J2EE PERFORMANCE:
From Benchmarking to Sizing and Capacity Planning

Samuel Kounev (skounev@acm.org)
TU-Darmstadt, Databases & Distributed Systems Group
SPEC-OSG Java Subcommittee

Agenda

• The J2EE Platform for E-Business Systems
• Benchmarking J2EE: ECperf, SPECjAppServer
• Sizing and Capacity Planning
• Approaches to Performance Modelling
• Future Work and Research Interests
• Conclusions
The J2EE Platform for E-Business Systems

- Technology of Choice for Today’s E-Business Systems
  - 80% of all enterprises use the Java language (Gartner)
  - 92% of companies that choose J2EE for enterprise computing are happy with their choice (Forrester)
  - 78% of executives view J2EE as the most effective platform for building and deploying Web Services (Giga poll)
  - The J2EE Platform market is $2 billion strong and growing
The J2EE Platform for E-Business (2)

- Industry standard defined by Sun Microsystems, Inc.
- Over 35 implementations on the market – App. Servers
- The latter provide a range of middleware services crucial for today’s e-commerce systems
- **Once functionality is standardized, performance becomes the distinguishing factor!**
- Needed are industry standard benchmarks to measure performance and scalability of App. Servers.
- Testing should be monitored and controlled to avoid speculations and misuse of results

Next

- The J2EE Platform for E-Business Systems
- **Benchmarking J2EE: ECperf, SPECjAppServer**
- Sizing and Capacity Planning
- Approaches to Performance Modelling
- Future Work and Research Interests
- Conclusions
The ECperf Benchmark

- It all started with **ECperf 1.0** released in June 2001

  ![ECperf](image)

- Developed at Sun in conjunction with App. Server vendors under the JCP
- Hosted on [http://ecperf.theserverside.com](http://ecperf.theserverside.com)
- Version 1.1 was released in April 2002
- ECperf is taken over by the SPEC-OSG Java Subcommittee
- The next version will be called **SPECjAppServer2001**

About SPEC

**Standard Performance Evaluation Corporation**

- Open Systems Group (OSG)
- High Performance Group (HPG)
- Graphics Performance Characterization Group (GPC)

**MISSION:** To establish, maintain, and endorse a standardized set of relevant benchmarks and metrics for performance evaluation of modern computer systems.

![Better Benchmarks](image)
Who We Are?

- Measures the performance of J2EE 1.2 App. Servers
- Based on EJB 1.1
- Derived from ECperf™ 1.1 developed under the JCP
- Uses a different metric and slightly modified workload
- Introduces categories of results
- Under the control of the OSG-Java Subcommittee

The SPECjAppServer2001 Benchmark

- Measures the performance of J2EE 1.2 App. Servers
- Based on EJB 1.1
- Derived from ECperf™ 1.1 developed under the JCP
- Uses a different metric and slightly modified workload
- Introduces categories of results
- Under the control of the OSG-Java Subcommittee
Three Business Domains modelled:

- **Order / Inventory Management**
  - B2C Customer Interactions, Online-Ordering
- **Just-in-Time Manufacturing**
  - Production / Assembly Lines
- **Supply-Chain Management**
  - B2B Interactions with External Suppliers
The SPECjAppServer Business Model

**CUSTOMER DOMAIN**
Order Entry Application
TXs: - Place Order
     - Change Order
     - Get Order Status
     - Cancel Order
     - Get Customer Status

**MANUFACTURING DOMAIN**
Parts → Planned Lines
Large Order Line → Widgets
Transactions: - Schedule Work Order
               - Update Work Order
               - Complete Work Order
               - Create Large Order

**CORPORATE DOMAIN**
Customer, Supplier, and Parts Info
TXs: - Check Credit
     - Get Percent Discount
     - New Customer

**SUPPLIER DOMAIN**
TXs:
- Send Purchase Order
- Deliver Purchase Order

**Figures:**
Figure 31: Scaling the Distributed Worldwide Business
**The SPECjAppServer Application Design**

- **Benchmark Components:**
  1. **EJBs** – J2EE appl. deployed on the *System Under Test (SUT)*
  2. **Supplier Emulator** – servlet simulating interactions with suppliers
  3. **Driver** – Java appl. running on a client machine

- **RDBMS for Persistence** - both CMP and BMP supported

- **Benchmark’s Throughput function of chosen Transaction Injection Rate** - \( Ir \)

- **Performance Metric provided is** BOPS/sec = total number of business TXs completed in the Customer Domain + total number of workorders completed in the Manufacturing Domain, normalized per second.

---

**The Centralized Workload**

---
The Distributed Workload

Current Benchmark Results

<table>
<thead>
<tr>
<th>Sponsor</th>
<th>J2EE system</th>
<th>Gbps/\mips/Sq</th>
<th>Price/Bps</th>
<th>J2EE Server</th>
<th>DBMS</th>
<th>Date Approved</th>
<th>PDF</th>
<th>ZIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM</td>
<td>eServer x330 cluster</td>
<td>32581.47</td>
<td>$11</td>
<td>WebSphere AB 4.0.3</td>
<td>DB2 7.2</td>
<td>4/23/02</td>
<td>PDF</td>
<td>ZIP</td>
</tr>
<tr>
<td>IBM</td>
<td>eServer x330 cluster</td>
<td>25659.13</td>
<td>$12</td>
<td>WebSphere AB 4.0.3</td>
<td>DB2 7.2</td>
<td>4/29/02</td>
<td>PDF</td>
<td>ZIP</td>
</tr>
<tr>
<td>HP</td>
<td>UL6000c U3 cluster</td>
<td>16696.17</td>
<td>$18</td>
<td>BEA WebLogic Server 7.0 Beta</td>
<td>Oracle 8i.7 (Weblogic JDriver)</td>
<td>2/25/02</td>
<td>PDF</td>
<td>ZIP</td>
</tr>
<tr>
<td>IBM</td>
<td>xSeries x350, Model 6K Y cluster</td>
<td>16634.40</td>
<td>$13</td>
<td>WebSphere AB 4.0.5</td>
<td>DB2 7.2</td>
<td>3/11/02</td>
<td>PDF</td>
<td>ZIP</td>
</tr>
<tr>
<td>HP</td>
<td>UL6000c U3 cluster</td>
<td>15180.43</td>
<td>$20</td>
<td>BEA WebLogic Server 7.0 Beta</td>
<td>Oracle 8i.7 (Oracle Driver)</td>
<td>2/19/02</td>
<td>PDF</td>
<td>ZIP</td>
</tr>
<tr>
<td>Framat</td>
<td>DL-580 cluster</td>
<td>14467.97</td>
<td>$22</td>
<td>Framat Application Server 3.0 SP1</td>
<td>Oracle 8i.6</td>
<td>4/29/02</td>
<td>PDF</td>
<td>ZIP</td>
</tr>
<tr>
<td>IBM</td>
<td>p64x0 model B380 cluster</td>
<td>10316.13</td>
<td>$27</td>
<td>WebSphere AB 4.0.5</td>
<td>DB2 7.2</td>
<td>1/7/02</td>
<td>PDF</td>
<td>ZIP</td>
</tr>
</tbody>
</table>

- Measures performance of **J2EE 1.3** App. Servers
- Based on **EJB 2.0**
- Will add a Web Tier (JSPs and Servlets)
- All applications will be accessed through the web frontend
- Will use messaging for intra-domain communications
- Will minimize the load on the database layer
From Benchmarking to Capacity Planning

- **J2EE Benchmarking** allows us to measure and compare the performance and scalability of J2EE-based environments.

- However, it does not address the problem of **predicting performance** for the purposes of system sizing and capacity planning.

- The issues of **sizing and capacity planning** are gaining in importance as the complexity and size of modern e-business applications increase.
Sizing and Capacity Planning Issues

System developers are often faced with questions such as:

- What are the max. load levels that the system can handle?
- What would the average response time, throughput and CPU utilization be for a particular workload?
- How much resources (servers, CPUs, memory) would be required to meet the Service Level Agreements (SLAs)?
- Which components of the system affect performance the most? Are they potential bottlenecks?

This is what sizing and capacity planning is all about. The main problem is how to predict the performance of a particular system under a particular workload.

If \( n = 1000 \), \( k=? \), \( m=? \), \( p=? \) so that all SLA are fulfilled.
Approaches to Performance Prediction

- Make an Educated Guess
- Generate Load and Measure Performance
- Build a Performance Model of the System

Approach 1: Make an Educated Guess

Rely on intuition, expert opinions, past experience, ad hoc procedures and general rules of thumbs.

**PROS:** Quick, easy and cheap.

**CONS:** Very inaccurate and risky.
Approach 2: Load Testing (brute force)

Use load-testing tools that generate artificial workloads and measure performance.

**PROS:** Provides accurate and realistic data. Could help to identify bottlenecks and fine-tune system prior to production.

**CONS:** Extremely expensive and time-consuming. Assumes that the system is available for testing.

Approach 3: Performance Modelling

Build and analyse performance models which capture the performance and scalability characteristics of the system.

**PROS:** Often much cheaper and quicker than load-testing. Could be applied at the design stage.

**CONS:** Extremely complex. Accuracy depends on how representative models are.
The J2EE Platform for E-Business Systems

Benchmarking J2EE: ECperf, SPECjAppServer

Sizing and Capacity Planning

Approaches to Performance Modelling

Future Work and Research Interests

Conclusions
Approaches to Performance Modelling

Performance Models

Simulation Models

Analytic Models

Hierarchical Models

CTMC Models

Petri Net Models

DTMC Models

Queueing Networks

Semi-Markov Models

Queueing Networks

Product-form

Extended

Non-Product-form
Petri Net Models

- PT Nets
- SPN
- GSPN
- Colored (GS)PN
- QPN Nets

Problems with Available Models

- Simulation Models
  - Accurate, but very expensive to develop and run

- Analytic Models
  - Much cheaper, but less accurate.
    - Continuous Time Markov Chains (CTMC)
      - suffer the state space explosion problem
    - Queueing Networks
      - fail to represent synchronisation aspects
    - Petri Nets
      - fail to model scheduling strategies

- None of the models above are suitable for modelling event-based systems.
Agenda

• The J2EE Platform for E-Business Systems
• Benchmarking J2EE: ECperf, SPECjAppServer
• Sizing and Capacity Planning
• Approaches to Performance Modelling
• Future Work and Research Interests
• Conclusions

Future Work and Research Interests

• Use the models available to study the performance of realistic applications
• Benchmarking Event-Based Systems
• Sizing and Capacity Planning for MOM Systems
Conclusions

• Benchmarking J2EE Performance is not easy, but it’s fun!

• SPEC has taken the problem seriously and is soon going to provide the right benchmarks and tools.

• Performance Prediction for Sizing and Capacity Planning is an extremely difficult and challenging area.

• Plenty of models have been developed in the last century, but most of them are:
  - highly-specialized
  - not utilized to their full potential
  - limited in their modeling power and representativeness
Related Work

- “Improving Data Access of J2EE Applications by Exploiting Asynchronous Messaging and Caching Services”
  Proceedings of the 28th International Conference on Very Large Data Bases (VLDB).

- „Performance Issues in E-Business Systems“

- „Eliminating ECperf Persistence Bottlenecks when using RDBMS with Pessimistic Concurrency Control“
  © S.Kounev, 2001
  Submitted to the ECperf Expert Group at Sun Microsystems Inc., September 18, 2001

- „A Capacity Planning Methodology for Distributed E-Commerce Applications“

- „Performance Prediction, Sizing and Capacity Planning for Distributed E-Commerce Applications“

That was it...

Thank You for your Attention!

For more information visit:

http://www.spec.org
http://www.dvs1.informatik.tu-darmstadt.de/