

UNIVERSIDADE FEDERAL FLUMINENSE
INSTITUTO DE COMPUTAÇÃO
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Sistemas Distribuídos 2019.1

Profa.: Simone Martins

Gabarito Lista 4

1. Dê motivos para replicação de dados.

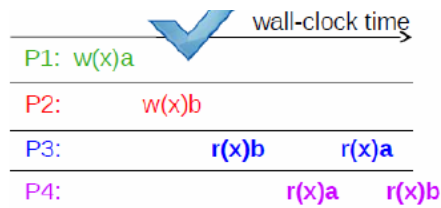
Resposta: Dados são replicados para aprimorar confiabilidade ou melhorar desempenho. No caso de aprimorar confiabilidade, quando um repositório falha, temos que um outro repositório pode continuar permitindo o acesso ao dado. No caso de melhorar o desempenho, se um dado está armazenado em um local que demora a ser transmitido para o requerente de dados, o dado pode ser replicado para um local mais perto do requerente.

2. Explique o compromisso entre manutenção de cópias atualizadas e desempenho de replicação.

Resposta: Para se manter todas as cópias atualizadas, necessita-se que muitas mensagens sejam trocadas entre os processos e os processos necessitam gastar tempo de processamento para executar o compromisso de manutenção de atualização das cópias. Pode-se evitar sincronização global (instantânea) diminuindo-se o número de mensagens e processamento gasto para executar esta manutenção, mas, neste caso, as cópias nem sempre são iguais em todos os lugares

3. Dê um exemplo de operações de escrita e leitura onde a consistência causal é obedecida, mas a consistência sequencial não.

Resposta:



Neste caso, temos consistência causal porque podemos ter a sequência $w(x)b, r(x)b, w(x)a, r(x)a$ em P3 e $w(x)a, r(x)a, w(x)b, r(x)b$ em P4. Mas não é possível ter a mesma sequência global que será executada em P3 e P4 que explique $r(x)b$ ocorrer antes de $r(x)a$ em P3 e $r(x)a$ ocorrer antes de $r(x)b$ em P4.

4. Suponha um item de dado x que é incrementado com o tempo. Dê um exemplo de situação onde, caso não seja utilizado um protocolo de consistência adequado, um cliente poderia escrever um valor $x = x + 1$ em item de dado x e ler esse mesmo item de dado x com valor $x - 1$.

Resposta: Caso um cliente incremente localmente o valor x , e ele receba posteriormente uma escrita que chegou atrasada de uma outra réplica para o valor x , ele pode ler o valor de x novamente como $x-1$.

5. Explique o que são modelos de consistência centrado em cliente.

Resposta: Consistência centrada no cliente dá a um único cliente uma garantia de consistência de acesso a um depósito de dados.

6. Descreva uma implementação simples para prover consistência de leitura de suas escritas para exibir páginas da Web que acabaram de ser atualizadas.

Resposta: A implementação mais simples é permitir que o navegador sempre verifique se

está exibindo a versão mais recente de uma página. Isso requer o envio de uma solicitação para o servidor da Web.

7. Lembrando de uma maneira de implementar consistência centrada em clientes descrita em sala de aula. Cada operação de escrita possui um único identificador global gerado pelo servidor que executa a operação de escrita. Cada servidor que executa uma escrita registra esta operação de forma que ela possa ser executada em outro servidor. Cada cliente mantém dois conjuntos com identificadores de escrita: (i) Conjunto de leitura: contém os identificadores das operações de escrita relacionadas às leituras realizadas, (ii) Conjunto de escrita: identificador das escritas realizadas pelo cliente. Explique como se pode implementar um sistema com consistência de leitura monotônica e um com consistência leia suas escritas utilizando esse esquema.

Resposta: No caso da leitura monotônica, quando um cliente solicita uma leitura, o servidor verifica o conjunto de leitura do cliente para verificar se todas as operações de escrita até o momento foram realizadas localmente no servidor. Se existem operações de escrita que foram executadas em outros servidores, o servidor contacta os outros servidores para assegurar que todas as operações de escrita sejam executadas localmente no servidor. Após a leitura, o conjunto de leitura do cliente é atualizado com as escritas relevantes para o dado que foi lido. No caso de consistência leia suas escritas, antes de um servidor executar uma solicitação de leitura, o servidor obtém dos outros servidores todas as operações de escrita do conjunto de escrita do cliente.