. Teria sentido limitar a quantidade de threads em um processo servidor?
3. Descrevemos no texto um servidor de arquivo multithread mostrando por que ele é melhor do que um servidor monothread e um servidor com máquina de estado finito. Há alguma circunstância na qual um servidor monothread poderia ser melhor? Dê um exemplo.
9. Proxies podem suportar transparência de replicação invocando cada réplica, como explicado no texto. O lado servidor de uma aplicação pode estar sujeito a uma chamada replicada?
10. Construir um servidor concorrente por meio da multiplicação de um processo tem algumas vantagens e desvantagens em comparação com servidores multithread. Cite algumas.
12. Como podemos impedir que uma aplicação evite um gerenciador de janela e, assim, consiga bagunçar completamente uma tela?
13. Um servidor que mantém uma conexão TCP/IP com um cliente é com estado ou sem estado?
14. Imagine um servidor Web que mantenha uma tabela na qual endereços IP de clientes sejam mapeados para as páginas Web acessadas mais recentemente. Quando um cliente se conecta ao servidor, este consulta o cliente em sua tabela e, caso o encontre, retorna a página registrada. Esse servidor é com estado ou sem estado?
17. Considere um processo P que requer acesso ao arquivo F , disponível na máquina em que P está executando no momento em questão. Quando P passa para outra máquina, ele ainda requer acesso a F . Se a vinculação arquivo-máquina for fixa, como poderia ser implementada a referência a F no âmbito do sistema?
18. Descreva com detalhes como pacotes TCP fluem no caso de transferência TCP junto com a informação sobre endereços de fonte e destino nos vários cabeçalhos.

CAPÍTULO 4

1. Em muitos protocolos de camadas, cada camada tem seu próprio cabeçalho. Por certo seria mais eficiente ter um único cabeçalho à frente de cada mensagem que contivesse todos os controles do que ter todos esses cabeçalhos separados. Por que isso não é feito?
2. Por que serviços de comunicação de nível de transporte freqüentemente são inadequados para construir aplicações distribuídas?
3. Um serviço multicast confiável permite que um remetente passe mensagens confiáveis para um conjunto de receptores. O melhor lugar para esse serviço é uma camada de middleware, ou ele deveria ser parte de uma camada de nível mais baixo?
7. Considere que um cliente chama uma RPC assíncrona para um servidor e, na seqüência, espera até que o servidor retorne um resultado usando uma outra RPC assíncrona. Essa abordagem é o mesmo que deixar o cliente executar uma RPC normal?
12. Suponha que você só possa usar primitivas de comunicação assíncronas transientes, entre elas apenas uma primitiva assíncrona *receive*. Como você implementaria primitivas para comunicação transiente *síncrona*?
13. Suponha que você só possa utilizar primitivas de comunicação transiente síncrona. Como você implementaria primitivas para comunicação transiente *assíncrona*?
14. Faz sentido implementar comunicação persistente assíncrona por meio de RPCs?
16. Tabelas de roteamento no WebSphere da IBM e em muitos outros sistemas de enfileiramento de mensagens são configuradas manualmente. Descreva um modo simples de fazer isso automaticamente.
17. Com comunicação persistente, um receptor geralmente tem seu próprio buffer local no qual mensagens podem ser armazenadas quando o receptor não estiver em execução. Para criar tal buffer, talvez seja preciso especificar seu tamanho. Cite um argumento a favor e outro contra a especificação do tamanho.
18. Explique por que a comunicação transiente síncrona tem problemas inerentes de escalabilidade e como eles podem ser resolvidos.
25. Quando se trata de procurar arquivos em um sistema peer-to-peer não estruturado, pode ser útil restringir a busca a nós que tenham arquivos semelhantes aos seus. Explique como o gossiping pode ajudá-lo a achar esses nós.

CAPÍTULO 5

1. Dê um exemplo de onde um endereço de uma entidade E precisa ser resolvido para um outro endereço a fim de poder acessar E .
2. Você consideraria que um URL como *http://www.acme.org/index.html* é independente de localização? E o endereço *http://www.acme.nl/index.html*?
3. Dê alguns exemplos de identificadores verdadeiros.
4. Um identificador tem permissão de conter informações sobre a entidade que ele referencia?
6. Observe o sistema Chord como mostra a Figura 5.4 e considere que o nó 7 acabou de se juntar à rede. Qual seria sua tabela de derivação? Haveria quaisquer mudanças em outras tabelas de derivação?
7. Considere um sistema Chord baseado em DHT no qual k bits de um espaço de identificadores de m bits foram reservados para designar a superpares. Se os identificadores forem designados aleatoriamente, quantos superpares podemos esperar que um sistema de N nós tenha?
8. Se inserirmos um nó em um sistema Chord, precisaremos atualizar imediatamente todas as tabelas de derivação?
9. Qual é a maior desvantagem de consultas recursivas na resolução de uma chave em um sistema baseado em DHT?
10. Servidores de nomes de nível alto em DNS, isto é, servidores de nomes que implementam nós no espaço de nomes DNS que está próximo da raiz, em geral não suportam resolução recursiva de nomes. Poderíamos esperar grande aprimoramento de desempenho caso suportassem?
19. Explique como o DNS pode ser usado para implementar uma abordagem baseada em localização nativa para localizar hospedeiros móveis.

- Figura 5.4 da pergunta 6 corresponde ao slide 24 da aula do módulo 5

CAPÍTULO 6

2. Considere o comportamento de duas máquinas em um sistema distribuído. Ambas têm relógios que devem pulsar 1.000 vezes por milissegundo. Um deles realmente pulsa a essa taxa, mas o outro pulsa somente 990 vezes por milissegundo. Se as atualizações UTC chegam uma vez por minuto, qual será a máxima defasagem entre os relógios?
4. Quando um nó sincroniza seu relógio com o de outro nó, em geral é uma boa idéia também levar em conta medições anteriores. Por quê? Dê um exemplo de como essas leituras anteriores podem ser levadas em conta.
5. Adicione uma nova mensagem à Figura 6.9 que seja concorrente com a mensagem A, isto é, que não acontece antes de A ou não acontece depois de A.
6. Para conseguir multicast totalmente ordenado com marcas de tempo Lamport, é estritamente necessário que cada mensagem seja reconhecida?
12. Como as entradas na Tabela 6.1 mudariam se admitíssemos que os algoritmos podem ser implementados sobre uma LAN que suporta broadcast por hardware?
13. Um sistema distribuído pode ter vários recursos independentes. Imagine que o processo 0 quer acessar o recurso A e o processo 1 quer acessar o recurso B. O algoritmo de Ricart e Agrawala pode resultar em deadlocks? Explique sua resposta.
7. Considere uma camada de comunicação na qual as mensagens são entregues somente na ordem em que foram enviadas. Dê um exemplo no qual até mesmo essa ordenação é desnecessariamente restritiva.
8. Muitos algoritmos distribuídos requerem a utilização de um processo coordenador. Até que ponto esses algoritmos realmente são considerados distribuídos? Comente sua resposta.
9. Na abordagem centralizada da exclusão mútua (Figura 6.14), ao receber uma mensagem de um processo que está liberando seu acesso exclusivo aos recursos que estava usando, o coordenador normalmente concede permissão ao primeiro processo na fila. Cite um outro algoritmo possível para o coordenador.
10. Considere novamente a Figura 6.14. Suponha que o coordenador caia. Isso sempre derruba o sistema? Se não derrubar, sob quais circunstâncias isso acontece? Há algum modo de evitar o problema e fazer com que o sistema seja capaz de tolerar quedas de coordenador?
14. Suponha que dois processos detectem a morte do coordenador simultaneamente e ambos decidam convocar uma eleição que utilize o algoritmo do valentão. O que acontecerá?
15. Na Figura 6.20 temos duas mensagens *ELEIÇÃO* que circulam simultaneamente. Embora não haja problema em ter duas delas, seria mais elegante se uma fosse eliminada. Proponha um algoritmo para fazer isso sem afetar a operação do algoritmo de eleição básico.

- Figura 6.9 da pergunta 5 corresponde à Figura do lado esquerdo do slide 67 da aula (completa) do módulo 7. Além disso, troque “mensagem A” por “mensagem m1”.
- Figura 6.14 da pergunta 9 e 10 corresponde ao slide 8 da aula (completa) do módulo 7
- Tabela 6.1 da pergunta 12 corresponde ao slide 20 da aula (completa) do módulo 7
- Considere a Figura 6.20 da pergunta 15 como a do slide 27 da aula (completa) do módulo 7

CAPÍTULO 7

9. Que tipo de consistência você usaria para implementar um mercado eletrônico de ações? Explique sua resposta.
10. Considere uma caixa postal pessoal para um usuário móvel, implementada como parte de um banco de dados distribuído de longa distância. Que tipo de consistência centrada no cliente seria mais adequado?
11. Descreva uma implementação simples de consistência leia-suas-escritas para apresentar páginas Web que acabaram de ser atualizadas.
13. Quando se usa um leasing, é necessário que os relógios de um cliente e do servidor, respectivamente, estejam fortemente sincronizados?
19. Para implementar multicast totalmente ordenado por meio de um seqüenciador, uma abordagem seria primeiro repassar uma operação ao seqüenciador, que então lhe designaria um número exclusivo e, na seqüência, faria multicast da operação. Cite duas abordagens alternativas e compare as três soluções.
20. Um arquivo é replicado em dez servidores. Faça uma lista de todas as combinações de quórum de leitura e quórum de escrita que são permitidas pelo algoritmo de votação.