GlassFish in real life

Deployment Stories

Wotif.com
GlassFish Case Study

Greg Luck
Chief Architect, Wotif.com

Wotif.com Case Study
Key facts and figures

- Australia's No 1 Hotel Booking Site

- Rapidly Growing elsewhere

- 1,000,000 users

- 10,000 concurrent users

- 3.6 million room nights sold per year

- 9000 hotels

- Very Large Pages by design (some > 1.5MB)

Source: http://blogs.sun.com/stories/entry/wotif
http://www.networksasia.net/ena/article/articleDetail.jsp?id=393040
http://info.wotif.com/pdf/Wotif.com_Facts_Figures.pdf

Wotif.com Case Study
Screen shots – Home Page

Source: http://wotif.com

Wotif.com Case Study
Screen shots – Search Results

- + 15 more screenfuls for a total of 1.5MB

- Support many of these per second

Source: http://www.wotif.com/search/Simple?refine=simpleSearch&country=1&region=1&viewType=all

Wotif.com Case Study

Technology Platform

- Sun AMD64 Servers

- RHEL5

- 64bit Java technology – 5 moving to 6

- GlassFish V1 UR1

- Open Message Queue

- Oracle & MySQL

- Full stack Java EE 5

- Also lots of open source, including ehcache

Sun

Wotif.com Case Study
Why we chose Glassfish?

- Needed to move off Orion

- Wanted Java EE 5, open source or cheap

- Wanted commercial support

- Path to the future

- Considered JBoss, Geronimo, non-Java EE like Tomcat and Resin.

- Evaluated by doing a small test port of the busiest pages and running it in our performance and stress test environment

- Worked Great!

Sun microsystems

Wotif.com Case Study
Porting Experience

- Started with 8 spikes including performance, stability and security

- 5 devs took one month

- J2EE 1.3 > Java EE 5 (Servlet 2.5, JSP 2.1, EJB 3)

- Required extensive regression testing

- Dev issues:
  - Bugs
  - Memory use
  - Startup and deploy times

- Upgraded Dev machines to 3GB and some to dual core

- Went live January 2007

Wotif.com Case Study
Bugs Raised

- We have raised around 10 bugs on Glassfish. Most of these are now closed.

- There is a PermGen memory leak on redeploy. Developers have to restart after about 10 deploys.

- The guide to configuring domain.xml is very patchy. It is a lot of trial and error to tune your app. We need a tuning guide. I have already suggested this to Jeanfrancois Arcand.

- The QuickStart feature is **very** buggy. We turned it off to resolve a whole host of bugs we had. We suggest it is turned off by default until it is fixed.

- Unfortunately we hit a show stopper in SSL with Grizzly. Used Catalina which is done using a system property

Wotif.com Case Study

Extending Glassfish with Other Open Source Projects

- Ehcache – distributed cache. We use inter alia for HTTP response caching

- Spnego – A JMAC plugin for Kerberos

Wotif.com Case Study
Ehcache

- Required some minor changes to the Caching Filter to support GlassFish. Version 1.2.4 supports GlassFish

- Hibernate Persistent Object and Collection Caches

- Distributed Page Caches in a "Customer" cluster

- AJAX XML Response Caches

- Distributed Java technology Object Caches

- 5 GB Heap. Requires fine tuning of GC

- Uses SelfPopulating and Blocking Cache

- Web Cache Full Page & Fragment Web Filters

- Disk Store and Persistent Disk Stores

Sun

Wotif.com Case Study

Spnego

- Spnego is an SPNEGO and Kerberos plugin for GlassFish.

- Enables GlassFish to participate in a Single Sign On environment using Kerberos.

- Works with IE, Firefox and Safari on Linux, Windows and Mac OS X

- First version finished and will be released soon

- Requires GlassFish V2 and JDK6

- Developed by Wotif.com and Sun.

- CDDL License

- http://spnego.ocean.net.au

Wotif.com Case Study
Future Plans

- Moving to GlassFish V2 in August

- Implementing an Enterprise Message Queue using Open Message Queue 4.

- Moving to Java 6 in August

- Implementing an enterprise wide Kerberos Single Sign On IAM. Will start using SPNEGO module with GlassFish in August.

- Maybe: Port two RoR apps to JRuby and run in GlassFish

Relevant Sessions at JavaOne

Ehcache

Session ID: TS-6175 Session Title: Distributed Caching, Using the JCACHE API and ehcache, Including a Case Study on Wotif.com

Track: The Next Generation Web

Room: Esplanade 307-310

Date: Tuesday 08-MAY-07 Start Time: 10:50

SPNEGO

JMAC Birds of a Feather Session (Ron Monzillo of Sun)

CommunityOne

Q&A

Sun microsystems

Harvard's Dataverse Network on GlassFish

Merce Crosas

Who are we? What do we do?

Introduction to the Dataverse Network

- The Dataverse Network is a data-sharing project housed at Harvard's Institute of Quantitative Social Science.

- It has been written from scratch since 2006 (based on its precursor, the Virtual Data Center).

- It allows researchers and students to:
  - Disseminate and cite data
  - Replicate published work
  - Perform on-line data analysis

- It's currently used by Harvard and MIT at: http://dvn.iq.harvard.edu/dvn (Beta)

- Other Universities and research centers are adopting it as well (for example, ODUM at the UNC).

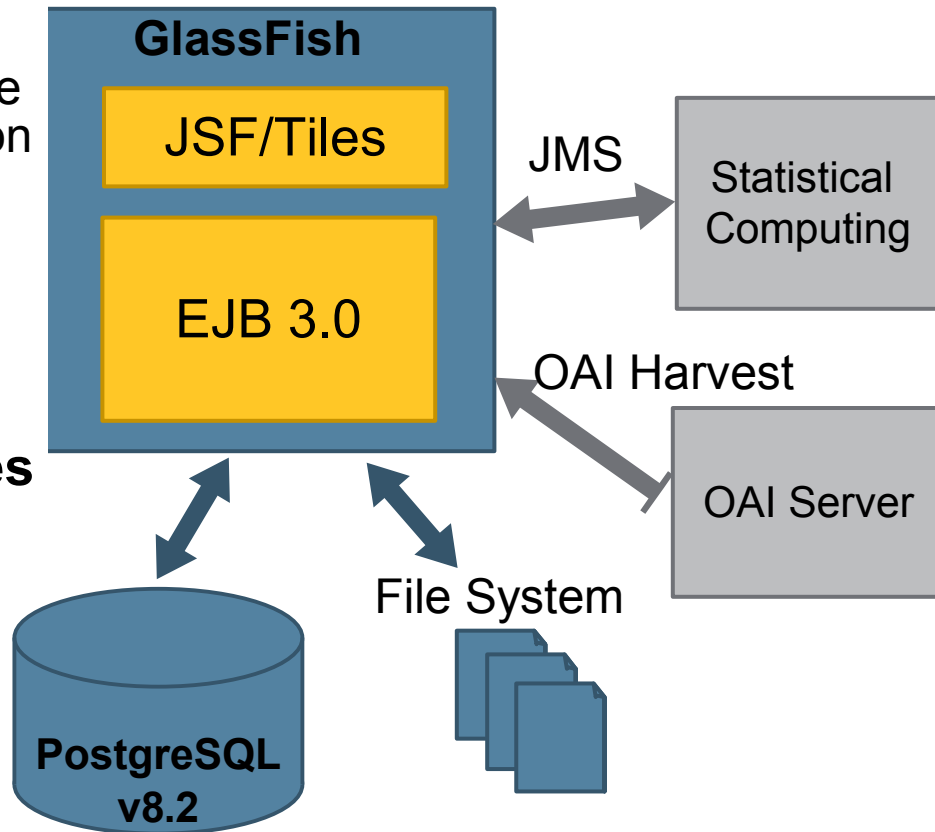- We are extending it to support data in other research fields.

Why GlassFish?

- Two main reasons:
  - Free, Open Source – The Dataverse Network is open source and relies on other open source technologies.
  - Java EE 5 compliant – Need a maintainable and rapid to develop architecture: EJB3 and JavaServer Faces.

Other Open Source Technologies we use:

- Apache Lucene 2.0 (index/search)
- Shale Tiles Integration
- OAIHarvester, OAICat
- R, Zelig (statistical computing)

GlassFish

JSF/Tiles

EJB 3.0

JMS → Statistical Computing

OAI Harvest ← OAI Server

PostgreSQL v8.2

File System

Configuring GlassFish: postgreSQL

Configuring GlassFish: JMS

Connection Factories (2)

New... | Delete | Enable | Disable

	JNDI Name	↑↓
☐	jms/DSBQueueConnectionFactory	
☐	jms/IndexMessageFactory	

Resources > JMS Resources > Connection F[...]

Edit JMS Connection Factory [...]

Editing a Java Message Service (JMS) connectio[...]
connector resource.

Load Defaults

General Settings

JNDI Name: jms/DSBQueueConnectionFactory

Pool Name: jms/DSBQueueConnectionFactory

Type   javax.jms.QueueConnectionFactory

Description: [   ]

Status: ☑ Enabled

Pool Settings

Initial and Minimum Pool Size: [8]   Connections
Minimum and initial number of connections maintained in the pool

Maximum Pool Size: [32]   Connections
Maximum number of connections that can be created to satisfy client requests

Pool Resize Quantity: [2]   Connections
Number of connections to be removed when pool idle timeout expires

Idle Timeout: [300]   Seconds
Maximum time that connection can remain idle in the pool

Max Wait Time: [60000]   Milliseconds
Amount of time caller waits before connection timeout is sent

Glassfish Message Beans ←→ http → Statistical Computing

Destination Resources (2)

New... | Delete | Enable | Disable
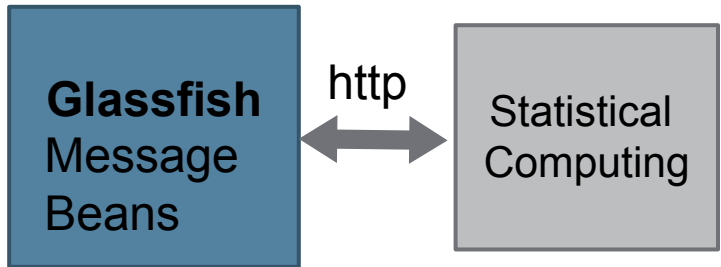
	JNDI Name	↑↓	Enabled	↑↓	Resource
☐	jms/IndexMessage		true		javax.jms.Queue
☐	jms/DSBIngest		true		javax.jms.Queue

Sun microsystems

Configuring GlassFish: Mail Sessions

Sun Java™ System Application Server Admin Console

Enterprise Applications
- Web Applications
- EJB Modules
- Connector Modules
- Lifecycle Modules
- Application Client Modules
- Web Services
- JBI
  - Service Assemblies
  - Components
  - Shared Libraries
- Custom MBeans
- Resources
  - JDBC
    - **JDBC Resources**
      - jdbc/__TimerPool
      - jdbc/__CallFlowPool
      - jdbc/__default
      - jdbc/VDCNetDS
    - **Connection Pools**
      - __CallFlowPool
      - __TimerPool
      - DerbyPool
      - VDCNet-devpool
  - JMS Resources
    - **Connection Factories**
      - jms/DSBQueueConne
      - jms/IndexMessageFa
    - **Destination Resource**
  - JavaMail Sessions
    - mail/notifyMailSession
  - JNDI

Resources > JavaMail Sessions > mail/notifyMailSession

Edit JavaMail Session   [Save]

A JavaMail session resource represents a mail session in the JavaMail API.

[Load Defaults]

General Settings

JNDI Name:	mail/notifyMailSession
Mail Host:	vdc-build.hmdc.harvard.edu
	DNS name of the default mail server
Default User:	dataversenotify
	User name to provide when connecting to a mail server; must contain only alphanumeric, underscore, dash, or dot characters
Default Return Address:	do-not-reply@vdc-build.hmdc.harvard.edu
	E-mail address of the default user
Description:	
	Makes it easier to find this session later
Status:	☑ Enabled

Advanced

Store Protocol:	imap
	Either IMAP or POP3; default is IMAP
Store Protocol Class:	com.sun.mail.imap.IMAPStore
	Default is com.sun.mail.imap.IMAPStore
Transport Protocol:	smtp
	Default is SMTP
Transport Protocol Class:	com.sun.mail.smtp.SMTPTransport
	Default is com.sun.mail.smtp.SMTPTransport
Debug:	☐ Enabled

Additional Properties (0)

[Add Property] [Delete Properties]

Name	Value
No items found	

Setting E-mail notifications to users

Sun microsystems

Developing and Deploying our Application

- We use NetBeans 5.5.1 as our development environment.

- In Development, we deploy the EAR file into GlassFish through NetBeans.

- In Production, for deployment we use ANT scripts generated by NetBeans.
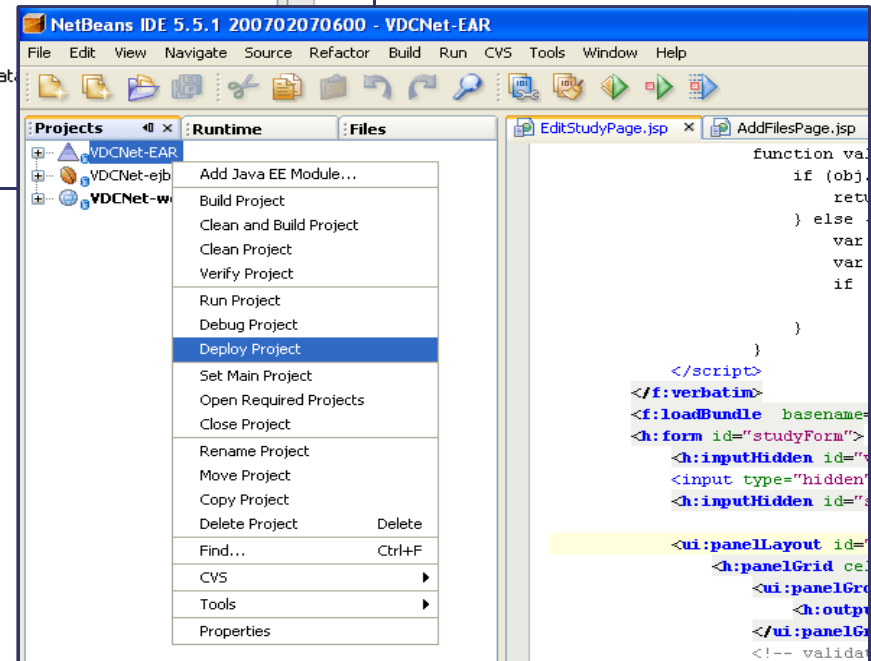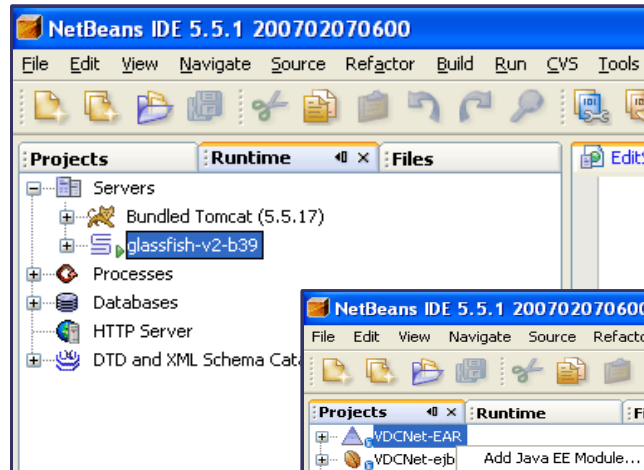
Our Production (Beta)

- The IQSS Dataverse Network is hosted at Harvard.

- OS production: Linux

- GlassFish Version: v2-b41

- Thousands of users once it's in full production.

- Any researcher, department or center can create their own "dataverse".

Fully Custom "Dataverses"

A dataverse is an on-line virtual archive for research data

Journals

Researchers

Archives

Dataverse Functionality

Persistent data citation

Download data files and documentation

Data Analysis

Admin Functionality

Add Studies, Upload Files, Create collections, custom settings (edit banner, homepage), permissions, etc

For More Information

- This Friday at 12:10!
  - TS 4656: Harvard's Dataverse Network: A JSF/EJB 3.0 Technology Data Sharing Solution on Java EE 5
- IQSS Dataverse Network at:
  - http://dvn.iq.harvard.edu/dvn/
- Dataverse Network Project Site:
  - http://thedata.org/
- Gary King, "An Introduction to the Dataverse Network as an Infrastructure for Data Sharing":
  - http://gking.harvard.edu/files/dvn.pdf

DEMO

IQSS Dataverse Network

GlassFish in Real Life: Peerflix.com

Cyril Bouteille

CTO
Peerflix, Inc.
http://www.peerflix.com

Peerflix: the DVD Trading Community

Why rent when you can own?

- Created a vibrant community around movies to:
    - Discover DVDs
    - Connect/converse with other members
    - Trade the DVDs you have for ones you want

- Offering Peer-to-Peer DVD trading service

1 Join for FREE

2 Make 2 lists
- Add DVDs you WANT.
- Add DVDs you HAVE to sell.

3 Trade DVDs
- Mail DVDs to members
- Receive DVDs from members
- Keep or trade what you get

Easy DVD mailing System

Try it FREE

How does the Peermailer work?
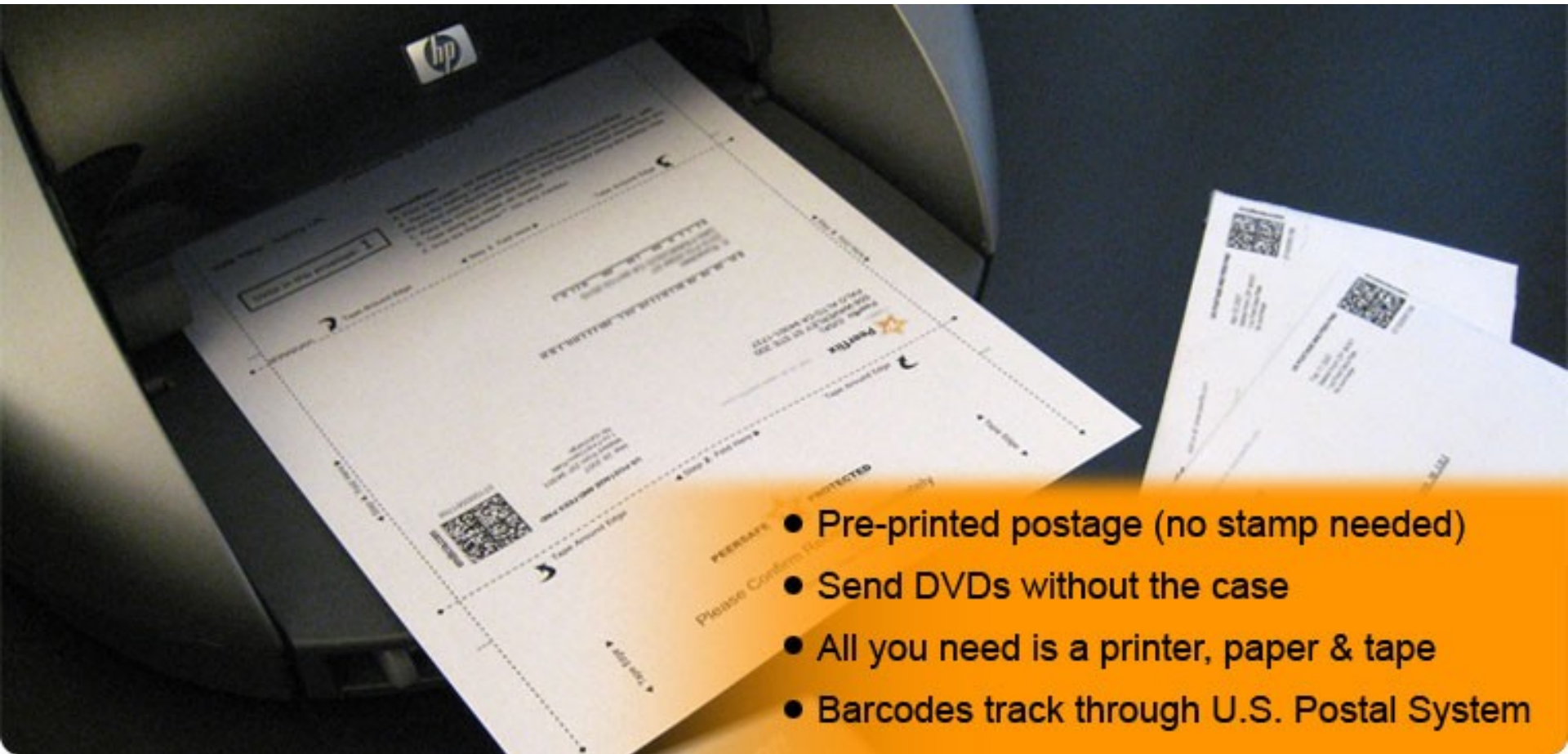
- Pre-printed postage (no stamp needed)
- Send DVDs without the case
- All you need is a printer, paper & tape
- Barcodes track through U.S. Postal System

Early-stages of Peerflix

- Peerflix started in 2004 with an ASP.NET web site

- Growth to 300k members led to scalability issues:
    - Complexity => longer TTM for features
    - Traffic => reduced availability @ peak times

- In 2006, a new platform was needed:
    - Extensibility to add features + expand in new markets
    - Horizontal scalability to support traffic growth
    - Low cost of ownership and operation
    - Evolution vs. Revolution?

Growth through a next-generation Enterprise Java platform

Not Invented Here

- Complete re-architecture of the web site using Java EE & open-source technologies:
  - Software redesign with JSP/JSF, EJB3 & JDO2
  - Glassfish V1 UR1 / SJSAS 9.0.1 PE
  - Solaris 10 on AMD-based X Fire servers
  - Kodo Java Data Objects on MySQL 5 Database
- Peerflix V2 launched on December 18th, 2006
  - A 6-month project with a team of 8 engineers
  - PR brought 360% increase in traffic volume
  - New platform was stable with no unplanned downtime

Why Glassfish?
Why open-source?

- No licensing costs for development and production use

- Affordable production support options
  - Integrated from server to JVM and Operating System

- Ahead in Java EE 5 compliance

- Strong vibrant community with active forums

- Low-maintenance w/ easy to use admin tools

- Clean and coordinated server architecture

DEMO

http://www.peerflix.com

GlassFish in real life

Deployment Stories

http://blogs.sun.com/stories

Sun microsystems