The Road to JBoss

Designing a Migration Path

Presentation Overview

- Discuss experiences with introducing Professional Open Source to a traditional financial services company.
- Share the political and external forces that could influence your company's introduction to JBoss.
- Showing how having a flexible design made introducing Hibernate easier.

Bio - Steven Whyte

- Senior J2EE Lead at Evolved Digital Systems.
- Was Server-Side Lead on the project discussed here.
- Over 15 Years software development experience.
- 9 Years working with Java.

Company "X"

- One of the largest financial services companies dealing with HR and Payroll Services.
- In 2001 began planning the replacement of its legacy systems and merging all of its purchased products into one unified product offering.
- After building a proof of concept using Swing Client, talking to J2EE and JBoss, decision was made to move forward with project.
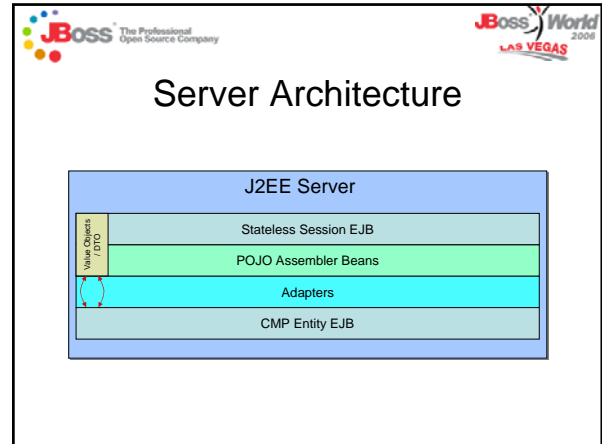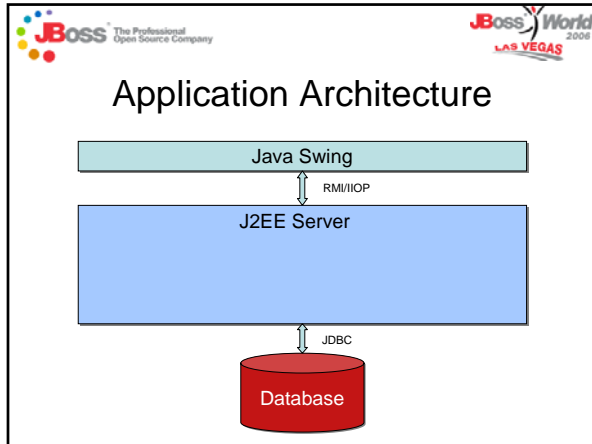

Company "X"

- Development was to begin with the assumption that the server side vendor to be decided in the future, so must adhere to J2EE standard and begin using BEA Weblogic.

Project Overview

- Data Design Team
  - 10 Data Modelers
  - 8 DBA / Logical Modelers
- Analysis Team
  - 12 to 15 Analyst
  - 8 UI Designers
- Development Team
  - 30 Server Side Developers
  - 25 UI Developers
  - 15 QA
- Operations
  - 6 Online Engineering
- Performance
  - 4 Performance Analyst
- Project Management
  - 10 Project Managers
- R&D
  - 4 R&D Resources
- Development Management
  - 7 Managers

Application Architecture

Java Swing

RMI/IIOP

J2EE Server

JDBC

Database

Server Architecture

J2EE Server
Stateless Session EJB
POJO Assembler Beans
Adapters
CMP Entity EJB

Value Objects / DTO

Design Decisions

- One to One mapping between Entity EJB and Database Tables
- No CMR Relationships
- Domain Objects to hold object graphs
- Object graphs (relationships) built by walking the relationship one at a time (N+1)

Maintaining J2EE Standards

- Not to take advantages of vendor specific extensions.
  - Standard EJB-QL
    - No Order By / OrderBy
    - No SubQueries

Obstacles in Introducing JBoss

"Open Source Myths"

- Coded by hackers.
- No formal Support.
- No formal QA.
- How can they compete against big vendors?

Vendor Selection 2003

- Vendors
  - BEA WebLogic
  - IBM Websphere
  - SUN Application Server

Vendor Selection 2003

- Proof of concept proving
  - Clustering Features
    - Fail Over
    - Load Balancing
  - Performance
  - Vendor Responsiveness

Vendor Selection 2003

- BEA Weblogic
  - Fail Over and Load Balancing work except automatic JMS failover.
  - Performance was satisfactory, compared to what we were using, WebLogic!
  - Already use their products, you should already know our responsiveness!

Vendor Selection 2003

- IBM Websphere
  - Clustering Feature proved.
  - Performance was good, although it was ran in IBM facilities.
  - Treated executives like kings! You know where this is going!

Vendor Selection 2003

- Sun (Insert Name Here) Application Server
  - Clustering feature proved.
  - Performance was good in Sun's labs.
  - Very Technically responsive.

Vendor Selection 2003

And the Winner is…..

IBM.

Moving Forward with IBM

- Meet with Global Services to discuss migration plans.
- Back up the truck with new IBM hardware.
- And Oh by the way, you look like you could use Identity Management, System Monitoring, and other various word salads.

The Migration

- Several IBM Consultants to assist us in converting to Websphere.
- We are giving 6 weeks to convert while 40+ developers are still developing.
- Develop training plans for development moving forward.

Migration Issues

- Currently developing on JDK 1.4.X, but Websphere only supported 1.3.X.
  - ➢ ☐ The swing portion of the application and certain persistence mechanism took advantage of 1.4 feature and could not be removed from code.
- Build process was Ant driven and IDE independent.
  - ➢ IBM tried to force us to use Websphere Studio for build process, but we stood our ground, "Make it work to our process!"

Migration Issues

- 1 Gig PC too small for footprint of Websphere Studio, Application Server and Local Database.
  - ➢ Need to get new PC, sorry new IBM PC only could upgrade to 1 Gig.
- Various Bugs with product, although IBM was very responsive to issues.

Migration Outcome

- With several more IBM consultants, we could not get our application running on Websphere.
  - ➢ Decision was made to stay with BEA, "The Application runs on it!"

Lessons Learned from Migration

- Value of XDoclet was a big help in making migration between application servers as painless as possible.
- The value of having a tool to fit your processes.
  - ➢ If we had used Studio instead of our process, we estimated that the average time a developer spent coding would increase 25%.

Moving Forward with BEA

- It works, so let's use it!
- BEA's feelings were hurt, so responsiveness was even slower, and boy aren't you lucky you have us!
- Application is growing, so development is slowing.

Why Not JBoss!

- Developers and Some Technical Managers say why not JBoss!
  - How can we prove JBoss is a viable vendor choice?
  - How can we take an issue and show how JBoss can help us address it?
  - Time for a covert, grass-roots developer initiative.

Application is growing, so development is slowing!

- Average build time had increased to 23 to 44 minutes. (Varied due to PC size)
- In a schedule oriented project, this sounds like a great opportunity.
- What was the culprit?
  - Using CMP Beans required us to use appc to compile cmp code, stubs and skeletons.
  - Appc took almost 15 minutes of the 23 minute build.

Fitting JBoss into the Picture

- Company still had it's misconceptions of Open-Source.
- Let's use JBoss in Development only.
  - JBoss does not require appc.
  - We have Xdoclet already in use, so support for both WebLogic and JBoss should be easy.
  - 15 Minutes / Build * 4 builds / Day * 40 Developers = 40 hours/day saved!

Migrating to JBoss

- Must not break support for BEA Weblogic.
- Must not affect build process, other than reduce build times considerably.
- Must not affect deployment processes with Weblogic.

Migration Issues

- Deployment Descriptors
  - XDoclet should handle application server configurations.
  - Must change any @weblogic tag to a supportable @ejb tag if possible. Only use vendor tags only as last resort.
  - We were developing to J2EE standards right! Or were we ……………

Migration Issues

- Deployment Descriptors - continued
  - Many pieces of code took advantage of Sub-Selects.
    - Broke these queries out to JDBC Queries.
  - Use of ORDERBY pervasive through out the application, but JBoss supports ORDER BY.
    - Yes we need to use vendor specific XDoclet tags to accommodate this problem, but wait…..

Migration Issues

- Deployment Descriptors - continued
  - Why is BEA putting its vendor extensions in the standard ejb-jar.xml?
    - We need to make the ejb-jar.xml deployment descriptors for JBoss and Weblogic. Easy enough……
    - Running XDoclet twice through 500+ Entity Beans and 70 Session Bean added another 2 minutes to build!
    - Customize XDoclet to generate both a Standard ejb-jar.xml for Jboss and a "Standard" ejb-jar.xml for Weblogic! (added 10 seconds to generation)

JBoss in Development

- Now that we can run the application in JBoss, we can compare to Weblogic.
- Performance
  - Suite of Unit and Regression Testing ran quicker on JBoss.
  - JBoss used ~30% less memory than Weblogic.
- Support
  - Issues were resolved via forum considerably faster than with BEA support.
- Let's save these metrics for when Weblogic falters.

Application Issues

- Under performance load testing application crushes servers.
  - Entity Beans identified as a bottleneck along with adapter layer.
  - Need to change persistence mechanism.

Persistence Layer

- Need to use persistence model that reduces adapting from Bean to POJO.
- Development suggest Hibernate to replace Entity CMP's.
- Take small piece of system with problems and prototype Hibernate persistence.

Hibernate Prototype

- Selected part of system where large object graphs were required.
- Large object graphs required many queries to build.
  - Graph required 35 queries to populate object graph.
  - Using Hibernate, this was reduced to under 5 queries.

Hibernate Prototype

- After prototyping Hibernate, performance was decreased from 35 seconds to under 3.
- Development Management approves using Hibernate for persistence mechanism, but…………

Hibernate Conversion

- Must only take less than 3 weeks, remember all that matters is schedule!
- We estimated it would take at least 3 months to replace Entity CMP, and adapter layer.
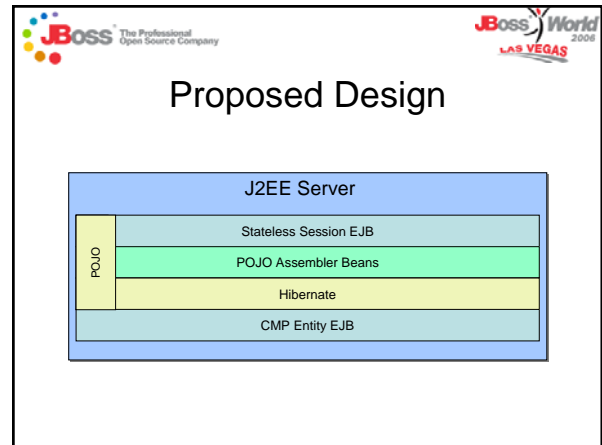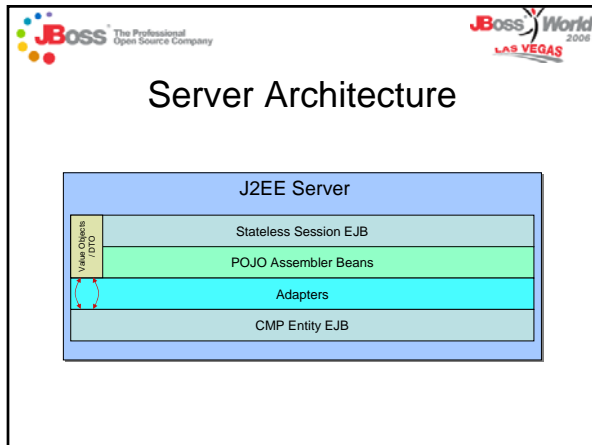
Server Architecture

J2EE Server	
Value Objects /DTO	Stateless Session EJB
	POJO Assembler Beans
	Adapters
	CMP Entity EJB

Proposed Design

J2EE Server	
POJO	Stateless Session EJB
	POJO Assembler Beans
	Hibernate
	CMP Entity EJB

Hibernate Conversion

- Decision
  - ➤ Must be done in 3 weeks!
- Alternatives
  - ➤ Stay with what we have and fix only critical performance area.
  - ➤ Or…………..
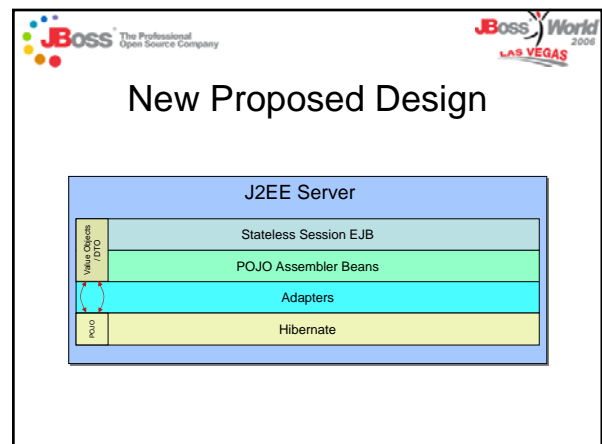
New Proposed Design

J2EE Server	
Value Objects /DTO	Stateless Session EJB
	POJO Assembler Beans
	Adapters
POJO	Hibernate

New Proposed Design

- Replace Entity CMP with Hibernate.
- Still need adapter layer to bridge Hibernate to Value Objects.
- And now when we find performance issues, we will remove adapters and complete conversion.

Object Model

Hiding Persistence Layer

- Calls from adapter to get entities, intercepted by EJBHomeProxy, and returns Hibernate Value Objects
- Calls to finders and EJBQL intercepted by EJBHomeProxy and routed to FinderService which queries Hibernate.
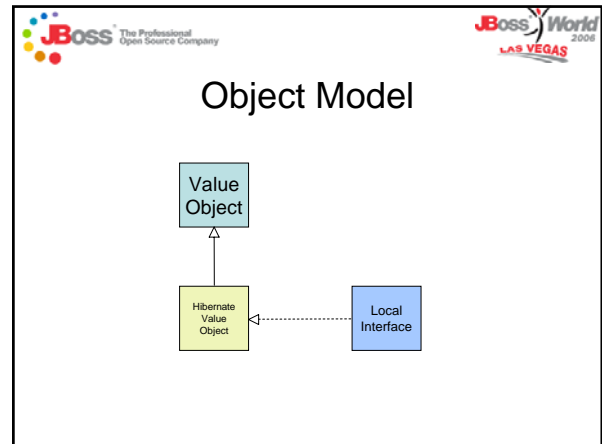
Hibernate Conversion

- Replacement of Entity CMP's replaced by Hibernate in 4 weeks.
- Performance gains realized.
- Phase Two
  - ➢ Remove adapters in future.
  - ➢ Let Hibernate manage relationships.

Two JBoss JEMS Successes

- Application runs better under JBoss.
- Application runs better with Hibernate as persistence layer, even on Weblogic.
- Now that we removed 700 Entity Beans, build time is reduced again!
- What else do we need to do to use JBoss products in production!

What's Next

- Well since Hibernate has improved application performance, Weblogic is running better.
- Management still does not accept JBoss as a viable alternative.
- Must wait for Weblogic to hit wall.
- Big user test coming up!

Application Hits Wall

- We suggest JBoss as alternative.
- Open Source Myths are brought out.
  - Can get support from JBoss! And by the way, just from user forums, the average support issue was resolved quicker on JBoss.
  - Code is very stable, did not require patches.
  - During Hibernate Conversion issues were resolved in hours not days.

Let's Look at JBoss!

Let the Games Begin!

- Politics
  - Departments with Loyalty to BEA
    - Will do little to help!
  - Sales Reps with Executives ears
    - White Paper Scare!
      - Weblogic 8.1 runs better than JBoss 2.x
    - Fuzzy Math
      - Even though you are not paying for the software, the support will cost you more in the long run than if you bought our software and our support contract.

JBoss into Production

- JBoss comes in to assist setting up of JBoss on production servers.
- Performance testing proves JBoss does work better. Development knows this already!
- JBoss Clustering runs more efficiently than Weblogic.

Lessons Learned

- Sell the use of JEMS products by introducing the benefits to what management understands.
  - Our project was all about schedule, so anything that reduced schedule, got immediate consideration.
- Watch the political forces inside and outside of your organization.
  - Size up your organization, who is aligned with who. Who do the vendors align with internally.

Lessons Learned

- Try to dispel the open source myths.
  - Professional Open Source is developed by professional developers and, in my opinion, offers better support.

Thanks for your Time

Questions