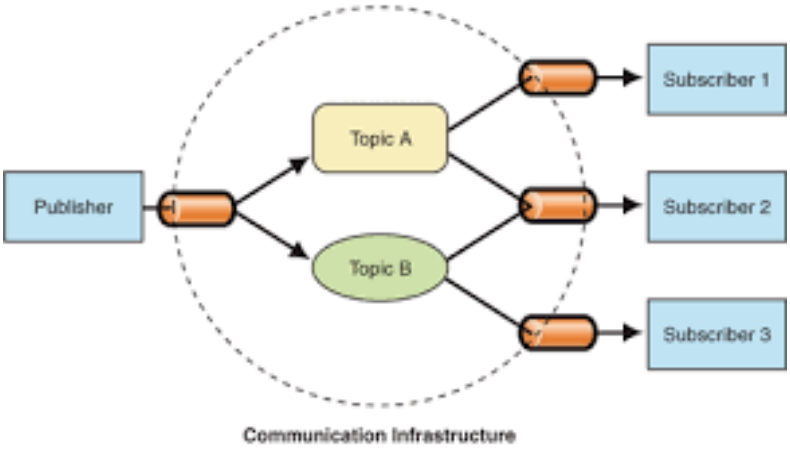
Trabalho Prático 2 (TP 02) – Disciplina de Sistemas Distribuídos

Neste trabalho o aluno utilizará o conceito de SOCKETS, DATAGRAMS e THREADS para construir uma versão de uma solução distribuída. Não há utilização de middlewares neste TP.

Qualquer linguagem será aceita. Não serão aceitas soluções apenas multicore.

Conforme destacado no TP1 em muitas ferramentas, o conceito de publicadores, subscritores e tópicos vêm se tornando comum, portanto amplamente utilizado na indústria. A primeira parte do TP2 consiste em implementar uma solução com publicadores, subscritores e tópicos que seja executada num cluster de máquinas, virtuais ou não. Publicadores criam tópicos e enviam mensagens a tais tópicos, enquanto subscritores se registram em tópicos e passam a receber notificações de mensagens publicadas nos mesmos.



Na segunda parte do trabalho iremos nos concentrar em uma das causas do problema chamado COERÊNCIA ENTRE ESTADOS DE UM MESMO OBJETO. Tal problema pode ser ocasionado pela CONCORRÊNCIA DE ACESSOS aos objetos compartilhados ou pela REPLICAÇÃO dos objetos com o intuito de garantir resiliência ou mesmo melhor desempenho. Neste trabalho iremos nos concentrar com a causa do problema de COERÊNCIA chamada CONCORRÊNCIA.

Em linhas gerais, múltiplos processos ou threads precisam acessar um determinado objeto concorrentemente e se nada for feito o estado de tal objeto é corrompido. Para evitar a causa e consequentemente sanar o problema de coerência gerado pela concorrência, iremos implementar um PROTOCOLO DE SINCRONIZAÇÃO DE OBJETOS sobre a infraestrutura criada na parte 1 do TP2. Neste TP pode ser utilizado soluções como Kafka (https://kafka.apache.org/) e outras, assim como o código do professor, disponível em - https://github.com/hpclabdecom/simplepubsub

Conforme dito, mensagens são publicadas em tópicos e recebidas por entidades denominadas subscritores, portanto para um objeto X iremos ter um tópico Tx e vários publicadores e subscritores

interessados em Tx. Cada aplicação ou tarefa interessada no objeto X precisa ter um subscritor e um publicador para Tx. Antes de alterar o estado de X, há necessidade de propagar e analisar mensagens recebidas pela solução pub/sub, por isto a criação de um novo protocolo de COERÊNCIA DE OBJETOS usando uma solução pub/sub.

Maiores detalhes de tal protocolo, assim como uma versão inicial em Java da solução distribuída Pub/Sub, é oferecida pelo professor da disciplina durante a construção do TP2. A ideia deste TP é conseguir vislumbrar as inúmeras possibilidades de implementação de serviços complexos sobre uma solução existente no mercado composta por publicadores, subscritores e tópicos. O aluno irá conhecer as alternativas clássicas de sincronizar o acesso a objetos compartilhados e será desafiado a pensar em algo inovador e ligeiramente distinto do que já foi feito na literatura.

Um dos pontos fundamentais ao garantir acesso exclusivo a um objeto X é a ordem das operações realizadas com ele durante o bloqueio. Imagine que o objeto X é do tipo conta bancária e este foi bloqueado por um processo ou uma thread. Durante o bloqueio, inúmeras operações de saque e depósitos foram realizadas, assim como operações de alterações cadastrais, tais como limites e novos dependentes da conta. Após todas estas operações, o objeto X é desbloqueado e, por questões óbvias, os demais processos ou threads interessados em X deveriam enxergar as operações na ordem em que foram realizadas. Iremos estudar os relógios vetoriais, detalhados em 1987 (paper), mas também iremos investigar como garantir a ordem das operações em X usando de maneira inteligente o sistema de mensagens adotado pela solução pub/sub escolhida para o TP2.