

## Cap 7 exercicios

1) As vantagens:

- Maior capacidade de garantir a continuidade das operações
- Redução de riscos
- Segurança da informação e Confiabilidade
- Melhoria no desempenho das aplicações
- Disponibilidade
- Redução de tráfego de redes em horário nobre

2) Os problemas são a complexidade e custo de atualização porque quanto maior a complexidade do software, mais alto será o custo então quando fazemos atualizações no sistema, precisamos sempre atualizar as replicas e se você pensar em uma situação real onde temos o mesmo sistema em vários cantos do mundo, toda vez que atualizamos um desses sistemas precisamos atualizar todas as replicas e isso custa tempo e dinheiro. Na redundância de dados, quanto mais dados forem replicados, mais tempo adicional é requerido para as atualizações. No Espaço adicional para armazenamento, em relação a replicação, quanto mais dados forem replicado, maior vai ser a necessidade de espaço para o armazenamento dos dados

3)Consistência Sequencial: Todos os processos executam todas as operações de leitura e escrita em uma mesma ordem global, e as operações em um processo aparecem na ordem que são emitidas pelo programa que está sendo executado pelo processo. Exemplo:

ex:

P1: w(x)a  
P2:       w(x)b  
P3:           r(x)b   r(x)a  
P4:               r(x)a   r(x)a

Consistência Causal: Representa um enfraquecimento da consistência sequencial no sentido que faz distinção entre eventos que são potencialmente relacionados por causalidade e os que não são

ex:

P1: w(x)a  
P2:       r(x)a   w(x)b  
P3:           r(x)a   r(x)b  
P4:               r(x)a   r(x)b

O modelo de consistência sequencial é mais estrito no sentido que ele define que todos os processos tem que ver as mesmas operações na mesma ordem, enquanto o modelo de consistência causal diminui isso para os que são potencialmente relacionados. Isso é menos estrito.

4) A motivação geral é do mesmo modo que agrupamos dados, agrupar operações pode ser benéfico na redução de troca de mensagens entre as réplicas. De modo que as leituras e escritas não sejam conhecidas por todos os processos, mas apenas o resultado final das operações.

5) Consistência eventual: Deixa as atualizações acontecerem, e se preocupa com a sua ordenação depois. Este é uma forma específica de consistência fraca, o sistema de armazenamento garante que se nenhuma atualizações novas sejam feitas para o objeto, eventualmente todos os acessos

retornarão o ultimo valor atualizado. Se não ocorrer falhas, o tamanho máximo da janela de inconsistência pode ser determinado baseado nos fatores tais como os atrasos de comunicação, a carga no sistema e o número de réplicas envolvidas no esquema de replicações.

Exemplos: sincronizadores de arquivos, Amazon's Dynamo, Bayou

6) A motivação geral para modelos de Consistência Centrados em Dados:

- Visam fornecer uma visão consistente de um depósito de dados no âmbito de um sistema
- Processos concorrentes podem estar atualizando um depósito de dados simultaneamente

7) Temos esses dois modelos. O baseado na diferença dos valores dos dados assume que temos um valor máximo que podemos ter de diferença entre os valores, então imagina que meu limite é 60, se eu tiver varios valores sendo atualizados e fazemos a diferença entre eles, essa nova atualização nao pode ficar acima de 60

Já a consistência de mensagens não recebidas, no caso imagina que o sistema no máximo pode ter 3 então se ocorrer uma atualização, eu nao preciso enviar por enquanto essa atualização para o sistema mas quando chega na 3 atualização entao chega no limite e preciso atualizar o sistema