Load Management for Distributed Object-Oriented Environments

Markus Lindermeier,
Department of Computer Science,
University of Technology Munich,
Germany

Email: linderme@in.tum.de
**Motivation**

- Distributed systems like CORBA and DCOM cause new problems
  - Applications are distributed over multiple hosts
  - Heterogeneous hosts with different computation power
- Mapping of application objects to hosts
- Load imbalance because of background load
- Overload of server objects because of too many clients

Load Management deals with all these problems and tries to improve the performance by distributing workload
Classification of Load Management Systems

- Implementation and integration
  - Application level: Implemented by the programmer
  - System level: Functionality provided by the middleware
  - Service level: Hybrid approach

- Load distribution entities
  - Data: Used by application level load management
  - Server processes: Existing tools for process migration
  - Objects and requests: Fine-grained

- Strategy used for load distribution
A load management system consists of three components.

- This eases design and implementation of the overall system.
General Design Principles

• System level implementation
  – Ease the development of distributed applications
  – Avoid conflicts with contradictory strategies
  – Enable an efficient implementation of load distribution

• Objects as load distribution entities
  – Processes (Server) are too coarse-grained
  – Objects are the natural granule for object-oriented systems
  – Object replication enables distribution of requests
Load Distribution Mechanisms

- Initial placement
  - Performs the mapping of objects to hosts at creation time
  - Find a host that has enough computation power
  - Applicable to all objects

- Migration
  - Move an object to another host
  - Stop request processing
  - Transfer the object state to the new object
  - Enables the compensation of load imbalance (background load)
  - Dynamically improves the mapping of objects to hosts
  - Applicable to all objects
Load Distribution Mechanisms

- Replication
  - Similar to migration but the original object is not removed (replicas)
  - Requests are divided up among the replicas
    - Dynamic request assignment
    - Static request assignment
  - Enables to deal with object (server) overload
  - Applicable to "replication safe objects" only
A Load Management Architecture for CORBA

- Implementation Repository (IMR): Decision/evaluation component
- System load is gathered via SNMP
- Object’s resource usage is gathered via the ORB/POA
- Load distribution functionality is integrated into the ORB/POA
Object Creation and Activation

- In CORBA objects are created by the programmer
- The load management system has to create objects and replicas on demand (migration and replication)
  - New POA policy (*ControlFlowPolicy*)
  - Servants (Objects) are created by a *ServantFactory*
  - The object is automatically activated by the POA
  - Persistence is achieved by a *PersistentServantFactory*
- The users may create objects by an IMR interface (*GenericFactory*)
  - Necessary when the new *ControlFlowPolicy* is used
  - Starts a server process if necessary and creates the object
Request Assignment

- Request assignment is done by the **LocationForward** mechanism
  - A client’s first request is directed at the IMR
  - Further requests are forwarded to the object (replica)
  - For invalid object reference the client falls back to the IMR
- **LocationForward** is sufficient for dynamic request assignment
- Static request assignment requires an extension
  - For invalid object references the client falls back to the last forwarded reference instead of the IMR
  - Drawback: This extension affects the client side
Test Case
Conclusion and Future Work

- The characteristics of distributed object-oriented environments require new load management concepts
- New load distribution mechanisms like replication
- Our approach applicable to various middleware environments
- The CORBA implementation proves the feasibility

- Currently, the load management system is tested with real-world applications