

Disciplina DCE540 - Computação Paralela e Distribuída	Método de entrega Moodle da disciplina	Data de entrega 21/10/2021 às 23h59
Professor Iago Augusto de Carvalho (iago.carvalho@unifal-mg.edu.br)		

Prova 03

Cada aluno deverá submeter um único arquivo .pdf com a resolução da prova.

A prova pode ser realizada de duas maneiras:

- Com papel e caneta, sendo posteriormente escaneada e enviada
- Digitada em algum editor de texto, e.g., Word ou LaTeX

A prova deverá ser entregue no Moodle da disciplina até a data limite.

- Atrasos não serão tolerados

Exercício 1 (15 %)

Quais são as razões para termos replicação de dados em sistemas distribuídos?

Exercício 2 (15 %)

Explique os principais problemas que podem ocorrer com dados replicados, especialmente aqueles problemas relacionados a consistência.

Exercício 3 (15%)

As operações abaixo respeitam o modelo de consistência sequencial? Diga se sim ou não e justifique

P1:	W(x)a	
P2:	R(x)NIL	R(x)a

Exercício 4 (20%)

Desenhe uma sequência de operações que respeitam o modelo de consistência casual, mas não respeitam o modelo de consistência sequencial. Dê um exemplo bem diferente daquele apresentado em aula.

Exercício 5 (15%)

Apresente as definições de

- Réplicas permanentes
- Réplicas inicializadas pelo servidor (ou pelo sistema distribuído)
- Réplicas inicializadas pelos clientes

Exercício 6 (20%)

Considere uma implementação válida do protocolo de atualização baseada em votos na qual N_R (quórum de leitura) é igual a 1 e temos que N_W (quórum de escrita) é igual a N , onde N é o número de réplicas. Diga se esta implementação é válida ou não. Além disso, caso seja válida, apresente suas vantagens e desvantagens.

Gabarito

Exercício 1

Existem duas principais razões para a replicação de dados em sistemas distribuídos: (i) Tolerância a falhas e segurança; e (ii) performance escalabilidade.

Por tolerância a falhas e segurança, entende-se que um sistema distribuído cujos dados sejam replicados tem maior facilidade em se recuperar de uma eventual falha, migrando o acesso aos dados a uma réplica e podendo utilizar estas réplicas como *backup* e recuperar os pontos falhos.

Por performance e escalabilidade, entende-se que com a replicação de dados é possível acrescentar novos usuários e processos ao sistema distribuído sem que ocorra uma grande sobrecarga no uso do sistema. A replicação também é utilizada para diminuir o tempo de acesso aos dados (*delay*, *ping*) para usuários de diferentes locais geográficos, além de realizar o balanceamento de carga (quando existe um grande número de acessos a um mesmo conjunto de dados).

Exercício 2

O principal problema é relativo a consistência dos dados. Quando um dado é modificado em algum lugar, esta modificação deve ser propagada a todas as suas réplicas como forma de garantir a consistência das operações do sistema distribuído.

Em sistemas de larga escala, esta propagação não é instantânea, o que pode levar a erros de consistência em algumas operações, os quais devem ser considerados pelo *middleware* do sistema distribuído.

Esta necessidade de atualização das réplicas também gera um grande tráfego de rede. Este fato pode levar ao congestionamento da rede de um sistema distribuído caso os arquivos sejam muito grandes e a frequência de atualização seja alta.

Exercício 3

Sim, elas respeitam o modelo de consistência sequencial, pela própria definição do modelo. A leitura $R(x)NIL$ (ler um dado vazio na variável x) é ocasionada pela demora na replicação dos dados e nada interfere com o modelo de consistência sequencial.

Exercício 4

Basicamente, qualquer sequência de leituras que possuam uma sub-estrutura parecida com a abaixo respeita o modelo de consistência casual, mas não a sequencial.

R(X)B		R(X)A	
	R(X)A		R(X)B

Exercício 5

- Réplicas permanentes são aquelas criadas junto com o próprio sistema distribuído. Normalmente elas são utilizadas como cópias de segurança dos dados e também visam oferecer um pequeno balanceamento de carga das ações. Estas réplicas permanentes costumam ser poucas e, muitas vezes, servem como espelho dos dados do sistema distribuído.
- Em sua maioria, são réplicas temporárias utilizadas para realizar o balanceamento de carga. Estas réplicas são criadas pelo sistema distribuído quando existe um número excessivo de acesso a um conjunto de dados (por exemplo, durante a realização de eventos promocionais). Além disso, estas réplicas costumam ser somente leitura (*read-only*) e seu único objetivo é prover um serviço a um grande número de clientes
- Estas são cópias *cache* que cada cliente pode manter em seu próprio dispositivo (ou em algum dispositivo de rede próximo a ele, e.g., um roteador LAN). Este tipo de réplica é muito útil quando um

cliente faz um grande número de requisições a um mesmo conjunto de dados para diminuir o tempo de acesso e leitura. Estas réplicas, assim como as inicializadas pelo servidor, são temporárias. Além disso, a atualização dos dados de uma réplica inicializada pelo cliente é do próprio cliente e não do sistema distribuído.

Exercício 6

Este tipo de implementação é chamado de *Read-One, Write-All* (ROWA). Ela é válida, pois temos que os valores de N_R e N_W respeitam ambas as restrições exibidas em aula. A principal vantagem é que é muito fácil obter acesso a um dado, senão que somente é necessário ter o aceite de um servidor de réplicas. Já a desvantagem é que é muito custoso escrever (atualizar) algum dado, pois é necessário obter permissão de todos os servidores de réplicas. Entretanto, isto garante que não exista nenhum conflito de escrita.