Performance and Scalability of EJB Applications

Emmanuel Cecchet
Julie Marguerite
Willy Zwaenepoel

Presented by Chen Fu
Unfolding J2EE

- Servlets
- EJB
  - Session Beans
    - Stateless
    - Stateful
  - Entity Beans
    - Bean Managed Persistence
      ==> Program managed persistence
    - Container Managed Persistence
      ==> Framework managed persistence
Servlets

http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&category=2990&item=4366543439
Servlets

http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&category=2990&item=436654343

The request gets to TomCat
Servlets

http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&category=2990&item=436654343

Used by TomCat to find the right Servlet Class
Servlets

http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&category=2990&item=4366543434

Parameters Passed to the Servlet Class
http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&category=2990&item=4366543439

```java
public class SomeServlet extends HttpServlet{
    public void init(ServletConfig config) throws ServletException{
        ...
    }

    public void destroy() {
        ...
    }

    protected void doGet(HttpServletRequest req,
                         HttpServletResponse resp)
        throws ServletException,
                java.io.IOException{
            ...
        }
    }
```
public class SomeServlet extends HttpServlet{
    public void init(ServletConfig config) throws ServletException{
        ...
    }

    public void destroy() {
        ...
    }

    protected void doGet(HttpServletRequest req,
            HttpServletResponse resp)
            throws ServletException,
            java.io.IOException{
        ...
    }
}

public void doDelete(HttpServletRequest req,
            HttpServletResponse resp)
            throws ServletException,
            java.io.IOException{
        ...
    }
}
Servlets

- Web container
- Servlet
  - Presentation logic
  - Business logic

Database
EJB

- Implementation of RMI server with some life cycle management
  - A bean class can post some interface for remote client (usually servlets) to call via RMI.
  - A bean class needs to implement some methods to be informed of some events:
    - Session Beans
    - Entity Beans
Session Beans

- void `ejbActivate()`  
- void `ejbPassivate()`  
- void `ejbRemove()`  
- void `setSessionFactory(SessionContext ctx)`

Stateful or Stateless
“Use connection pooling transaction management provided by the EJB server”
Entity Beans

- Entity Beans
  - void `ejbActivate()`
  - void `ejbPassivate()`
  - void `ejbRemove()`
  - void `ejbLoad()`
  - void `ejbStore()`
  - void `setEntityContext(EntityContext ctx)`
  - void `unsetEntityContext()`

- Each entity bean usually associate with one row from some table in the database
Entity Beans
J2EE Framework
Configuration Parameters

- Implementations of DynaServer
  - Servlets Only
  - Servlets + Session Beans
  - Servlets + Entity Beans
    - BMP
    - CMP
  - Servlets + Session Beans + Entity Beans (CMP)
    - With Local Interface
    - Without Local Interface

- Implementation of Container
  - Proxies
    - Pre-compiled
    - Reflection
  - Communication
    - General RMI
    - Jeremie (communication optimization)
## Code Base

<table>
<thead>
<tr>
<th></th>
<th>Servlets</th>
<th></th>
<th>Beans</th>
<th></th>
<th>Total</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Classes</td>
<td>Lines of code</td>
<td>Classes</td>
<td>Lines of code</td>
<td>Classes</td>
<td>Lines of code</td>
</tr>
<tr>
<td>Servlets-only</td>
<td>25</td>
<td>4590</td>
<td>-</td>
<td>-</td>
<td>25</td>
<td>4590</td>
</tr>
<tr>
<td>Session beans</td>
<td>22</td>
<td>2730</td>
<td>51</td>
<td>5270</td>
<td>76</td>
<td>8000</td>
</tr>
<tr>
<td>EB CMP</td>
<td>23</td>
<td>3980</td>
<td>40</td>
<td>6780</td>
<td>63</td>
<td>10760</td>
</tr>
<tr>
<td>EB BMP</td>
<td>23</td>
<td>3980</td>
<td>40</td>
<td>9850</td>
<td>63</td>
<td>13830</td>
</tr>
<tr>
<td>Session façade</td>
<td>22</td>
<td>2660</td>
<td>85</td>
<td>10780</td>
<td>107</td>
<td>13440</td>
</tr>
<tr>
<td>EJB 2.0 local</td>
<td>22</td>
<td>2725</td>
<td>91</td>
<td>11070</td>
<td>113</td>
<td>13795</td>
</tr>
</tbody>
</table>
Max Throughput for

- RMI
- Entity Bean
- Fine-grained access
Session Beans -- scalability

- JBoss 2.4.4
- JOnAS 2.4.4 - RMI
- JBoss 2.4.4 optimized calls
- JOnAS 2.4.4 - Jeremie
- Servlets-only

Throughput in requests/minute

Number of clients

0 100 200 300 400 500 600 700 800 900 1000 1100 1200 1300 1400 1500
Session Beans -- breakdown

![Bar chart showing execution time breakdown for different technologies. The chart compares JBoss, JOnAS, RMI, JBoss optimized, and JOnAS Jeremie. The categories include Garbage Collector, Security, Naming, MM-MySQL, Connection pooling, Transaction manager, Container, Reflexion, Beans (edu.rice.*), and Communication.](chart_image)
Entity Beans – scalability

- JBoss 2.4.4
- JOnAS 2.4.4 - RMI
- JBoss 2.4.4 optimized calls
- JOnAS 2.4.4 - Jeremie

Throughput in requests/minutes

0 20 40 60 80 100 120 140 160 180 200 220 240 260 280 300

2000 1800 1600 1400 1200 1000 800 600 400 200 0

JBoss 2.4.4
JOnAS 2.4.4 - RMI
JBoss 2.4.4 optimized calls
JOnAS 2.4.4 - Jeremie
Session Façade -- scalability

- JBoss 2.4.4
- JOnAS 2.4.4 - RMI
- JBoss 2.4.4 optimized calls
- JOnAS 2.4.4 - Jeremie

Throughput in requests/minute

0 100 200 300 400 500

0 500 1000 1500 2000 2500 3000 3500 4000
Session Façade – breakdown
Other Factors

- Local Interface
  - Get performance gain in all configuration, (optimized communication or not) Why?

- JDK 1.4
  - Reflection speed up is moderate
  - Other part actually slow down
  - No significant improvement
Summary
Summary

- Framework design and application design (separation of modules) has big impact on performance
  - Bean code execution time is small
  - Fine-grained entity bean access does not scale
  - Optimization on remote/local communication, container design all have impact on performance.
Problems

- Impact of different performance of backend database server
- Tuning of performance parameters?
  - E.g. TomCat worker thread number.