Dependable Distributed Middleware: Pay now or Pay Later!

Diana Szentiványi[1], Isabelle Ravot[1], Simin Nadjm-Tehrani[1], Rachid Guerraoui[2]

[1]*Dept. of Computer and Information Science, Linköping University, Sweden*
[2]*Distributed Programming Lab, Swiss Federal Institute of Technology in Lausanne'*
diasz@ida.liu.se

1. Introduction

Telecommunication systems have high level of availability as a major requirement. Currently, combinations of hardware-oriented techniques are used for achieving fault tolerance (FT). Where FT is software based, it is separately implemented in each application. In this paper we investigate achievement of server fault tolerance by providing support in the middleware. Specifically, an instance of CORBA is used for implementing generic fault handling.

We study performance/availability trade-offs when extending an existing open source ORB (OpenORB) by using a replication algorithm that employs a combination of leader election and consensus to ensure full dependability [1]. We present measurements of roundtrip time overheads and failover times performed when running a telecommunication application from Ericsson Radio Systems on top of the extended ORB.

2. Experimental platform

The algorithm of [1] works as follows: the client sends a request to the server replica with the smallest index among the ones seen to be up. The replica processes the request. If it is indeed the leader, it will try to commit the obtained result and update the state among the other witness replicas, using the consensus primitive. If it is not, the client will resend its request to some other replica. The algorithm gives two guarantees: eventually, only one server replica receives and processes requests; even if two replicas process requests, they will not commit inconsistent results. Each replica is equipped with an unreliable failure detector that gives information about which replicas are up. For the algorithm to terminate, at all times, a majority of replicas has to be up.

In the extended ORB, portable interceptors were used to execute requests on the target replica, as well as the operations required by the distributed algorithm. A leader election unit was used to send "I-am-alive" messages to similar units in the replica group.

3. Measurements

To measure roundtrip time overheads we used a client that called six of the methods of the server in a loop of 100, 200, and 400 iterations respectively. Average roundtrip times were obtained both for the case of the non-replicated service, and for the FT-ORB supported replicated group. The group size ranged from 3 to 5 replicas.

To measure failover times, a leader was forced to crash, and the client eventually sent its request to some other replica. This new replica had to set its state up-to-date according to state information previously transferred by the processing replica to its witnesses. The time taken for this procedure, during the time from the original request to the eventual reply, was measured as failover time.

4. Results

Roundtrip time overheads were in the range of 90% to 580%, varying for different server methods, and for different number of iterations. The number of replicas in the group clearly influenced the overhead. The reason was mainly the consensus protocol part, that implies sending out a message to all group members, and awaiting answers from a majority of them.

We compared our results with overhead values obtained for our implementation of a fault-tolerant CORBA platform closely following the FT-CORBA standard [2]. In that setting, extra infrastructure units such as failure detector or replication manager were essential for the failover to take place. We considered cold and warm passive, as well as active replication of only application objects.

The comparison revealed that the overheads for the fully dependable platform were higher than in a passive replication scenario, and much lower than in active replication, in the FT-CORBA platform.

References

[1] P. Dutta, S. Frølund, R. Guerraoui, and B. Pochon. *An Efficient Universal Construction for Message-Passing Systems*. In Proceedings of the 16th International Symposium on Distributed Computing, October 2002.

[2] D. Szentiványi and S. Nadjm-Tehrani. *Building and Evaluating a Fault-Tolerant CORBA Infrastructure*. In Proceedings of the DSN Workshop on Dependable Middleware-Based Systems (WDMS'02) - Washington, DC, June 2002.