

EJB Performance

SeaJUG Presentation

March 19, 2002

Dennis Sosnoski
Sosnoski Software Solutions, Inc.
<http://www.sosnoski.com>

Overview of talk

- Enterprise JavaBean (EJB) background
 - EJB basics, EJB 1.1 vs. 2.0
 - Entity, Session, and Message-driven beans
 - Application architecture
 - EJB Performance Issues
 - Communications overhead
 - Database overhead
 - Transaction and application server issues
 - Conclusions and Resources
-

Why use EJBs?

- Platform has advantages of:
 - Robust – lose only transactions in progress
 - Scalable – add more servers as needed
 - Secure – security built into design
 - Componentized – software design benefits
 - Java – hardware & OS independent, inherently more secure than other choices
 - Top choice for enterprise developers
-

EJB Open Architecture

- Wide choice of servers available:
 - Free open source – JBoss
 - Budget closed source – Orion, etc.
 - Commercial closed source – BEA Weblogic, IBM WebSphere, iPlanet, Oracle, etc.
 - Use what's convenient/cost-effective
 - Change when necessary
-

Bean Basics

- EJB 1.1 Specification first to see extensive use
 - EJB 2.0 Specification released April, 2001
 - Added Message bean type
 - Dramatically enhanced entity beans
 - Provides better performance and portability
 - 2.0 compliance growing
-

Bean Interactions

- Access to beans via remote interfaces
 - Fully distributed, beans can be anywhere
 - Same methods for calls between beans and from externally
 - EJB 2.0 adds local interfaces
 - Same JVM only
 - Better performance, but less flexible
 - All access uses authorization
-

Entity Beans

- Persistent data structures used by application
 - Two forms of persistence:
 - Bean-managed persistence
 - Container-managed persistence
 - Level of abstraction from database
 - Bean represents a “logical record”
 - Controlled interface for consistent use
 - EJB 2.0 adds container-managed relationships, EJB Query Language
-

Session Beans

- Workflow or task oriented
 - May access data directly as well as through entity beans
 - Stateless vs. stateful session beans
 - Stateless beans just methods with no “memory”
 - Stateful beans conduct conversation with client over time
-

Message–Driven Beans

- Transaction–aware components for asynchronous messages
 - Loosens coupling between sender and receiver
 - Similar to stateless session bean in operation
 - EJB 2.0 Only
-

EJB Application Architecture

- Business methods accessible via:
 - Remote Java (RMI–JRMP, RMI–IIOP), including servlets & JavaServer pages (JSPs)
 - Other languages via CORBA
 - COM interfaces to beans
 - XML web services (including SOAP)
 - Build once, access everywhere
-

Communications Overhead

- Remote calls transport objects
 - Serialization expensive, especially for complex structures – object graph issue
 - Added cost for serialized object transport
 - Required even inside same JVM
 - EJB 1.1 handled through object passing
 - Granularity of access methods – passing data objects rather than values
 - Structuring beans for internal communications
-

Local calls

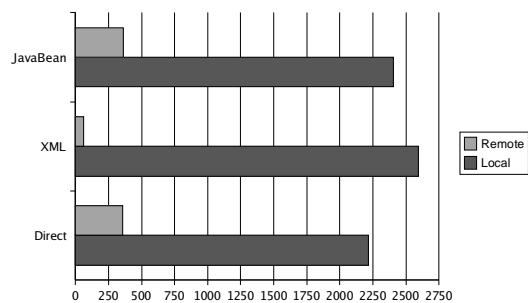
- Added in EJB 2.0
 - "Direct" interface to bean methods
 - Objects passed directly, not duplicated
 - Must be intra-JVM
 - Limitations a consequence
 - Generally inside an EAR only
 - Clustering flexibility reduced (including JSP-servlet front end, if used there)
-

Performance example

- Get information for JSP page from EJB
 - "JavaBean" interface passes read-only value object
 - XML interface passes JDOM Document
 - Direct interface calls EJB methods for each value
 - Same output generated in all cases
 - Code walkthrough and demo
-

Results of testing

- Pages in 10 seconds:



- Benefits of local calls obvious
-

Database Overhead

- The fundamental issue – only the database knows the state of the data
 - Different clustered system can update
 - Changes can come from outside application
 - Not unique to EJB architecture
 - No general solution
 - Some partial solutions/guidelines
-

Basic Entity Bean Issues

- Bean partitioning:
 - Each bean a row in some table
 - Not every row of every table a bean!
 - Bean–managed vs. Container–managed:
 - EJB 2.0 CMP potentially simpler and more portable (especially with CMR, EJB QL)
 - Vendor–dependent CMP optimizations can help performance, at cost of portability
 - Bean–managed gives more detailed control
-

Bean–Managed Persistence

- Bean–managed persistence issues:
 - ejbLoad() and ejbStore() calls – dirty flag
 - Lazy loading of dependent objects
 - Cache values within transaction
 - Use prepared statements or stored procedures
 - ORM frameworks can help reduce complexity:
 - TopLink
 - CocoBase
 - Framework becomes dependency for your application
-

Transaction Performance Issues

- Database transaction choices and performance implications
 - TRANSACTION_READ_UNCOMMITTED
 - TRANSACTION_READ_COMMITTED
 - TRANSACTION_REPEATABLE_READ
 - TRANSACTION_SERIALIZABLE most costly
 - Transaction partitioning concerns
 - Don't create new transactions unnecessarily
 - Avoid waits for external action
 - Optimistic locking possibilities
-

Special approaches

- Session beans can bypass entity bean layer
 - Useful for preliminary results list – use entity bean for details
 - Data that rarely changes can be cached
 - Distorts the structure, so use with care!
-

Application Server Tuning

- General issues
 - Choice of JVM and JIT options
 - Java heap size
 - Thread counts and OS overhead
 - Bean pool sizes, timeouts, etc.
 - Clustering vs. multiprocessor systems
 - Special server extensions
 - Non-shared database access options
 - Transaction isolation options
 - OS optimizations
-

Conclusions

- Establish requirements up-front
 - Consider data flows in design
 - Avoid excessive remote interface calls
 - Use EJB 2.0 for CMP and local interfaces
 - Many issues specific to application:
 - Early prototyping allows reality checks
 - Instrument code for tuning design
 - Use application server tuning
 - EJB scalability is automatic, but performance requires work
-

Resources

- EJB home – [http://java.sun.com/products/ejb/EJB Learning Center](http://java.sun.com/products/ejb/EJB%20Learning%20Center)
 - – <http://java.sun.com/products/ejb/training.html> EJB Online Tutorial –
 - <http://developer.java.sun.com/developer/onlineTraining/Beans/EJBTutorial/>
 - JBoss – <http://www.jboss.org/Orion> –
 - <http://www.orionserver.com/>
 - BEA Weblogic –
 - <http://www.bea.com/products/weblogic/server/index.shtml> WebSphere
 - –
 - <http://www-4.ibm.com/software/webservers/appserv/iPlanet> –
 - <http://www.iplanet.com/Oracle> –
 - <http://www.oracle.com/ip/deploy/ias/index.html>
 - Author web site – <http://www.sosnoski.com>
-