Questões Cap. 2

2.1 Dê três exemplos específicos e contrastantes dos níveis de heterogeneidade cada vez maiores experimentados nos sistemas distribuídos atuais, conforme definido na Seção 2.2. página 39

R: Hardware: os sistemas distribuídos são cada vez mais heterogêneos com PCs (geralmente baseados em Intel), smartphones, nós de sensores de recursos limitados e computadores com cluster de recursos ou processadores de vários núcleos.
Sistemas operacionais: um sistema distribuído pode incluir computadores com Windows, MAC OS, vários tipos diferentes de Unix e também mais sistemas operacionais especializados para smartphones ou nós de sensores.
Redes: a Internet também é cada vez mais heterogênea abraçando tecnologias sem fio e estilos ad hoc de redes.

2.2 Quais problemas você antevê no acoplamento direto entre entidades que se comunicam, que está implícito nas estratégias de invocação remota? Consequentemente, quais vantagens você prevê a partir de um nível de desacoplamento, conforme o oferecido pelo não acoplamento espacial e temporal? Nota: talvez você queira rever sua resposta depois de ler os Capítulos 5 e 6. página 43

R: O cliente está intrinsecamente ligado ao servidor e vice-versa e isso é inflexível em termos de lidar com falha, por exemplo, se o servidor falhar e um servidor de backup assumir o gerenciamento de pedidos. Geralmente, esse nível de acoplamento torna-se difícil de lidar com mudanças.
Clientes e servidores devem existir ao mesmo tempo e, portanto, não é possível operar em ambientes mais voláteis quando qualquer uma das partes pode estar indisponível, por exemplo, desconectada no caso de um nó móvel.
O benefício do desacoplamento espacial é proporcionar maiores graus de liberdade ao lidar com a mudança, por exemplo, se um novo servidor começar a lidar com solicitações.
O benefício do desacoplamento de tempo é permitir que as entidades se comuniquem quando elas podem ir e vir.

2.3 Descreva e ilustre a arquitetura cliente-servidor de um ou mais aplicativos de Internet importantes (por exemplo, Web, correio eletrônico ou News). página 46

R: Exemplo a Web: Os navegadores são clientes de Servidores de Nomes de Domínio (DNS) e servidores web (HTTP). Algumas intranets são configuradas para interpor um servidor proxy. Os servidores proxy cumprem vários propósitos - quando estão localizados no mesmo local que o cliente, reduzem os atrasos da rede e o tráfego da rede. Quando eles estão no mesmo site que o servidor, eles formam um ponto de verificação de segurança e podem reduzir a carga no servidor.

2.4 Para os aplicativos discutidos no Exercício 2.1, quais estratégias de posicionamento são empregadas na implementação dos serviços associados? página 48

R: As páginas principais da Web são mantidas em um sistema de arquivos em um único servidor. As informações na web como um todo é, portanto, particionado entre muitos servidores web. A replicação não faz parte dos protocolos da web, mas um site da Web fortemente usado pode fornecer vários servidores com cópias idênticas do sistema de arquivos relevantes usando um dos meios bem conhecidos para replicar dados com mudança lenta. As solicitações HTTP podem ser multiplexadas entre os servidores idênticos usando o mecanismo de compartilhamento de carga DNS bastante básico. Além disso, os servidores proxy da web suportam a replicação através do uso de réplicas em cache de páginas usadas recentemente e os navegadores suportam a replicação mantendo um cache local de páginas acessadas recentemente.

2.5 Um mecanismo de busca é um servidor Web que responde aos pedidos do cliente para pesquisarem seus índices armazenados e (concomitantemente) executa várias tarefas de Web crawling para construir e atualizar esses índices. Quais são os requisitos de sincronização entre essas atividades concomitantes? página 46

R: A sincronização necessária é algum sistema de locking para os índices de cache. Os clientes estão requisitando informação dele, enquanto o crawler está alimentando-o com informações. Para evitar que o índice seja corrompido, o crawler deve bloquear as áreas do índice em que ele vai modificar / adicionar ou remover entradas.

2.6 Frequentemente, os computadores usados nos sistemas peer-to-peer são computadores desktop dos escritórios ou das casas dos usuários. Quais são as implicações disso na disponibilidade e na segurança dos objetos de dados compartilhados que eles contêm e até que ponto
qualquer vulnerabilidade pode ser superada por meio da replicação? páginas 47, 48

R: Como os sistemas peer-to-peer funcionam de maneira que as maquinas são cliente e servidor ao mesmo tempo, existe uma abertura no firewall do computador que deixa a

máquina mais vulnerável quando está conectada por esse sistema. A integridade do s recursos compartilhados passa a ser um problema sério nesse meio.

2.7 Liste os tipos de recurso local 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. página 49

R: Podem ser vulneráveis tanto o sistema operacional quanto os programas e aplicativos instalados no computador do usuário, os arquivos de dados, textos, planilhas, imagens e outros, assim como os dispositivos locais do computador do usuário e seus dispositivos remotos, aos quais ele esteja conectado

2.8 Dê exemplos de aplicações em que o uso de código móvel seja vantajoso. página 49

R: Podemos ilustrar exemplos, como instalação e atualização de programas pela internet, softwares de auditoria computacional e de gerência de configuração, atualização sob demanda de notícias, imagens multimídia, áudios, entre outros tantos.

2.9 Considere uma empresa de aluguel de carros hipotética e esboce uma solução de três camadas físicas para seu serviço distribuído de aluguel de carros. Use sua resposta para ilustrar vantagens e desvantagens de uma solução de três camadas físicas, considerando problemas como desempenho, mudança de escala, tratamento de falhas e manutenção do software com o passar do tempo. página 53

R: Uma solução de três níveis pode consistir em:

- Um front-end baseado na web que oferece uma interface de usuário para o serviço de aluguel de carros (a lógica de apresentação);
- Um nível médio que suporta as operações principais associadas ao negócio de aluguel de automóveis, incluindo a localização de uma determinada marca e modelo, a verificação da disponibilidade e dos preços, a obtenção de um orçamento e a compra de um carro específico (a lógica da aplicação);
- Um banco de dados que armazena todos os dados persistentes associados ao estoque (a lógica de dados).
  Performasse: Esta abordagem introduz latência extra em que as solicitações devem ir da interface baseada na web para a camada intermediária e, em seguida, para o banco de dados (e voltar). No entanto, a carga de processamento também é distribuída por três máquinas (especialmente sobre a camada intermediária e a base de dados) e isso pode ajudar com o desempenho. Por esta última razão, a solução de três níveis pode escalar melhor. Isto pode ser melhorado por outras estratégias de colocação

complementares, incluindo a replicação.
Falha: Há um elemento extra envolvido e isso aumenta a probabilidade de uma falha ocorrer no sistema. Igualmente, as falhas são mais difíceis de lidar, por exemplo, se a camada intermediária estiver disponível e o banco de dados falhar. A abordagem de três níveis é muito melhor para a evolução devido à separação intrínseca de preocupações. Por exemplo, a camada intermediária contém somente a lógica do aplicativo e, portanto, deve ser mais fácil de atualizar e fazer manutenção.

2.10 Dê um exemplo concreto do dilema apresentado pelo princípio fim-a-fim de Saltzer, no contexto do fornecimento de suporte de middleware para aplicativos distribuídos (talvez você queira enfocar um aspecto do fornecimento de sistemas distribuídos confiáveis, por exemplo, relacionado à tolerância a falhas ou à segurança). página 60

R: O argumento de end-to end de Saltzer afirma que as funções relacionadas à comunicação só podem ser implementadas de forma completa e confiável com o conhecimento e a ajuda da aplicação e, portanto, fornece essa função como uma característica do sistema de comunicação em si (ou middleware) nem sempre é sensata. Um exemplo concreto é a comunicação segura. Suponha que em um dado sistema, o subsistema de comunicação forneça comunicação criptografada. Isso é útil, mas insuficiente. Por exemplo, o caminho da rede para o software de aplicação pode ser comprometido e estar desprotegido. Essa solução também não trata de participantes mal-intencionados na troca de dados. Considere também o intercâmbio confiável de dados implementado pela introdução da proteção de checksum em saltos individuais na rede. Novamente, isso é insuficiente, pois os dados podem ser corrompidos por nós intermediários, por exemplo, gateways ou mesmo nos sistemas finais.

2.11 Considere um servidor simples que executa pedidos do cliente sem acessar 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? página 62

R: A taxa de chegada dos pedidos do cliente é imprevisível. Se o servidor utiliza threads para executar as solicitações em simultâneo, poderá não ser capaz de atribuir tempo suficiente a um pedido específico dentro de um determinado limite de tempo. Se o servidor coloca a fila na solicitação e executa uma de cada vez, eles podem esperar na fila por um tempo ilimitado. Para executar solicitações dentro do tempo limitado, limite o número de clientes de acordo com sua capacidade. Para lidar com mais clientes, use um servidor com mais

processadores. Depois disso, (ou em vez disso) replicará o serviço continuamente. A solução pode ser dispendiosa e, em alguns casos, manter as réplicas consistentes pode levar até ciclos de processamento úteis, reduzindo as que estão disponíveis para executar solicitações.

2.12 Para cada um dos fatores que contribuem para o tempo gasto na transmissão de uma mensagem entre dois processos por um canal de comunicação, cite medidas necessárias para estabelecer um limite para sua contribuição no tempo total. Por que essas medidas não são tomadas nos sistemas distribuídos de propósito geral atuais? página 63

R: A transmissão de uma mensagem envolve fatores como latência de propagação, largura de banda, e sobrecarga de processamento. Para estabelecer limites, pode-se otimizar a largura de banda, reduzir a latência com otimizações de rede e usar protocolos eficientes. Nos sistemas distribuídos de propósito geral, essas medidas nem sempre são tomadas devido à variabilidade das condições de rede, a imprevisibilidade da carga de trabalho e a complexidade do gerenciamento de recursos distribuídos, o que torna difícil garantir limites rígidos de tempo.

2.13 O serviço Network Time Protocol pode ser usado para sincronizar relógios de computador. Explique por que, mesmo com esse serviço, nenhum limite garantido é dado para a diferença entre dois relógios. página 64

R: Qualquer cliente que utilize o serviço NTP deve comunicar com ele por meio de mensagens passadas por um canal de comunicação. Se um limite for definido no momento que transmitimos uma mensagem através de um canal de comunicação, então a diferença entre o relógio do cliente e o valor fornecido pelo serviço NTP também seria limitada. Com um tempo de transmissão de mensagens limitado, as diferenças de relógios são necessariamente ilimitadas

2.14 Considere dois serviços de comunicação para uso em sistemas distribuídos 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 manipulá-las, mas sempre chegam com o conteúdo correto. Descreva as classes de falha 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?

R: O Serviço A pode ter:
Falhas arbitrárias:

- como os checksums não são aplicados nos corpos das mensagens, estes podem ser corrompidos.
- Mensagens duplicadas, Falhas de omissão (mensagens perdidas).
  Como é usado e um sistema distribuído assíncrono, ele não pode sofrer de falhas de temporização.
  Validade - é negada por causa das mensagens perdidas;
  Integridade - é negada por causa das mensagens corrompidas e mensagens duplicadas.
  O serviço B pode ter:
  Falhas de omissão (mensagens perdidas, mensagens descartadas).
  Como o sistema distribuído no qual ele é usado é assíncrono, ele não pode sofrer de falhas de temporização.
  Ele passa o teste de integridade, mas não o teste de validade, portanto, não pode ser chamado confiável.

  2.15 Considere dois processos, X e Y, que utilizam o serviço de comunicação B do Exercício 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 requisição de X para Y, seguida de Y executando
  a requisição, seguida de uma mensagem de resposta de Y para X. Descreva as classes de falha que podem ser exibidas por uma invocação.

R: Uma invocação pode sofrer as seguintes falhas: Falhas de crash: X ou Y pode falhar. Portanto, uma invocação pode sofrer falhas de crash. Falhas de omissão: como o Serviço B sofre de falhas de omissão as mensagens de solicitação ou resposta podem ser perdidas.

2.16 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.

R: Uma leitura básica de disco apresenta falhas arbitrárias. Isso pode ser mascarado usando checksum em cada bloco de disco (tornando improvável que valores errados não sejam detectados). Quando um valor incorreto é detectado, a leitura retorna num valor em vez de um valor errado.
As falhas de omissão podem ser mascaradas replicando cada bloco de disco em dois discos independentes o que torna falhas de omissão improváveis.

2.17 Defina a propriedade de integridade da comunicação confiável e liste todas as possíveis ameaças à integridade de usuários e de componentes do sistema. Quais medidas podem ser

tomadas para garantir a propriedade de integridade diante de cada uma dessas fontes de ameaças?

R: Integridade: a mensagem recebida é idêntica à enviada e nenhuma mensagem é entregue mais de uma vez.

Ameaças dos usuários: mensagens falsas, reproduzir mensagens antigas, alterar mensagens durante a transmissão.

Ameaças de componentes do sistema: -mensagens podem ser corrompidas no caminho.

- Mensagens podem ser duplicadas por protocolos de comunicação que retransmitem as mensagens.
  Medidas que podem ser adotadas:
- Para ameaças de usuários - use canais seguros ou técnicas de autenticação. - Para ameaças de componentes do sistema. Checksums para detectar mensagens corrompidas - mas depois temos um problema de validade (mensagem perdidas). As mensagens duplicadas podem ser detectadas se números de sequência forem anexados a mensagens.

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

R: Ameaças aos processos: sem autenticação servidores, existem muitas ameaças. Um inimigo pode acessar os arquivos ou caixas de correio de outros usuários ou configurar servidores 'spoof'. Por exemplo. Um servidor pode ser configurado para 'spoof' serviço de um banco e receber detalhes de transações financeiras do usuário.
Ameaças aos canais de comunicação: IP spoofing - enviar solicitações para servidores com um endereço de fonte falsa, ataques man-in-the-middle.
Negação de serviço: inundar um serviço publicamente disponível com mensagens irrelevantes