Prioritized Mutual Exclusion support for RT - CORBA

Goals

Supporting prioritized mutual exclusion on a distributed system by way of implementing the extensions to known protocol for prioritized mutual exclusion for distributed systems. The extensions are meant for intent locking support coupled with read, write and upgrade lock modes.

Work Done (During Last Month)

- Changing the random wait interval resulted in introduction of some race conditions. The Debug traces are made to provide complete state of each node before and after it handles a request or makes a request.
- Set of race conditions in the implementation were identified by studying the traces of execution. The important ones that required change in the fundamental protocol are:
  - Pretending release when a pretending parent makes a new request.
  - Race between propagation of release and grant messages.
- The implementation is complete with the race conditions having been identified and handled.
- The simulation was made to generate statistical data about the number of requests being made, number of messages being sent and latency of the requests for different types of requests.
- The statistics generated by a sample run suggested some important properties about the relationships between the variables which are:
  - Number of requests being made
  - Number of participant nodes making requests
  - Randomization time intervals for critical sections and request intervals
  - Number of messages being sent by varying one of the variables above

Work Done During Spring 2002

- Installation of ACE+TAO was completed without errors and compiler errors were overcome with some fixes and adaptations.
- Documentation to install ACE+TAO on Red hat was prepared.
- The extensions of the protocol without priority support and with priority support were understood.
- With an improved understanding and identification of potential problems in the preliminary draft of the protocol, revisions were made to refine the protocol. A couple of revisions were made to separate prioritized version from the unprioritized version. This resulted in significant divergence between the two protocols as they have different set of constraints.
- The draft was finalized for unprioritized as well as prioritized versions of the protocol. Focus was shifted to unprioritized version of the protocol.

- The proposal for the NSF was made with introduction of path compression in the protocol.
- Implementation of the unprioritized version was carried out and completed with introduction of some race conditions and solutions to them.
- Implementation is made to generate complete state trace of the execution to aid in the identification of race conditions.
- The simulation for set of different variables is generating statistical data and the relationships between them are being studied.

Future Work (Up to foreseeable future)

- Running the simulation with varying conditions should discover the relationships between the variables cited above.
- Publication for the unprioritized (FIFO priorities) version of the protocol.
- Implementation of the prioritized version of the protocol should be carried out once the properties of the unprioritized version is understood.
- Publication for the prioritized (strict priorities) version of the protocol.
- The results can be compared with a similar protocol that has no multi-granularity support.
- Protocol should be embedded in some ORB implementation e.g. TAO to evaluate the performance in the working ORB environment.

Nirmit Desai
nvdesai@unity.ncsu.edu
Date: 05-10-2002