Re-architecting J2EE Application Server with Isolates

Marcin Jarzab
Intern at Sun Labs (Barcelona project)
University of Science and Technology (Krakow, Poland)
Introduction

• Isolate is an independent computational domain introduced by the JSR 121
  – *API for creation and lifecycle management*
  – *Fully compatible with existing applications*
  – *Different implementation strategies*

• MVM (Multitasking Virtual Machine) is a modified Java Virtual Machine (JVM) architecture enabling *isolated* execution of multiple Java applications in a single address space
MVM architecture

- **Mserver**: first isolate application that listens on a socket for connections from *Jlogin* process and creates new isolate for application
- **Jlogin**: connects to *Mserver* to run isolate for application
- **Isolates**: independent computational environments run in MVM
- **Links**: for efficient communication between *Isolates*
Motivation, Goals

• Integrate Isolate architecture into the J2EE platform
  – Scalability
  – Clean termination of applications
  – Resource Management

• Study how new APIs (Isolates, Resource Management) can be introduced into the application server architecture
  – Programming model
  – Performance
  – New features
J2EE Application Model
J2EE Reference Implementation
Possible approaches to introducing Isolates into J2EE

- Multiple J2EE servers in MVM
- One J2EE server with each component (Servlet and EJB) in a separate isolate
- One J2EE server with each application in a separate isolate
  - Application domain
Application domain

- Separate configuration environment disjoint from other domains
- Contains J2EE application resources (JDBC, JavaMail, Connectors)
- Unit of management for resource consumption
J2EE environment with Isolates

MVM

Administration server
- Administration services
- Tomcat
- CORBA (ORB, Naming server)

JMS server
- CORBA (ORB, Transaction service)
- Security (Realms database)
- JMS Resources (Queues, Topics)

Application domain
- Tomcat
- CORBA (ORB, Transaction service)
- Security (Realms database)
- App. Resources (JDBC, JavaMail)

Database
- JDBC – Links or TCP/IP

Links

RMI-IIOP

JDBC – Links or TCP/IP
Conclusions

• Deeper understanding of all software layers
  – J2EE specification
  – Request flow in EJB and Web servers, deployment logic
• Built J2EE application server prototype based on MVM and Isolates proved its usability in J2EE environments
• Good platform for further experiments