Prioritized Scalable Distributed Concurrency Services

Group members
Nirmit Desai: nvdesai@cs.ncsu.edu

Project web
http://www4.ncsu.edu/~nvdesai/middlepro.html
http://www4.ncsu.edu/~nvdesai/pubs.html
http://www4.ncsu.edu/~nvdesai/docs.html

Motivation

As researchers share computational resources, such as objects in large distributed environments, it becomes difficult to achieve scalability of synchronization. Concurrency protocols currently lack scalability. The need to enhance current middleware services to meet these requirements is imminent. Also, emerging technologies such as sensor networks operate under tight energy constraints. Communication costs being dominant consumer of energy for such systems, its important to minimize the message overhead.

Approach
The project has multiple objectives to be accomplished by following steps:

- Extend the protocol for scalable distributed concurrency services to support priority levels of requests.
- Experiment with the various priority levels to study the response time behavior for different levels of priority.
- Find a mechanism/policy to bound the response time of requests based on their priorities. Experiment with keeping the concurrency level constant.
- Make progress on the TAO front

Current status

- Protocol implementation on MPI was done. The problem of MP_THREAD_MULTIPLE was solved by not calling mpi_thread_init() (undocumented fact).
- The ratio of inter arrival request time to the critical section length was found useful to study the behavior under changing concurrency level. This is different from the current work mentioned above. Behavior is superlinear at higher lower ratio.
- Experiments on LINUX machines were carried out and comparison with Naimi's fine grain as well as coarse grain was made. Use of network latency simulator was made to have random network latency with mean value of 150 msec. Previously the coarse grain option wasn't studied. Our protocol has better message overhead behavior. Here ratio=10.
- Submission to IPDPS 2003 with the results obtained by MPI implementation on IBM SP. Experiment was repeated for various ratio values.
- Survey of various approaches to fault tolerant distributed systems and algorithms. Idea was to study the problem in the context of our protocol.

- Poster was prepared for OOPSLA student research and poster sessions. It will be presented at seattle from November 3 – 8. Print should be received by today.
- Problem with the current request latency measurements was the effect of increasing concurrency. Actually the element of interest is the distribution among nodes. The solution might be to keep the concurrency level constant.
- $$\text{Concurrency level } C = \frac{\text{Critical section length}}{\text{Inter arrival time of requests}}$$

  Experiments are being run for C = 5 and then measuring the request latencies.

Future steps

- Solution to measuring the latency at constant concurrency level should be found.
- The high level direction now is to run the simulations with different levels of priorities. Priority inversions should be avoided or bounded. The implementation for prioritized protocol is already done. The results should reflect the preference for prioritized requests.
- Follow up with Dr. Schmidt regarding implementation of concurrency services of TAO.