

Disciplina DCE540 - Computação Paralela e Distribuída	Método de entrega Moodle da disciplina	Data de entrega 01/04/2022 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 diferenças entre erro, falha e falta? Escolha um componente de um sistema distribuído (um computador, um banco de dados, uma impressora, um software, um meio de comunicação, um serviço em nuvem, dentre outros) e apresente um erro, uma falha e uma falta para este componente.

Exercício 2 (20%)

Considere uma implementação 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.

Exercício 3 (15 %)

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

Exercício 4 (15%)

Qual modelo de consistência é implementado pelas operações abaixo? Explique como a consistência é obtida neste caso

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

Exercício 5 (15%)

Qual é a utilidade da redundância na tolerância a falhas? Tome como base o sistema acadêmico da UNIFAL. Como você implementaria redundância de informação e redundância física neste sistema?

Exercício 6 (20%)

Em relação a tolerância a falhas, apresente as definições de

- a) Disponibilidade
- b) Confiabilidade
- c) Segurança
- d) Manutenibilidade

Gabarito

Exercício 1

Uma **falha** é quando um componente deixa de ser útil ou funcional, isto é, quando ele não consegue mais cumprir suas tarefas. Já um **erro** refere-se ao estado (ou parte) do componente que deu origem a falha. Já uma **falta** é o que levou ao erro.

Vamos tomar como exemplo de componente uma impressora em rede, conectada a um sistema distribuído utilizando um cabo de rede. Pode-se dizer que a impressora falhou quando ela deixa de responder apropriadamente as requisições a ela enviadas. Por exemplo, quando usuários enviam arquivos para impressão e ela não consegue mais imprimir. Um possível erro referente a esta falha pode estar localizado nos cartuchos de tinta da impressora, sendo que a falta relacionada é o esgotamento dos cartuchos.

Exercício 2

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.

Exercício 3

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 4

Este é um modelo de consistência sequencial. Neste modelo, ocorre uma demora na replicação dos dados: note que a leitura de $R(x)NIL$ (ler um dado vazio na variável x) por $P2$ é obtida mesmo após o processo de escrita em $P1$. Entretanto, alguns momentos após, os dados são replicados e $P2$ consegue acessar o correto valor da variável x . Entretanto, este fato está de acordo com o modelo de consistência sequencial, pois se as operações de todos os processos fossem executadas na mesma ordem sequencial, então as operações de cada processo apareceriam nessa sequência na ordem especificada pelo seu programa.

Exercício 5

Redundância é útil para fazer com que as falhas de um sistema distribuído (ou de um componente) sejam transparentes a seus usuários. No sistema acadêmico da UNIFAL, poderíamos implementar redundância de informação ao prover códigos de verificação de erros em todas as requisições e respostas com origem ou destino no sistema acadêmico. Já a redundância física poderia ser atendida ao criarmos cópias físicas dos servidores que hospedam o sistema acadêmico da UNIFAL junto de um *middleware* inteligente que fosse capaz de tratar as falhas destes diferentes servidores.

Exercício 6

- a) Diz que o sistema (ou componente) tem que estar pronto para uso sempre que for requisitado por um usuário. Ou seja, ele tem que estar pronto para uso imediato sempre que for requisitado.

- b) Diz que o sistema (ou componente) tem que funcionar continuamente sem interrupção do serviço. Confiabilidade faz referência ao espaço de tempo que o sistema (ou componente) opera sem apresentar nenhuma falha.
- c) Diz que, no caso de uma falha, nenhum evento catastrófico pode ocorrer. Um sistema seguro é aquele cujas falhas não resultem em eventos extremamente graves a seus usuários ou a outras pessoas.
- d) Refere-se ao tempo de recuperação de um sistema (ou componente) após uma falha e a facilidade de recuperação do mesmo após a falha.