CS 6378: Project I

Instructor: Ravi Prakash

Assigned on: September 20, 2012
Due date: October 11, 2012

You are required to implement the Ricart-Agrawala algorithm for distributed mutual exclusion with the Roucairol-Carvalho optimization. This is an individual project and you are expected to demonstrate its operation to the instructor and/or the TA.

1 Requirements

1. There are ten nodes in the system, numbered from zero to nine. Each node executes on a different machine. You can choose the machines from the fifty machines set aside for network programming (net01.utdallas.edu - net50.utdallas.edu)
2. There are reliable socket connections (TCP) between each pair of nodes. The algorithm messages are sent over these connections.
3. Each node goes through the following sequence of operations until each node has successfully exited the critical section 20 times:
   (a) Waits for a period of time that is uniformly distributed in the range [10, 20] time units before trying to enter the critical section.
   (b) On entry into the critical section, exits it after 3 time units.
4. Subsequently, the odd numbered nodes continue to issue requests for entry into the critical section at the same rate as before. The even numbered nodes, after exiting the critical section, wait for a period of time that is uniformly distributed in the range [40, 50] time units before trying to enter the critical section. As earlier, the critical section execution lasts 3 time units.
5. Once a node has successfully existed the critical section 40 times (including the 20 mentioned earlier), it does not make any more attempt to enter the critical section, and sends a *completion notification* to node zero.
6. Node zero brings the entire computation to an end once its has received *completion notification* from all the nodes and has itself finished 40 critical section executions.

2 Data Collection

For your implementation of the mutual exclusion algorithm report the following:

1. The total number of messages exchanged.
2. For each node, report the following for each of its attempts to enter the critical section:
   (a) The number of messages exchanged.
   (b) The elapsed time between making a request and being able to enter the critical section.
   (c) The maximum and minimum number of messages a node had to exchange (send requests and receive replies) to enter its critical section.

3. If you observe that the time a node waits between making a request and entering its critical section changes significantly over the duration of your experiment, please provide explanations for such an observation.

3 Point Distribution

Implementation (50%): Source code of your well structured and well documented program. You may write your code in C, C++ or Java.

Correctness (50%): Output that your program produces and the statistical analysis of the results.

4 Submission Information

Please submit your files (source code, makefile, outfile files, report containing your observations and explanations) as one "zipped" or "tarred" archive using eLearning.