Modular Quality of Service-enabled Load Management Service for Component-based Distributed Systems

The Problem

Objective: Modular load management service offering runtime configuration and performance optimisation as well as QoS support for existing distributed applications.

Monitoring:
- Provides different monitoring granularity levels.
- Runtime selection of monitors for optimal application workload management.
- Predicts near-future workload prediction.
- done for all service levels.
- Validates the workload model.

Modeling:
- Creates a workload model that is continuously updated and validated.

The Context

Distributed Application Cluster

QoS-enabled Distributed Application Cluster

Load Distribution Algorithms:
- Runtime change tuning
- Platform-independent representation
- Scalability
- Runtime estimatability of algorithms set

Target Use-Case

Server Farm

Distributed Application 1

Distributed Application 2

Server Farms host different distributed applications, with different usage patterns.

Resource sharing among them is important for optimally handling the workload.

The Framework offers the possibility of:
- Interconnecting with similar services
- Sharing available resources between framework instances
- Dynamically adding/removing servers to the group

These capabilities ensure optimal response times.

Fallout protection

Distributed Application Cluster

1. Incoming request is dispatched to all servers — there is no single entry point in the system.
2. Servers analyse global system load and decide if they accept the transaction — there is no single decision point.
3. The server accepting the transaction announces the others — the decision is validated by the other servers ensuring that the selected server has not failed.

The possibility of disabling this mechanism and using one entry point is included (minimise overhead).

Summary and current results:

Framework modularity:
- Runtime module activation/deactivation/replacement
- Selection of active modules based on application complexity

At runtime:
- Select optimal load distribution algorithm (or adjust its parameters)
- Change load metrics used
- Dynamically change the number of servers in the cluster

Integrate an end-to-end QoS service into the middleware platform (application server):
- Offer QoS at application level (for QoS unaware applications)
- Detect mapping of existing QoS research on global policies and local enforcement techniques

Advantages of hosting multiple applications in a farm of servers:
- Resource sharing
- Unified monitoring leads to lower overhead (monitors can be shared between distributed applications)
- Central management point for the server farm (all distributed applications)