

Exercícios de Fixação – Chapter 07

Respostas

- 1) O resultado de qualquer execução é o mesmo como se as operações (read e write) sobre o data store, realizadas por todos os processos distribuídos, fossem executadas na mesma ordem por todos os processos igualmente.
- 2) As operações de leitura no item x, nos processos P3 e P4, não são consistentes, ou seja, a mesma sequência em um deveria ser vista no outro, independente da sequência (a e b ou b e a, porém em ambos).
- 3) Na consistência causal, a ordem das operações vistas em todos os processos deve ser a mesma indicada pela relação de causalidade. Já na consistência sequencial, não importa a ordem das operações, desde que a ordem seja a mesma em todos os processos.
- 4) Porque a ordem das operações (c-b e b-c) difere entre os processos P3 e P4.
- 5) Porque dado que o valor do item x foi alterado para “b” depois da sua leitura como “a”, não é possível que em P3 esse valor fosse primeiro “b” e depois “a”.
- 6) A diferença é o escopo da garantia de consistência. Na consistência centrada em dados, o objetivo é fornecer uma visão consistente do data store para todos os processos do sistema, enquanto na consistência centrada no cliente o foco é garantir uma visão consistente dos dados apenas para um único processo cliente.
- 7) O modelo é o monotonic reads, em que operações sucessivas de reads seguirão a mesma ordem em que foram emitidas. O artigo discute que uma forma de garantir isso, para um cliente específico, é direcionar as operações de read, sucessivas, para um mesmo servidor.
- 8) Células muito pequenas (too small) não são apropriadas pois acabem exigindo mais de uma célula para um mesmo grupo de hosts, o que implica em ter dois servidores (um em cada célula) com o mesmo papel. Já células muito grandes (too large) têm o efeito contrário, ou seja, em uma única célula tem-se grupos distintos de hosts; o problema é que cada célula normalmente tem apenas um servidor e nesse caso exigiria dois servidores distintos, um para atender cada grupo, ou seja, um dos grupos precisaria acessar seu servidor fora da célula (“mais distante”). Células com o tamanho adequado (just right) são aquelas que conseguem acomodar bem os hosts de um grupo e, portanto, seu servidor atenderia todos esses hosts.
- 9) Réplicas permanentes são aquelas definidas a priori, ou seja, antes do início do funcionamento do sistema. Essas réplicas ficam mais distantes dos usuários. Normalmente são usadas para fins de balanceamento de carga e alta-disponibilidade e estão todas no mesmo data center. Ao contrário das réplicas permanentes, as réplicas iniciadas pelos clientes são criadas dinamicamente e são as mais

próximas do usuário; seu objetivo é performance, ou seja, reduzir o tempo de acesso do cliente aos dados da réplica. Réplicas iniciadas por servidores, assim como as iniciadas pelo cliente, são criadas dinamicamente e se localizam em posição intermediária em relação as réplicas permanentes (mais distantes do cliente) e iniciadas pelo cliente (mais próximas do cliente).

- 10) No protocolo de consistência sequencial primary-based, cada item de dado no data store tem um servidor primário associado, o qual é um processo responsável por coordenar as atualizações (operações de write) sobre o item de dado; o servidor primário pode ser local ou remoto ao cliente iniciando a operação de write. Já os protocolos de consistência sequencia baseados em replicação, as operações de write são executadas nos diferentes servidores onde se encontram as réplicas do item de dado atualizado; esta abordagem pode ser do tipo ativa, onde as requisições de atualização são enviadas a todas as réplicas, ou do tipo quorum-based, em que as operações de atualização são enviadas apenas para um subset do total servidores de réplica.