Dr. Tu CSCI 6401 Concurrent Programming Fall 2014

Project Assignment

Implementation of a Framework of Distributed Semaphore by Broadcast

Due: Tuesday, December 2, 2014

Algorithm

The algorithm for distributed semaphore by broadcast given in the textbook (Fig. 9.13).

Design and Implement the Framework

As described by the algorithm, each semaphore user has a helper process which is in charge of communications with other helper processes. These helpers are solely working for the semaphore and are independent to any specific application. These helpers form a framework of the distributed semaphore. A user of the framework should be able to activate a helper process (class `DisSemHelper`) on each participant computer. An essential requirement of the broadcast-based algorithms is to establish a connection to every other node from every node. Therefore, every `DisSemHelper` object needs to know the IP addresses and port numbers of all the other nodes. You are suggested to have an `Initiator` process (running on any one of the nodes) and collect the IP addresses and port numbers of all the other nodes. In doing so, the constructor of `Initiator` takes the port number that the process listens to. The constructor of each `DisSemHelper` has four parameters: the node ID number, the port number of the `DisSemHelper`, `Initiator`'s IP address, and `Initiator`'s port number.

In order to achieve flexibility, the framework should allow each application process to communicate with its helper in either of the two ways: the application program instantiates the helper as a thread, or the application program communicate with its helper process by messaging. You should minimize application programmers' involvement in messaging. One way to achieve this is to write a class `DisSem` that has two public methods `P()` and `V()`. Conceivably, its constructor takes information regarding the semaphore name (unique), the node address and the port number of its helper process.

Applications

The distributed semaphore was originally proposed for a distributed database system in which multiple database servers have dedicated network connections. You are encouraged to devise an application of the distributed semaphore framework such as a controller of a floating software license (resource sharing), online auction, and collaborative editing. The purpose of your application is to illustrate your framework.

Implementation

You may use any of the following communication mechanisms.

- Java sockets
- Websocket
- other message queue

Your implementation of the algorithm in Figiure 9.13 must be divided into two parts, (1) the distributed semaphore framework that includes the helpers (class `Helper` as process Helper); and class DisSem which extract the messaging part in process User. (2) The application programs that utilize distributed semaphore. When execute your program, each Helper object runs in a computer. The user application programs can either be deployed in the computer that runs its Helper object, or run in yet another computer. Every user application program communicates with the framework by instantiating an DisSem object and then calls its methods P_op and V_op. Method P_op first send semop[i](I, reqP, lc); …; then receive go[i](ts); … Method V_op just send semop[i](I, reqV, lc); …

Testing
Be sure to have an adequate test plan for the implementations of your framework, and to prepare four test cases for your application.

Documents
Your program should be well documented. A report on your design, implementation, testing plan and test cases is required. The report is due at the time we do project inspection.