

## Report on paper "Lightweight Fault Tolerance in CORBA"

Henrik Åhlander  
henrik@ahlander.com

As a part of the course Distributed Computer Systems given at IMIT/KTH we should read a paper and then write a report showing we have understood the content well. I have chosen "Lightweight Fault Tolerance in CORBA" written by Pascal Felber.

### ***Problem description and motivation for the paper***

CORBA is an architecture for distributed objects that could be used across heterogeneous platforms and programming languages. This paper shows that some distributed applications that are using CORBA could have a more easier implementation of fault tolerance than with using the methods specified in the standard specification of fault-tolerant CORBA (FT-CORBA).

Many companies were involved in the standardisation process and the result, FT-CORBA, is a very general framework. It is easy to implement on the clients but also very complicated on the server. This could result in problems for the FT-CORBA implementor and the application developer.

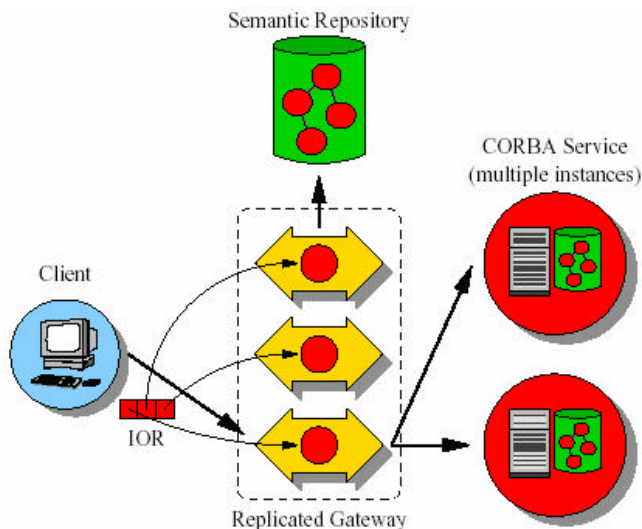
When creating a distributed fault-tolerant system reliability and availability is very important. Reliability can be provided through the use of transactions and high availability is generally achieved through replication mechanisms and load balancing. Different companies have historically achieved these objectives in many unlike ways.

Replication is used for fault-tolerance and to ensure high availability. A replicated server should appear as a single logical entity to its clients. For that reason all copies of the replicated server must contain the same data. There are two main techniques for replication. In passive replication the client is communicating with one of the replicas and that replica is then synchronizing with all the others. In active replication the client sends its requests to all replicas through the use of atomic broadcast but it only waits for the first response. All servers are however handling the request in the same order and making the same changes in state (using this technique the servers must be deterministic).

The suggested solution by this paper does not work for all applications and the author's intention is not to replace FT-CORBA with his solution. He believes however that distributed applications can benefit from the simplicity, transparency and low cost of his approach.

## Problem solution

The author proposes a middleware architecture, a special replicated gateway, that mediates the data exchange between the clients and the servers in a fault-tolerant manner, see figure 1. The replicated gateways communicate with each other and also take care of load balancing and increase the system's reliability and availability. A gateway is fully transparent to the clients and servers and re-engineering of those is not needed. To the client, a gateway appears to be the server, but the request is passed to one or more of the servers.



**Figure 1.** This illustration, taken from the paper, shows how the client, the gateways and the servers interact with each other.

The gateway should use semantic knowledge of server objects to make the communication more efficient. Semantic analyze was earlier most used for databases but is now also considered in the context for distributed systems. The semantic knowledge tells the gateway if a request would change the object's state or if it is for example read only or deterministic. It could also describe the relationships between multiple requests if they for example depend on each other. By knowing this, the gateway could select the least expensive protocol that still guarantees the replicas to remain consistent. If the request for example is read only, the request is only sent to one of the servers. A write request must be sent to all servers with atomic broadcast if the consistency shouldn't be broken.

The client-side of FT-CORBA is kept minimal just specifying object references that can contain multiple profiles using different replicas (multi-profile IORs) and simple rules for iterating through the profiles in case of failure. The solution proposed by the author uses the same client-side as FT-CORBA.

The servers are not aware of each other and there is no communication between them. Due to this reason all replication must be controlled by the gateways. If the servers are not deterministic it is not possible to have

them replicated, at least not if the states of the servers couldn't be read and set. If the servers instead are deterministic update request could be sent to every server. In this case it is important that the gateways communicate with each other to make sure that update requests and read requests are performed in the same order everywhere.

When using CORBA object references could be returned to the clients. These references could lead the clients to the servers directly without passing the gateway. If this happens this server could be updated while the others are not. The consistence will then be broken. To avoid this the gateway scans the responses for object references and if any is found it changes it to a multi-profile IOR for proxies on the gateways leading to the replicated objects. This scan is costly and the paper therefore suggests the semantic show know which requests that could response with references and then only scan those.

### ***Strengths***

A strength of this solution is that it is transparent to both clients and servers and no re-engineering is needed on the servers to get fault tolerance. The solution provides high availability, has load balancing and uses semantic to work smarter.

### ***Weaknesses***

The solution works only for some types of applications if the system should be able to maintain consistency. Problems could also arise if communication links between the gateway and the server fail and the gateway cannot know if the request was performed or not because of the lack of state management.

### ***The students suggestion on improvement***

I still don't have any suggestions.