

Programação Distribuída - Trabalho 1

Objetivo

O objetivo geral do trabalho é desenvolver uma aplicação distribuída que modela o acesso concorrente a um arquivo de registros, de tal forma que processos servidores desse recurso tenham que realizar a sincronização do acesso ao mesmo, de acordo com o tipo de operação solicitada por processos clientes.

Descrição

Em um cenário hipotético, três possíveis operações podem ser realizadas sobre um recurso compartilhado (arquivo de registros) sendo essas as operações *inserção*, *eliminação* e *leitura*. Cada operação possui uma necessidade diferente com relação ao acesso do recurso e à sincronização com outros processos que também manipulam o mesmo. Uma operação de leitura apenas examina o conteúdo do arquivo, dessa forma diversas operações de leitura podem executar de forma concorrente. Inserções ocorrem adicionando dados ao final do arquivo, dessa forma estas devem ser mutualmente exclusivas, podendo uma inserção ocorrer em paralelo com um número qualquer de leituras. Eliminações, assim como inserções, devem ser mutualmente exclusivas. Além disso, nenhuma outra operação poderá ocorrer em paralelo com uma eliminação, incluindo leituras.

A sua tarefa é criar uma aplicação composta por diversos processos servidores (por exemplo 5) que possuem capacidade de atender a qualquer uma dessas operações vindas de processos clientes (por exemplo 10). Cada operação solicitada por clientes terá um tempo de processamento de 100ms, sem considerar o tempo de espera para atendimento, e isso deverá ser considerado na

sua implementação (pode ser utilizado um delay entre o início e o término de uma operação).

- Os clientes apenas realizam solicitações a qualquer um dos servidores mantidos em uma lista (em um arquivo de configuração) e aguardam pelo atendimento, sem necessitarem de qualquer sincronização;
- Processos servidores atendem requisições de clientes e sincronizam-se entre si por meio de um protocolo baseado em operações *lock()/acquire()* e *unlock()/release()* genéricas, implementadas em um processo à parte (*lock server*). Esse processo será responsável por manter a informação sobre o estado atual de diversas travas (*locks*);
- Os clientes devem gerar operações de inserção, eliminação e busca aleatoriamente, podendo o grupo definir pesos relativos para cada tipo de operação para cada processo cliente (por exemplo, 70% de buscas, 20% de inserções e 10% de eliminações), sendo que essa informação pode ser passada por linha de comando aos processos clientes ou por meio de um arquivo de configuração.

Qualquer linguagem de programação poderá ser utilizada para o desenvolvimento do trabalho, desde que sejam utilizadas APIs baseadas em RPC ou RMI. É importante que o seu trabalho seja validado em um ambiente distribuído composto diversas máquinas.

Apresentação e entrega

Este trabalho deverá ser realizado em duplas ou trios e apresentado no dia 24/04 ou 26/04 (apresentação em torno de 10 minutos). É importante que os grupos se organizem para a apresentação na primeira data, pois a segunda data será apenas utilizada pra apresentações dos grupos restantes. Para a entrega, é esperado que apenas um dos integrantes envie pelo Moodle um arquivo *.tar.gz*, contendo o código fonte da implementação.