Fault-Tolerant CORBA, A Closer Look

Hadi Salimi
M.S. Seminar
School of Computer Engineering,
Iran University of Science and Technology,
hsalimi@iust.ac.ir
Advisor: Dr. Mohsen Sharifi

Agenda
- CORBA Overview
- FT-CORBA
- Strategies
- Other Paradigms
- Challenges
History

- Object-Oriented Programming 1960s
- Distributed Computing 1980s
- Distributed Object Middleware
- CORBA 1990s

CORBA

- It Stands for Common Object Request Broker Architecture.
- OMG, (A consortium over 800 companies) has produced the CORBA standard; scores of vendors conform.
- The CORBA allows objects to invoke services from other objects, hiding differences in location, programming language or Platform.
CORBA Architecture

FT-CORBA

Object-Oriented Programming 1960s

Distributed Computing 1980s

Distributed Object Middleware

Fault-Tolerant Computing

CORBA 1990s

FT-CORBA
FT-CORBA

- At 1998 OMG issued the RFP
- In early 2000 the first version released
- The last version, December 2001
- using
  - Replication
  - Object Groups

Replication

- The various approaches to fault-tolerant CORBA are alike on their use of replication.
- The behind idea is to mask the failure of an object from a client.
Replication

Active and Passive Replication

Object Replication

Active
Passive
Warm
Cold

Active
Rep 1
Rep 2
Rep 3

Passive
Primary
Backup 1
Backup 2

FT-CORBA
9

Object Groups

- Object group represents a replicated object and the group members represent the individual replicas of the object.
- Each object group has an Interoperable Object Group Reference (IOGR).
**FT-CORBA Architecture**

- **Replication Manager**
  - Property Manager
  - Object Group Manager

- **Generic Factory**

- **Fault Detector**
- **Fault Notifier**

- **Host H**
  - Client Object
  - Factory
  - Server Replica
  - Fault Detector
  - Logging

- **Host H1**
  - Server Replica
  - Fault Detector
  - Logging

- **Host H2**
  - Server Replica
  - Fault Detector
  - Logging

- **Is_alive()**

**Other Systems**

- **Delta-4**
  - 1990 – supported by CEC through ESPRIT Project
- **Arjuna**
  - 1994 – Newcastle University
- **Orbix-Isis**
- **Electra**
  - 1995 – Zurich University
- **DOORS**
  - 1997 - Bell Labs Research
- **IRL**
  - 1999 – Rome University
- **Eternal**
Strategies

- The Integration Approach
  - Where support for replication is integrated into the ORB.
- The Interception Approach
  - Where support for replication is provided transparently underneath the ORB.
- The Service Approach
  - Where support for replication is provided through an object service.

Integration Approach

- Replace CORBA’s TCP/IP based protocol IIOP with Group Communication Protocol.
- So this is non-compliant with standard ORB.
- There is no need to change the clients and the servers on the ORB.
  - Electra
  - Orbix-Isis.
Service Approach

- The ORB is not going to be changed.
- But the application must be aware of the service so the application code will change.
- Every interaction is managed by ORB, So this will be performance poor.
  - DOORS
  - IRL
  - Friends

Interception Approach

- If the intercceptions are specific to the operating system, then they need to be ported to the target OS.
  - Eternal
## Comparison

<table>
<thead>
<tr>
<th>Approach</th>
<th>Advantages</th>
<th>Disadvantages</th>
<th>Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integration</td>
<td>Transparent to the Application</td>
<td>Needs porting modified ORB for every object</td>
<td>Electra, Orbix+Isis</td>
</tr>
<tr>
<td>Service</td>
<td>CORBA-Compliant can exploit CORBA's Interoperability to work with any ORB</td>
<td>Not always Transparent to Application</td>
<td>OGS, FTS, FRIENDS, NewTop</td>
</tr>
<tr>
<td>Interception</td>
<td>Transparent to Application and ORB</td>
<td>Needs to be ported to every operating system</td>
<td>Eternal</td>
</tr>
<tr>
<td>FT-CORBA Standard</td>
<td>Standardized, configurable support</td>
<td>Leaves Implementation details open, Requires extensions to the standard ORB Core</td>
<td>Eternal, DOORS</td>
</tr>
</tbody>
</table>

**FT-CORBA Standard**

- Based on Eternal, Doors.
- Parameters:
  - Replication Style: Active, Warm or Cold
  - Application-Controlled or Infrastructure-Controlled
    - Membership Style: Adding and Removal of replicas
    - Consistency Style: Checkpointing, Logging
  - Initial Number of Replicas
  - Minimum Number of Replicas
  - Checkpoint Interval
Server and Client Transparency

- With the FT-CORBA Standard, group specific information can be embedded into object references.
- Client: needs to be recompiled/relinked with new libraries.
- Server: Checkpointable Interfaces for state retrieval must be realized.
Integration with non-replicated Objects

- The communication of a replicated client with non-replicated servers
- server with non-replicated clients

Replica Deployment

- Replicas of a CORBA object can not be supported across different FT-CORBA implementations.
Open Issues

- Supporting the CCM
- FT/RT CORBA
- FT and Security
- Replication and Transactions

References


References (Cont.)


Fault-Tolerant CORBA, A Closer Look

Hadi Salimi
M.S. Seminar
School of Computer Engineering,
Iran University of Science and Technology,
hsalimi@iust.ac.ir
Advisor: Dr. Mohsen Sharifi