

Respostas SD

Capítulo 1

1.P: Uma definição alternativa de sistema distribuído, é uma aglomeração de vários computadores físicos, mas que apenas é visível como um todo para o utilizador.

Indique um exemplo de onde este seja conveniente.

R: Um dos casos onde isto pode ser conveniente é na computação paralela, isto é, quando pensamos em desenvolver uma aplicação para correr em paralelo preocupamo-nos de saber todos os pcs que vão executar as partes da aplicação, mas agora se fosse possível desenvolvermos uma aplicação sem ser necessário ter essa preocupação e saber que isso ia correr distribuídamente a vida seria muito mais fácil.

2. P: Qual é o papel da camada de middleware num sistema distribuído?

R: Aumentar a transparência que falta num sistema distribuído em rede. Este aumenta a transparência de sistema distribuído para um só.

3. P: Muitas redes de sistemas estão organizados em back office e front office. Como é que uma organização pode transparecer isto para um sistema distribuído.

R: Não é necessário criar toda esta transparência pois é mesmo necessário ver isto como dois sistemas completamente diferentes. Pois destinam-se a diferentes utilizadores.

4. P: Explique o que significa transparência na distribuição e indique alguns tipos de transparência.

R: Vários tipos de transparência:

Acesso -> Esconder diferenças na representação de dados e na forma como os recursos são acedidos

Localização -> Esconder localização dos recursos

Migração -> Esconder o facto de um recurso mudar de lugar

Relocação -> Esconder o facto de um recurso ter mudado de lugar enquanto estava a ser usado

Replicação -> Esconder que um recurso é replicado

Concorrência -> Esconder que um recurso pode ser partilhado por diferentes utilizadores competitivos

Falha -> Esconder a falha e recuperação de um recurso

5. P: Porque é que é muitas vezes difícil esconder uma ocorrência de falha e possível recuperação em sistemas distribuídos?

R: É difícil prever que irá acontecer uma falha, pois para se prever esta tem de acontecer logo é quase impossível transparecer ao utilizador que tal facto aconteceu pois este apercebe-se que o sistema não está "normal"

6. P: Porque é que nem sempre é boa ideia implementar o maior grau possível de transparência?

R: Isto pode dar origem a perda de funcionalidades por parte do serviço que esta transparência

esconde e acima de tudo diminuiu o desempenho de certos serviços que as vezes os utilizadores preferem ocorrer falha a não perderem desempenho.

7. P: O que é um sistema distribuido aberto e que beneficios este oferece?

R: Este possibilita a total mudança do sistema já existente. É possível ser desenvolvido por várias entidades de modo a melhorar ou alterar a funcionalidade do mesmo.

8. P: Descreve precisamente o que entendes por scalable systems?

R: Um sistema escalavel é um sistema que respeitando o numero de componentes, o tamanho geografico e o numero de sistemas administrativos este pode crescer em dimensão sem perder desempenho.

9:P: A escalabilidade pode ser alcançada aplicando diferentes tecnicas. Quais sao essas tecnicas?

R: A escalabilidade pode ser alcançada garantindo que aplicação é capaz de ser distribuida, replicada e em cache

10. P: Explica o que entendes por organização virtual e dá uma sugestão de como esta organização pode ser implementada.

R: Esta organização virtual providencia aos utilizadores a existência de espaços diferentes onde este é só um mas este também é visto como vários recursos aglomerados num só.

11. Quando é abortada uma transacção, dissemos que o mundo era colocado no seu ponto anterior como se a transacção nunca tivesse ocorrido. É mentira, dê um exemplo onde é impossível voltar atrás.

R: Quando mandamos imprimir um documento e que a meio verificamos que o mesmo contem um erro.

12. Executar um conjunto correlacionado de transações requerem alguma forma de coordenação. O que deve fazer o coordenador?

R: Este deve garantir que se uma transacção falha todas as outras devem ser descartadas. E todas elas apenas podem terminar quando receberem indicação do coordenador que todas podem terminar.

13. Argumentamos que a transparencia na distribuição nem sempre de ser universal em todos os sistemas. Esta afirmação não é de toda verdade para todos os tipos de transparencia. Dê um exemplo:

R: Estes sistemas normalmente são sistemas que não convem que falhem. Um bom exemplo deste sistema é o sistema de saude.

14. Já demos alguns exemplos de sistemas distribuidos pervasive, sistemas domesticos, sistemas de cidades de saude e redes de sensores. Aumente a lista com mais exemplos.

R:

Capítulo 2

1- Se um cliente e um servidor estiverem colocados muito longe um do outro, a performance global do sistema é afectada. Como podemos contornar o problema.

R: Se o sistema tiver dividido em varias camadas logicas, podemos fazer com que mais camadas logicas corram do lado cliente.

Dividir o lado do cliente em várias partes, de forma a que quando uma estiver a aguardar a resposta do servidor se poder agendar outra para processar.

2. O que é uma arquitectura cliente servidor em 3 camadas?

R: A arquitetura cliente servidor em 3 camadas é constituída por user interface/aplicação/dados. Temos o user interface do lado do cliente, que vai comunicar com o servidor da aplicação, que por sua vez também vai ser cliente do servidor de dados.

3. Qual a diferença entre uma distribuição vertical e uma distribuição horizontal?

R: Distribuição vertical refere-se a uma arquitetura de multiplas camadas distribuidas ao longo de varias maquinas. Distribuição horizontal é a distribuição de uma camada por várias máquinas. Por exemplo para aceder a uma unica base de dados distribuida.

4. Considerando uma cadeia de processos, P_1, P_2, \dots, P_n que implementam uma arquitectura cliente servidor multicamada. O processo P_i é cliente do processo P_{i+1} e responde para P_{i-1} apenas após receber a resposta de P_{i+1} . Quais são os principais problemas nesta organização quando olhamos para a performance do pergunta-resposta do processo P_i .

R: O principal problema será que o tempo de resposta de P_i para P_{i-1} dependerá da resposta de P_{i+1} para P_i .

5. Numa overlay network estruturada, as mensagens são encaminhadas de acordo com a topologia da rede(overlay). Qual é a principal desvantagem desta abordagem?

R: Sendo uma overlay network estruturada uma rede em que os elementos da rede conhecem o caminho para os outros elementos através de um logical path, isto requer a existencia de hash tables para determinar os caminhos. No caso de o elemento da rede que queremos atingir estar muito proximo o uso de hash table pode tirar performance, em contra partida a redes não estruturadas.

Capítulo 3

1: Neste problema comparamos a leitura dum ficheiro usando um servidor single-thread e um servidor multi-thread. Este demorou 15 msec para realizar o trabalho e fazer o resto do processo necessario, assumindo que os dados necessarios estavam na cache na memoria principal. Se for necessaria uma operação no disco são necessarios 75 msec adicionais, os quais a thread dorme. Quantos pedidos/sec pode o servidor tratar se for single thread? E multi-thread?

R: (pela resposta oficial) No caso de processamento multithread $\frac{2}{3}$ das vezes teremos o dados em cache logo perdemos 15msec e $\frac{1}{3}$ teremos os dados em disco logo perdemos 90msec. Ou seja 40msec em media. E no caso de singlethread temos que ir sempre ao disco e perdemos 90msec por operacao.

2: Fara sentido limitar o numero de threads no processo dum servidor?

R: Sim, por duas razoes, a primeira porque o uso de threads requer um uso acrescido de memoria uma vez que as threads usam memoria stack propria. A segunda razão e talvez mais importante é porque, apartir dum certo numero threads o acesso a memoria tornasse caotico, ou seja não aproveita a caracteristica das memorias de tentar prever qual parte de memoria será usada mais provavelmente nos instantes seguintes.

3: No texto descrevemos um servidor de ficheiros multithread, mostrando porque é que este é melhor single thread e uma maquina de estados finita. Existe alguma circunstancia em que um sistema single thread seja melhor? Dê um exemplo.

R: Existem tarefas que são tão rapidas a executar, que é preferivel executar com um sistema single thread, porque usando multi thread acaba por perder performance, por exemplo enquanto se vai preparando o scheduling(alocar memoria para o thread, criar o thread, salvarguardar o contexto do processador e colocar os novos dados), ja poderia estar a executar a tal tarefa super mega hyper rapidissima :P

4: Associar estaticamente uma unica thread com um processo “leve” nem sempre é uma boa ideia. Porque não?

R: ~~Porque se o processo vai correr apenas essa thread, se a thread crashar, pára logo todo o processo.~~

Como um processo “leve” corre ao nível da kernel e um thread corre ao nível de utilizador será necessário estar a transferir a memoria do espaço de utilizador para o espaço de kernel. Esta acção reduz significativamente o desempenho.

5: Ter apenas um processo “leve” por processo também não é uma boa ideia. Porquê?

R: Como quando um processo “leve” é lançado por um processo este é visto como uma thread, este vai ser executado no nível de utilizador, mas como este tem as mesmas caracteristicas dum processo se este bloquear/crashar o processo que o lançou vai bloquear à espera deste.

6: Descreva um esquema simples onde neste exista tantos processos “leves” como threads.
R:

Capítulo 4

1. Em muitas camadas protocolares, cada camada tem o seu próprio cabeçalho. Certamente era mais eficiente existir apenas um cabeçalho numa mensagem com a informação de todas as camadas. Porque é que não é feito?

R: Cada camada tem um cabeçalho e dados para que cada camada seja mais independente e transparente das outras camadas. Se houvesse apenas um cabeçalho para todas as camadas essa transparência e independência seriam destruídas.

2. Porque é que os serviços de comunicação ao nível do transporte são normalmente inapropriados para construir aplicações distribuídas.

R: Porque dificilmente oferecem a transparência necessária para que os desenvolvedores de aplicações consigam produzir sistemas compatíveis e transparentes. daí ter que se deve usar sempre a camada intermédia de middleware.

A camada de middleware em rede é constituída normalmente pelas 3 últimas camadas o que não engloba a camada de transporte. Se construíssemos ao nível da camada de transporte não estaríamos a usar a interface de middleware.

3. Um serviço multicast confiável permite passar mensagens para um grupo de receptores. Deve esse serviço pertencer à camada de middleware ou deve fazer parte da camada de baixo-nível?

R: Depende dos casos, é mais simples implementar logo na camada de transporte, mas se quisermos garantir alguma escalabilidade ao sistema, convém implementá-la nas camadas mais altas.

Capítulo 8

1. Sistemas seguros oferecem normalmente um alto nível de segurança porquê?

R: Se por exemplo as respostas dadas por servidores não podem ser consideradas de confiança, porque em algum ponto correram o risco de ter sido adulteradas por algum fulano maroto, então não faz sentido dizer que é um sistema seguro. para tal acontecer, o servidor precisa de dar a confiança necessária ao cliente, implementando soluções de segurança com o cliente.

2. O que faz o modelo fail-stop em caso de falha tão difícil de implementar.

R: Torna-se difícil de implementar no caso de falha, pois ao falhar perdesse a conexão ao elementos que falhou e será necessário estar repetidamente a verificar se a falha recuperou o que pode tornar o sistema lento. E é ainda difícil para os outros elementos do sistema compreenderem a gravidade da falha e do estado do elemento que falhou.

3. Considere um browser que retorna um página desactualizada que estava na cache, em vez da versão mais recente que já está no servidor. Isto é uma falha? Se sim, de que tipo?

R: Depende da consistência que foi garantida ao utilizador, por exemplo se um web-browser estiver constantemente a carregar as páginas do servidor, vai perder eficiência, por outro lado, se garantir que no máximo uma página está desactualizada T periodos de tempo, corre o risco de pelo meio fornecer informação desactualizada. Que é uma falha do tipo fudido :P

Pode acontecer que quando é pedido ao browser para recarregar uma página, este vá ao servidor verificar a ultima actualizacao da pagina e este devido a carga não consiga responder ao pedido da cache num tempo definido, levando esta apresentar a ultima versão existente em cache. Esta falha pode consistir num problema de comunicação(o que quando isto acontece o browser detecta falha de comunicação) ou numa falha de resposta do servidor.

4. Pode o modelo de redundância tripla lidar com falhas Byzantinas?

R: (em primeiro lugar, o que esta merda de Bizantinas??

Falha silenciosa – Processador (ou processo) para e não responde mais

Falha Bizantina – Processador (ou processo) continua executando mas gerando informações incorretas, maliciosamente ou não)

Redundância modular tripla (ou **TMR**, do termo em inglês triple modular redundancy) **é uma forma de tolerância a falhas proposta por Von Neumann. A unidade de hardware é triplicada**, o que faz com que o trabalho seja feito em paralelo. **Isso gera entradas para um circuito de votação que retorna o voto da maioria**, resultando em somente uma saída. Isso significa que, **se um dos sistemas falharem os outros dois podem estar corretos e mascarar o erro**, o que não demanda técnicas de correção e substituição de elementos. **Ainda assim pode haver falha em mais de um componente e vulnerabilidade do votador.**

5. Que nome se dá a um conjunto de lobos que usam telemóvel?

R: Alcateis

Capítulo 10

1. Há uma diferença entre objectos remotos e objectos distribuidos. Qual?

R: Objectos remotos são objectos instanciados num unico servidor, mas que podem ser invocados por qualquer cliente. enquanto que objectos distribuidos são objectos que podem estar disponibilizados paralelamente em vários servidores, mas são vistos como um unico, sendo transparente para o cliente.

2: Porque é util definir as interfaces dum objecto na Interface Definition Language(?!?!?!?)
(comunidade onde com interfaces para software livre)?

R:

3: Algumas implementações de middleware de objectos distribuidos são inteiramente baseados em invocações de metodos dinamicos. Mesmo invocações estaticas são compiladas para dinamicas. Qual é o beneficio desta abordagem?

R:

4: Delinear um protocolo que implemente pelo menos uma semantica para a invocação dum objecto.

R:

5: Devem os objectos dos lados do servidor e do cliente

Outro Livro

Capítulo 1

1: Dê 5 tipos de recursos de hardware e 5 tipos de recursos de dados ou de software que podem ser úteis para partilhar. Dê exemplos das suas partilhas e como estas ocorrem em sistemas distribuídos (16, 20-22)

R: Hardware: Impressoras(so pode ser usado por utilizador de cada vez, há um FIFO para os processos pendentes), discos (há um processo de controlo através de permissões), cpu(o scheduling do cpu pode ser feito localmente ou de transparente pelo SD) e memória e drives
Dados/Software: Dados existentes, divisão duma aplicação entre vários computadores,

2: Como devem ser os relógios de dois computadores que estão ligados por uma rede ser sincronizados sem o uso de um relógio externo? Que factores contribuem para a precisão do procedimento que descreveu? Como podem os relógios existentes nos computadores ligados pela internet serem sincronizados? Discuta a precisão deste procedimento?

R: Podemos recorrer a técnica de Lamport... Pode-se sincronizar os relógios de computadores existentes na internet usando um servidor que contém o tempo correcto e todos os outros lhe perguntam a ele, como o exemplo do NTP.

3: Um utilizador chega a uma estação de comboios e na sua posse tem um PDA capaz de se ligar a rede wireless disponível na estação. Sugira como este utilizador pode ser informado dos serviços disponíveis na estação sem necessitar de introduzir informação sobre a estação. Que desafios técnicos pode isto trazer?

R: Podemos usar reconhecimento de localização(geolocalização) ou pelo reconhecimento de contexto. (usando uma gama de IP que indicasse a localidade)

É necessário que os dispositivos contivessem recursos(camera, GPS) para essa funcionalidade.

4: Quais são as vantagens e desvantagens de HTTP, URL e HTML para navegação em informações? São estas tecnologias convenientes como base para a computação cliente-servidor?(22)

R: os 3 "protocolos" têm grande credibilidade histórica e têm as vantagens de por exemplo, no caso do HTTP que é um protocolo de (...)

HTTP --> Vantagens:

Desvantagens: Protocolo e Envio/Resposta

5: Utilize o WWW como exemplo para ilustrar o conceito de partilha de recursos entre cliente e servidor.

Recursos na WWW e outros serviços são designados por URL. O que significam estas iniciais? Dê 3 tipos diferentes mas ordenados de recursos URL que podem ser designados por URLs.
R:

6: Dê um exemplo de HTTP URL. Liste os componentes principais de um HTTP URL, começando pelo como os seus limites são definidos e ilustrando cada um dos exemplos.
R:

7: Imagine um programa dum servidor escrito numa linguagem (c++ por exemplo), com uma implementação baseada em objectos do tipo BLOB que é destinado a ser acedida por clientes usando programas escritos numa linguagem diferentes(ex: java). Ambos os computadores podem ter diferentes configurações tanto de hardware como de software mas estão ambos ligados a internet. Descreva o problema derivado de cada um dos tipos de heterogeneidade necessitam de ser resolvidos para tornar possível o cliente poder invocar um método disponível do lado do servidor.

R: Tipos Heterogeneidade:

Redes: Utilização dos protocolos

Hardware do Computador: Tratamento dos dados recebidos através da placa de rede

Sistemas Operacionais: Os sistemas operacionais necessitem de implementar os protocolos tal como estes foram definidos estes não usam a mesma interface para o uso desses protocolos.

Linguagens de Programação: Cada linguagem de programação é diferente como esta lida com os arrays, com registos, com estruturas de dados logo estas diferenças necessitam de ser consideradas quando estão a ser desenvolvidos softwares

Implementações de diferentes programadores: A menos que estas aplicações sejam desenvolvidas em concordância entre os vários programadores seria impossível elas comunicarem.

Como estes computadores comunicam por rede usam placas para se comunicar e como foi dito a sua configuração é diferente. Um computador com uma placa de rede implementa os protocolos de comunicação em rede definidos nos RFC's mas como trata os dados recebidos é diferente do outro computador que também tem a sua maneira de tratar os dados. Como os objectos do tipo BLOB contém muita informação existe o risco desta informação ser tratada de maneira diferente ocorrendo assim erros na comunicação entre os programas. Estes necessitariam de ser implementados em consenso e usando um tipo de dados construído para a troca de mensagens entre os diferentes clientes.

8: Um sistema distribuído aberto permite que novos recursos e serviços sejam partilhados como o exemplo da alínea anterior o BLOB, possam ser adicionados e acedidos por uma variedade de clientes diferentes. Discuta neste contexto em diferem as diferenças entre sistema

aberto e a heterogeneidade.

R: No sistema aberto existe a partilha das interfaces usadas/desenvolvidas para que outros utilizadores os possam usar e assim construir programas diferentes que possam comunicar entre si, estes são baseados em sistemas de comunicação uniforme igual para todos os sistemas.

Temos o exemplo dos RFC's

9: Suponha que as operações do objecto BLOB estão separados em duas categorias - operações publicas que estão disponiveis para todos os utilizadores e operações protegidas que estão disponiveis apenas para certos utilizadores. Enumere todos os problemas envolvidos na protecção que só os utilizadores "registados" possam aceder as funcionalidades protegidas. Suponha que o acesso as funcionalidades protegidas oferece informação que não deve ser revelada para todos os utilizadores, quais são os problemas que isso acarreta.

R: Esta segurança entre utilizadores pode ser resolvido usando técnicas de criptografia.

Quando uma pessoa está a consultar a sua conta num banco ou a efectuar uma transacção só a esta pessoa diz respeito a informação que circula não sendo conveniente ser divulgada informação confidencial dum utilizador a outro diferente.

10: Os serviços INFO gerem um potencial muito grande de recursos, em que cada um pode ser acedido pelos utilizadores na internet utilizando apenas uma chave(uma string). Discuta uma abordagem a criação de nomes de recursos que consigam ter a menor perca de desempenho a medida que o numero de recursos aumeta. Sugira como um serviço INFO pode ser implementado de modo a evitar grandes percas de desempenho quando o numero de utilizadores aumenta.

R:

11: Liste os 3 componentes de software principais que podem falhar quando um processo dum cliente invoca um metodo num servidor de objectos, dando um exemplo de falha para cada um dos componentes. Sugira como os componentes podem ser produzidos de modo a tolerar a falha de outro.

R:

12: O processo dum servidor mantém informação partilhada como o objecto BLOB do ex 1.7. Dê argumentos para permitir e negar pedidos de clientes que possam ser executados concorrentemente pelo servidor. No caso que eles são executados concorrentemente dê exemplos de possiveis interferencias que possam ocorrer entre operações de clientes diferentes. Sugira como tal interferencia deve ser controlada.

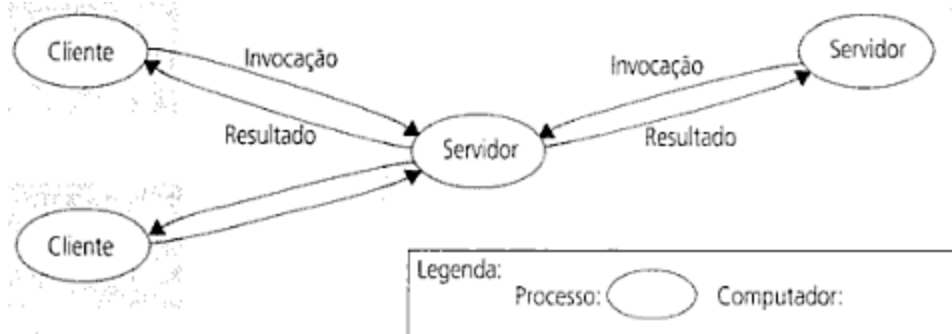
R:

13: Um serviço é implementado por vários servidores. Explique como os recursos devem ser transferidos entre eles. Seria satisfatorio que os clientes fizessem uma difusão selectiva de todos os pedidos multicast para o grupo de servidores de maneira a obter transparencia movel.

Capítulo 2

1: Descreva e ilustre a arquitectura cliente-servidor de um ou mais aplicativos de internet importantes. (43-44)

R: Um cliente é apenas um processo que pede informação a um servidor esperando pela resposta do servidor. Os servidores podem também por vezes efectuar tarefas de cliente em outros servidores. Pesquisas na internet(web crawlers), acesso pelos servidores a servidores com dados das páginas da internet, bases de dados, ...



2: Para os aplicativos discutidos na alinea anterior diga como os servidores cooperam no fornecimento de um serviço. (44-46)

R: Por exemplo um servidor web é muitas vezes cliente dum servidor de dados, dum servidor de DNS. Um servidor de pesquisas é muitas vezes um servidor-cliente pois efectua pedidos a outros servidores por HTTP

3: Como os aplicativos discutidos na alinea 2.1 envolvem o particionamento e/ou a replicação(ou armazenamento em cache) dos dados entre os servidores?(44-46)

R:

4: Um mecanismo de busca é um servidor web que responde aos pedidos do cliente para pesquisar os seus índices armazenados e simultaneamente executa várias tarefas de web crawling para construir e actualizar esses índices. Quais são os requisitos de sincronização entre essas actividades simultaneas? (43-44)

R: A operação de resposta e de pesquisa da informação é independente. Não existe grande necessidade de as sincronizar e estas podem ser executadas em simultaneamente.

5: Frequentemente, os computadores usados nos sistemas peer-to-peer são computadores desktop nas casas dos utilizadores ou escritórios. Quais são as implicações disso na disponibilidade e segurança dos objectos de dados compartilhados que eles contêm e até que ponto qualquer vulnerabilidade pode ser superada por meio de replicação? (43,46)

R: Com os meios de acesso a internet disponível nas diversas moradias e empresas muitas pessoas usam este serviço para obter software diverso, estando a tornar-se num cliente mas também num servidor o que por vezes pode tornar estes computadores vulneráveis, pois estes

utilizadores não usam/usam fracos tipos de protecção.

6: Liste os tipos de recurso local que são vulneráveis a um ataque de um programa não confiável, cujo download é feito de um site remoto e que é executado em um computador local. (44-46)

R: Temos o exemplo de código móvel -> applets de java

7: Dê exemplos de aplicações onde o uso de código móvel seja vantajoso. (44-46)

R: O uso de código móvel é vantajoso quando consultamos uma página que requer esta seja muita vez actualizada necessitando muitas das vezes de ser feito o download de informação do servidor. Por exemplo um corrector que queira dar aos seus clientes informação sobre a bolsa.

8: Que factores afectam a reatividade de um aplicativo que acede a dados partilhados, gerenciados por um servidor? Descreva as soluções disponíveis e a sua utilidade. (50-51)

R: As programas clientes frequentemente acedem a recursos partilhados. Quando um serviço remoto está envolvido a velocidade com que a resposta é gerada não é só determinada pela carga e desempenho do servidor e da rede, mas também pelo atraso de todos os componentes e software envolvidos. Esta pode ser mais lenta também sempre que é preciso aceder a dados visto que o acesso a estes é lento mesmo que estes dados estejam no mesmo computador.

9: Diferencie o uso de buffer e cache. (52-53)

R: A cache mantém informação para poder responder mais rapidamente a eventos pedidos como no caso de browsers Web em que sempre que é efectuado um pedido de visualização duma página este verifica primeiro a informação na cache se esta actualizada apresentando assim a informação de forma mais rápida, enquanto os buffers são usados para juntar informação e depois mostra-la(caso de audio/video) ou então coloca-la num disco(evitar demasiados acessos ao disco)

10: Dê alguns exemplos de falhas no hardware e no software que possam/não possam ser toleradas pelo uso de redundância em um sistema distribuído. Até que ponto o uso de redundância nos casos apropriados torna um sistema tolerante a falhas? (53-54)

R:

11: Considere um servidor simples que executa pedidos do cliente sem aceder a 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? (55)

R: Os processos nunca tem um tempo certo para terminar pois dependem muito do estado das variáveis do mesmo e de todos os processos a concorrer pelo processador.

12: Para cada um dos factores que contribuem para o tempo gasto para transmitir uma mensagem entre dois processos por um canal de comunicação, cite quais as medidas que

seriam necessárias para estabelecer um limite para sua contribuição no tempo total. Por que essas medidas não são tomadas nos sistemas distribuídos de propósito geral? (55-56)

R:

13: O serviço NTP pode ser usado para sincronizar relógios de computador. Explique por que, mesmo com esse serviço, nenhum limite é garantido para a diferença entre dois relógios. (55-56)

R: Mesmo que dois computadores sejam actualizados no mesmo espaço de tempo com tempos iguais estes distanciam-se no tempo pois nunca contam e modo igual. Para garantir que estes estivessem sempre sincronizados era necessário que estivessem constantemente a consultar o NTP e actualizar os seus relógios para garantir o mínimo atraso possível.

14: Considere dois serviços de comunicação para uso em SD assíncronos. No serviço A, as mensagens podem ser perdidas, duplicadas ou retardadas e somas de verificação se aplicam apenas aos cabeçalhos. No serviço B, as mensagens podem ser perdidas, retardadas, ou entregues rápido demais para o destinatário manipular-las, mas sempre chegam com o conteúdo correcto.

Descreva as classes de falhas exibidas para cada serviço. Classifique suas falhas de acordo com seu efeito sobre as propriedades de validade e integridade. O serviço B pode ser descrito como um serviço de comunicação confiável?(59-60),(62-63)

R: Classes de Falhas:

Parada por falha - A, B

Colapso - A,B

Omissão - A,B

Omissão de envio - A, B

Omissão de recepção -

Arbitrária - A, B

Penso que um serviço que apresente as mesmas falhas que o anterior mas que a mensagem chega sempre em condições é mais fiável do que um em que a mensagem nem sempre pode chegar completa.

15: Considere dois processos X e Y que utilizam o serviço comunicação B do ex 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 pedido de X para Y, seguida de Y executando o pedido, seguida de uma mensagem de resposta de Y para X. Descreva as classes de falha que podem ser exibidas por uma invocação. (59-60)

R:

16: Suponha que uma leitura de disco possa às vezes ler valores diferentes dos gravados. Cite os tipos de falha exibidos por uma leitura de disco. Sugira como essa falha pode ser mascarada para produzir uma forma de falha benigna diferente. Agora, sugira como se faz para mascarar a falha benigna (62-63)

R:

17: Defina a propriedade de integridade da comunicação confiável e liste todas as possíveis ameaças à integridade de usuários e dos componentes do sistema. Quais as medidas que podem ser tomadas para garantir a propriedade de integridade diante de cada uma dessas fontes de ameaças. (63-63),(72,73)

R:

18: Descreva as possíveis ocorrências de cada um dos principais tipos de ameaça à segurança(ameaças aos processos, canais de comunicação, negação de serviço) que poderiam ocorrer na Internet. (63-64)

R: