Ciro Guilherme Nass

Questão 1. Quais as vantagens e desvantagens dos sistemas distribuídos em relação aos sistemas centralizados?

Vantagens:

- Múltiplos computadores chamados de “nós”, cada um tem sua própria capacidade de processamento e armazenamento;
- Possuem uma escalabilidade maior assim como tolerância a falhas, compartilhamento de recursos e concorrência, por conta da quantidade de máquinas atuando na composição de um só sistema;
- Desempenho melhor por conta de cada máquina atuar como se fosse um núcleo de um processador, analogicamente, reduzindo seu tempo de processamento;
- Latência baixa por conta da distribuição de dados, permitindo assim um grande fluxo de dados entre as máquinas.

Desvantagens:

- Complexo demais por ser uma rede de dispositivos conectados entre si, ainda mais quando levamos em conta as diferenças de hardware, para a sincronização e coordenação;
- Segurança, manter todos os componentes da rede seguros de qualquer ameaça é algo desafiador, pois aumenta as chances de um deles ser atacado;
- Os custos de operação, são elevados, pois entram no cálculo a energia, manutenção e atualização de software e hardware caso preciso.

Questão 2. Cite cinco exemplos de sistemas distribuídos.

1. Criptomoedas;
2. Servidores da web;
3. Kubernetes;
4. Sistemas de computação em nuvem;
5. Redes peer-to-peer (P2P).

Questão 3. Classifique e defina as transparências desejáveis em sistemas distribuídos.

1. Transparência de Acesso
    - O sistema oculta as diferenças entre as interfaces dos recursos locais e remotos.

2. Transparência de Localização
    - Os usuários não precisam saber onde os recursos estão localizados.
3. Transparência de Migração
    - Permite que recursos sejam movidos entre diferentes locais sem impactar os usuários ou aplicações.
4. Transparência de Replicação
    - Oculta do usuário a existência de múltiplas cópias de um recurso, garantindo consistência.
5. Transparência de Concorrência
    - Garante que vários usuários ou processos possam acessar os recursos simultaneamente, sem interferir uns nos outros.
6. Transparência de Falhas
    - Oculta as falhas de hardware e software, garantindo que o sistema continue funcionando.
7. Transparência de Escalabilidade
    - Permite que o sistema cresça em número de usuários, dados ou nós sem perda significativa de desempenho.

Questão 4. Defina tolerante a faltas?

É um sistema resiliente que continua funcionando mesmo quando falhas acontecem, mesmo que com um desempenho afetado e entrega parcial de serviço.

Questão 5. Quais são e como são medidos os parâmetros de desempenho?

1. Latência
    - É o tempo total necessário para uma mensagem ou operação ser enviada, processada e respondida no sistema em segundos;
2. Vazão (Throughput)
    - É a quantidade de operações ou dados processados pelo sistema em um determinado período. Normalmente medido em operações por segundo ou bytes por segundo;
3. Escalabilidade
    - Capacidade do sistema de aumentar o desempenho proporcionalmente ao crescimento de recursos. Medido por número de nós ou recursos adicionados versus ganho de desempenho;
4. Tempo de Resposta
    - É o tempo total entre o envio de uma requisição e a resposta recebida pelo usuário. Em milissegundos ou segundos;
5. Uso de Recursos

- Representa o quanto dos recursos do sistema (CPU, memória, rede, armazenamento) está sendo utilizado. Medido Em percentual de uso (%), como uso de CPU ou memória;

6. Disponibilidade
    - Percentual de tempo em que o sistema está **operacional** e acessível aos usuários. Medido por uma proporção (ex: 99.99%) ou como o tempo de inatividade (downtime).
7. Confiabilidade
    - Capacidade do sistema de funcionar corretamente por longos períodos sem falhas. Medido pelo tempo médio entre falhas (MTBF);
8. Consistência
    - Garantia de que os dados em um sistema distribuído permanecem coerentes em todos os nós. Medido por uma comparação entre os estados dos dados replicados em diferentes locais.
9. Resiliência (Tolerância a Falhas)
    - Capacidade do sistema de continuar funcionando mesmo em caso de falhas. Medido pelo tempo de recuperação (Recovery Time Objective - RTO) ou percentual de disponibilidade após falhas.

Questão 6. Classifique e defina os tipos de rede.

PAN (Personal Area Network), Rede de curta distância projetada para dispositivos pessoais em um único ambiente; LAN (Local Area Network), Rede restrita a uma área física pequena, como um escritório, laboratório ou casa; MAN (Metropolitan Area Network), Rede que conecta várias LANs dentro de uma área urbana ou cidade; WAN (Wide Area Network), Rede que conecta diferentes regiões, estados ou até países; SAN (Storage Area Network), Rede dedicada ao armazenamento de dados em alta velocidade; WLAN (Wireless Local Area Network), Uma variação da LAN, mas utiliza comunicação sem fio (Wi-Fi); VPN (Virtual Private Network), Rede que cria uma conexão segura e criptografada entre usuários e redes públicas ou privadas; Redes Peer-to-Peer (P2P), Todos os dispositivos conectados possuem status igualitário (sem servidor central); Redes Cliente-Servidor, Uma arquitetura onde há dispositivos clientes que solicitam serviços a servidores; Redes Definidas por Software (SDN), Redes em que o controle é separado do hardware físico e gerido por software; Redes Unicast, Comunicação de um para um. Um nó transmite informações diretamente para outro nó; Redes Multicast, Comunicação de um para muitos, mas apenas para um grupo específico de nós; Redes Broadcast, Comunicação de um para todos. Um nó transmite para todos os nós na rede; Redes Cabeadas, Redes que utilizam cabos

físicos para transmitir dados; Redes Sem Fio, Redes que utilizam ondas de rádio, micro-ondas ou luz infravermelha.

Questão 7. Numa comunicação entre processos, quais são os elementos básicos da comunicação que devem ser considerados?

Os elementos a serem considerados são: Processos Comunicantes, Mecanismo de Comunicação, Modelo de Comunicação, Endereçamento, Dados a Serem Transmitidos, Protocolo de Comunicação, Confiabilidade, Desempenho, Concorrência e Sincronização, Segurança e Falhas e Recuperação.

Questão 8. O que é Marshalling e Unmarshalling?

Marshalling transforma um objeto ou estrutura em uma representação como um fluxo de bytes que possa ser transmitida ou armazenada; já o Unmarshalling, inverso ao Marshalling, reconstrói um objeto ou estrutura a partir de dados transmitidos.

Questão 9. As formas de endereçamento normalmente usadas em sistemas distribuídos são: (a) endereçamento máquina.processo; (b) descoberta de endereço via broadcast; (c) descoberta de endereço via um servidor de nomes. Discuta sobre os potenciais problemas de cada uma dessas formas.

a) Endereçamento máquina.processo
    a. Acoplamento forte - depende do endereço de máquina e do processo, o que cria o acoplamento de componentes
    b. Dificuldade de escalabilidade – gerenciar manualmente grandes sistemas pode ser impraticável
    c. Problemas de falhas – se acontecer uma falha a sua recuperação pode ser difícil.
    d. Mobilidade limitada – não suporta migração de máquinas pois o endereço é fixo
b) Descoberta de endereço via broadcast
    a. Uso excessivo da rede – consome grande banda larga pois mensagens são enviadas a todos os dispositivos
    b. Latência – o tempo de localizar um endereço pode ser alto, tornando os processos suscetíveis a ataques
    c. Segurança – Broadcast pode ser interceptado como dito no item acima
    d. Confiabilidade – Processos fora do tempo podem ser inacessíveis
c) Descoberta de endereço via um servidor de nomes
    a. Ponto único de falha – se falhar, todos os processos ficarão incapazes de localizar os endereços

b. Sobrecarga no servidor – se houver muitas solicitações de uma vez, pode perder o desempenho
c. Latência adicional – o uso de um servidor de nomes intermediário adiciona um atraso adicional na comunicação
d. Segurança – sendo um alvo crítico, se comprometido pode redirecionar os processos a endereços maliciosos

Questão 10. Qual a diferença entre comunicação síncrona (bloqueante) e assíncrona (não bloqueante)? Comente como resolver a problemática encontrada nas primitivas não-bloqueantes.

Na comunicação síncrona o emissor ou o receptor fica bloqueado até que a operação de envio ou recebimento seja concluída, o que é diferente na comunicação assíncrona, onde não há este bloqueio, portanto o emissor pode enviar quantas vezes quiser antes do primeiro ser recebido. Como na assíncrona não há garantida na entrega dos pacotes, temos o problema nas primitivas não-bloqueantes, tem-se algumas alternativas de como resolver este problema:

- Confirmação de Recebimento (Acknowledgment - ACK)
  - O receptor envia uma mensagem de confirmação ao emissor para garantir que os dados foram recebidos.
- Retries e Timeouts
  - Implementar retransmissões automáticas em caso de falha ou ausência de confirmação dentro de um prazo (timeout).
- Ordenação de Mensagens
  - Utilizar identificadores de sequência nas mensagens para que o receptor possa reordená-las caso cheguem fora de ordem.
- Buffers Controlados
  - Uso de promises, callbacks ou futuras para tratar a resposta no momento em que os dados estiverem disponíveis.
- Protocolos Robustecidos
  - Utilizar protocolos confiáveis que encapsulam os problemas típicos da comunicação assíncrona, como AMQP (Advanced Message Queuing Protocol) ou Kafka para filas de mensagens.