Performance Engineering
Middleware Systems

Dr. John Murphy
Dr. Liam Murphy
Agenda

• Overview
  – What is Performance Engineering?
  – What is PEL?
  – Component-Oriented Middleware Systems
• EJB-Express™
• Market Trends
• What the future holds?
What is Performance Engineering?

**Performance Engineering** \( n. \) the study of systems using mathematical methods to determine how they can operate in a more efficient manner by identifying and controlling congestion of scarce resources.

**Performance Engineering Laboratory (PEL)**

- Use theoretical analysis in systems where performance issues arise
- Develop tools and models to aid system design, performance analysis, testing and evaluation
What is the Performance Engineering Lab?

• Joint research group within Dublin City University and University College Dublin
  – DCU: Electronic Engineering; UCD: Computer Science
• 3 Academic staff
• 4 Support staff
• 24 Postgraduate researchers
• Performance related research
  • Multimedia Networking & Mobile Multimedia
  • Internet Enabled Computing Systems
Major PEL Projects (as of Oct. 2003)

“A Multimedia Streaming Application for Evolving Heterogeneous Mobile Networks” [2002 EI – ATRP (UCD/DCU)]

“Performance Optimisation of Components” [2002 EI – ATRP (DCU/UCD)]

“Network Independent Voice over IP for Wireless Local Area Networks” [2002 EI – RIF (DCU/UCD)]

“Automated Performance Analysis of EJB Application Servers” [2001 EI – ATRP (DCU/UCD)]

“Adaptive Quality of Service for MPEG4 Streaming” [2001 EI - RIF (UCD)]

“Mobile System Performance Evaluation Tool” [2000 EI - AR / Ericsson (DCU)]

“Audio/Video Feedback in Multimedia Networks” [1998 EI – Strategic (DCU)]
What are middleware systems?
Components

- A component is a lego™ like piece of code
- Allow developers to concentrate on business logic
- Decouple the logic from architecture
- Distributed Components offer:
  - Security
  - Robustness
    - Persistence
    - Transactional capability
  - Scalability
Components

- Reservation
- Payment
- Payment

ORB
Component Technologies

- CORBA (Common Object Request Broker Architecture)
  - Open standard created by OMG

- .NET framework
  - Developed by Microsoft

- EJB (Enterprise JavaBeans)
  - Developed by SUN
  - J2EE technology de facto standard
  - 65-75% enterprises (Gig 2002)
J2EE (Java 2 Enterprise Edition)
EJB (Enterprise JavaBeans)

SUN Microsystems’ definition of Enterprise JavaBeans is:

The Enterprise JavaBeans architecture is a component architecture for development & deployment of component-based distributed business applications.

Applications written using the EJB architecture are scalable, transactional, and multi-user secure.

The application may be written once, and deployed on any server platform that supports the Enterprise JavaBeans specification.
Distributed Objects

1. Invoke Method
2. Communicate Method Invocation

3. Invoke Method
4. Communicate Result
5. Return Result
Know Your Beans!

• Entity Beans
  – Hold persistent data structures used by application
  – Bean represents a logical record in DB

• Session Beans
  – Workflow or task oriented
  – Can be stateless or stateful (*conversational state*)

• Message Beans
  – Asynchronous messaging
  – Doesn’t have remote/local/home interfaces
  – Loosens coupling between sender and receiver
Design Decisions

What’s the best design?
J2EE design decisions

• Work performed by Rice University
• Auction site implemented in 6 different ways:
  – Servlets
  – Session Beans
    • Business logic implemented in a Session Bean
  – Entity Beans CMP (Container Managed Persistence)
    • Data Access implemented in an Entity Bean
  – Entity Beans BMP (Bean Managed Persistence)
  – Session Facade
    • Communication through a Stateless Session Bean
  – EJB 2.0 Local
## Auction site: Complexity

<table>
<thead>
<tr>
<th></th>
<th>Servlets</th>
<th>Beans</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Classes</td>
<td>LoC</td>
<td></td>
</tr>
<tr>
<td>Servlets-only</td>
<td>23</td>
<td>4540</td>
<td>25</td>
</tr>
<tr>
<td>Session Beans</td>
<td>22</td>
<td>2730</td>
<td>51</td>
</tr>
<tr>
<td>EB CMP</td>
<td>23</td>
<td>3980</td>
<td>40</td>
</tr>
<tr>
<td>EB BMP</td>
<td>23</td>
<td>3980</td>
<td>40</td>
</tr>
<tr>
<td>Session Facade</td>
<td>22</td>
<td>2660</td>
<td>85</td>
</tr>
<tr>
<td>EJB 2.0 Local</td>
<td>22</td>
<td>2725</td>
<td>91</td>
</tr>
</tbody>
</table>
Auction site: Performance Results

![Bar chart showing throughput requests per minute for different technologies: Servlets, Session Beans, EB CMP, EB BMP, Session Facade, EJB 2.0 Local. Servlets have the highest throughput.]
EJB Design Decisions

• Communication Overhead
  – When to use local vs. remote interface calls
• Entity Beans
  – Does the data model map to entity beans?
• Stateless vs. Stateful beans
• CMP vs. BMP
Other Performance Problems: What leading experts say

- “The performance challenges typically revolve around excessive or inefficient accesses to backend systems such as databases and mainframe systems.”
- “Poor design does not allow optimization”
- “Excessive method calls”
- “Not designing in an architectural context”
- “No consideration for concurrent users”
- “Tuning the number of JVMs serving the clients and tuning the number of threads per JVM”
Summary

• Component Technology
  – Allows developers to concentrate on business logic
  – Bring technology-specific performance problems
  – Removing an architectural context can create performance problems

• PEL is working to solve these issues
EJB-Express™

Taking EJB Performance to the Max
Performance: Becoming a Major Concern

- 67% of IT Managers expect an increase in budgets for performance management tools over the next 24 months (Newport Group research)
- Most enterprises will expend 25%+ more effort/time than necessary in troubleshooting application and network problems due to failure to use effective monitoring and testing tools*
- 20% of enterprise mission-critical applications will experience severe performance problems that could have been avoided by modelling network/application interactions*


Potential to save enterprises significant time and money through enhanced performance testing tools
Traditional Monitoring/Testing Tools:
- Need Production like environments
  - Only available at end of project
  - Costly to construct & maintain
- Find Problems one at a time
  - Multiple Fix-Test-Fix cycles
- Testing Takes Time and Money

Experience Shows:
- Time is Critical
- Mistakes are costly
EJB-Express™ Product Line

- **EJB-Inspector™**
  - What is your system doing?
- **EJB-Performance Inspector™**
  - What resources are being used?
- **EJB-Performance Predictor™**
  - Where are the performance bottlenecks?
- **EJB-Performance Modeller™**
  - What if...?
EJB-Inspector™

• What is it?
  – Generates UML Event Sequence diagrams

• Why use it?
  – Helps verify/debug the system

• Value?
  – Saves 2-3 days of looking at logs
Does this make sense?!
EJB-Performance Inspector™

• What is it?
  – Collects method-level performance data

• Why use it?
  – Find performance-hungry code
  – Provide planning data

• Value?
  – Can’t debug code without it!
Data Collection
EJB-Performance Predictor™

- **What is it?**
  - Finds further bottlenecks in the system
- **Why use it?**
  - Faster time to market
- **Value?**
  - Reduces multiple fix-test-fix cycles
Prediction

• Looks beyond the first performance problem
  – InSight™ Prediction

• Predicts performance for different hardware
  – ForeSight™ Prediction

• Expands the performance test envelope
  – ClearSight™ Prediction
Synchronous Messaging
Too Many Method Calls
Single Threaded Code
Inefficient Code
DB Indexes

InSight™ Prediction
Helps find all the Performance Problems at once
ForeSight™ Prediction

Find Performance Problems on less/different hardware

Content Providers

Firewall

Web Servers

Application Servers

DB Servers

DB Servers

Legacy System

LAN
Lack of time means few performance tests are run, leaving performance problems undetected.
EJB-Performance Modeller™

• What is it?
  – Expands EJB-Performance Predictor™

• Why use it?
  – Provides visibility across the enterprise
  – What if…?

• Value?
  – Dependent on system complexity
Market

• Application Server Software
  – Component based development will continue to grow in importance based on both J2EE and .Net
  – Market for performance testing tools for J2EE environments: approx €400 million (IDC figures)
• Key vertical markets: Banking, Insurance, Government, Internet-based businesses
• Key Players in this market:
  – Broad Performance Testing Tool Market: Mercury Interactive
  – Partially Comparable: Borland, Quest (Sitraka), Veritas (Precise), Hyperformix

EJB-Express™ Differentiator
Identifies root cause(s) of performance problems
How EJB-Express fits in…

• Load Test Tools
  – e.g. Mercury, Segue, Empirix, CA
  – Only say you have a problem

• Monitoring Tools
  – e.g. Topaz, PerformaSure
  – Only tell you which box has the problem

• Profiling Tools
  – e.g. JProbe, OptimizeIT
  – Only tell you the resource-hungry code

• **EJB-Express™**
  – Tells you what you need to do to reach performance goals!
The Future?

- Refine business plan
- Prepare launch strategy
- Reference site(s) for higher-value products
- Relationships with partners & distributors
The Future?

- Refine business plan
- Prepare launch strategy
- Reference site(s) for higher-value products
- Relationships with partners & distributors

But we have the logo…

Taking EJB Performance to the Max