**Sistemas Distribuídos**

**PROJETO: Acesso MRMW**

**(“Multiple Readers, Multiple Writers”)**

Individual ou dupla

Postagem no Canvas até: 08/11/2020

O acesso simultâneo de leitura e escrita pode proporcionar mais desempenho, disponibilidade e tolerância a falhas às aplicações.

Este projeto consiste na implementação de algoritmos que permitam esse tipo de acesso em um sistema distribuído, permitindo o paralelismo e preservando a consistência dos dados replicados em mais de um servidor.

**Projeto básico – CONSISTÊNCIA [3,0 pontos]**

Implemente um programa que permita o acesso paralelo de leitura e escrita a pelo menos uma área de dados, que deve estar replicada em pelo menos dois servidores. Os servidores não devem restringir o acesso dos processos leitores nem dos processos escritores, e ainda assim manter a consistência dos dados.

O sistema deve permitir o acompanhamento do estado dos dados para verificação da consistência dos mesmos.

**Opcional 1 – INTERFACE [1 ponto]**

Implementar uma interface única (centralizada) para monitoramento do estado das áreas de dados: onde as réplicas estão instaladas, qual o valor dos dados, quando foi realizada a última alteração e por quem.

**Opcional 2 – INATIVIDADE DE SERVIDORES [1 ponto]**

Implementar um mecanismo que permita identificar se há algum servidor inativo.

**Opcional 3 – BALANCEAMENTO DE CARGA [1 ponto]**

Distribuir a carga (número de acessos) uniformemente entre os servidores.

**Opcional 4 – DETECÇÂO E TRATAMENTO DE MULTIDÂO INSTANTÂNEA [2 pontos]**

O sistema deve detectar se a área de dados está sendo muito acessada e, nesse caso, providenciar nova réplica para atender os clientes. Quando a demanda de acesso diminuir, cópias desnecessárias devem ser removidas do sistema.

**Opcional 5 – TOLERÂNCIA A FALHAS [2 pontos]**

No caso de haver servidor inativo, ou se houver menos réplicas da área de dados que o necessário (pelo menos duas), instalar outra réplica dos dados, fazendo com que os clientes a acessem também.

**Opcional 6 – CHECKPOINT E LOG [2 pontos]**

O estado dos dados deve ser salvo periodicamente em um *snapshot*. As transações sobre eles devem ser armazenadas em um arquivo de *redo log*. Caso a última réplica dos dados seja perdida, os mesmos devem ser restaurados através do *checkpoint* gravado e das transações confirmadas.

**Opcional 7 – DIRETÓRIO [1 ponto]**

Implementar um diretório, através do qual todos os processos devem encontrar e acessar os dados.

**Opcional 8 – FALHAS NO DIRETÓRIO [1 ponto]**

Implementar mecanismo de tolerância a falhas do diretório.

**OBSERVAÇÕES**

1. No início do arquivo que contiver o programa, deve haver o seguinte cabeçalho preenchido pelo grupo:

Nomes: RAs:

Opcionais funcionando:

Observações: [opcional]

Valor do Projeto: \_\_\_\_\_\_\_ pontos

1. O trabalho vale até 10 pontos. Os grupos podem implementar quantos opcionais desejarem, mas a nota do projeto será limitada a 10 pontos. Dessa forma, cada grupo deve escolher um conjunto de opcionais, cujos pontos serão somados aos 3 pontos do projeto básico (obrigatório). Cada grupo tem a liberdade de escolher os opcionais que considerar mais convenientes.
2. Os projetos serão apresentado em data a ser definida.
3. Não serão aceitos trabalhos após a data de entrega.