



Project GlassFish

The Open Source Java EE Application Server

Binod P.G, Sivakumar Thyagarajan
Sun Microsystems





The Future of Java EE Glassfish

- **Overview**
- Cool/New features in GlassFish
- Sun Java System Application Server
- How to get involved
- Resources



What is GlassFish?

- Sun's **Open Source Application Server Platform Edition 9**
 - > CDDL license
 - > Open process
- Open access to code and binaries
 - > CVS access to source code
 - > Nightly builds, weekly promoted builds
- Must support Java EE **compatibility**
- Supports all platforms
 - > Linux, Windows, Solaris, Solaris_X86 and MacOS



The Future of Java EE Glassfish

- Overview
- **Cool/New features in GlassFish**
- Sun Java System Application Server
- How to get involved
- Resources



GlassFish Features

- Java EE 5 (formerly J2EE)
- Grizzly
- On-demand initialization
- SOA / Java Business Integration (JBI)
- Callflow
- Extensive self-management framework
- Java Web Start - appclient integration
- Web Services – WS-Management/ebXML registry integration/Fast Infoset



The J2EE™ Challenge

- J2EE is enormously powerful
 - > The industry standard for robust enterprise apps
- But that power sometimes gets in the way
 - > Too difficult to get started
 - > Even simple apps need boring boilerplate
- Can we keep the power, but make typical development tasks simpler?
- **YES:** and that is the focus of Java EE 5!



Ease of Development Improvements in Java™ EE 5

- POJO-based programming (Plain Old Java Object)
 - > More freedom, fewer requirements
- Extensive use of annotations
 - > Reduced need (practically none!) for deployment descriptors
- Resource Injection
 - > Inversion of control
- New APIs and frameworks



EJB 3.0 Example

```
@Stateless public class PayrollBean
    implements Payroll {

    @Resource DataSource empDB;

    public void setBenefitsDeduction (int empId, double
deduction) {
        ...
        Connection conn = empDB.getConnection();
        ...
    }

    ...
}
```



EJB 3.0 Client Example

```
// EJB 3.0 client view
```

```
@EJB ShoppingCart myCart;
```

```
...
```

```
Collection widgets = myCart.startToShop("widgets");
```

```
...
```



Java™ EE 5 Major Features

- Simplified web services support
- More web service standards support
- Greatly simplified EJB™ development
- New persistence API
- Easy web applications with JavaServer™ Faces



Web Service Example

```
@WebService(name="MySimpleWS");  
public class RandomClass {  
  
    @WebMethod  
    public String sayHello(String s) {...}  
  
    public void unpublished() {...}  
}
```



Web Service Client Example

```
@WebServiceRef (
    wsdlLocation=
        "http://localhost:8080/SayHelloService?WSDL");)

static javaone.SayHelloService wsService;

public static void main(String[] args) {

    javaone.SayHello wsPort = wsService.getHello();
    wsPort.sayHello("FOSS.in Attendees");
}
```



Java™ EE 5

- EJB 3.0 (JSR-220)
- JavaServer Faces (JSR-252)
- StAX (JSR-173)
- Web Services Metadata (JSR-181)
- New persistence API (JSR-220)
- JAXB (JSR-222)
- JAX-WS (JSR-224)
- Common Annotations (JSR-250)



Java™ EE 5 Status

- Most specs available for review now
- Specs at Proposed Final Draft stage – Q4 2005
- Java EE 5 SDK Beta release – Q1 2006
- Java EE 5 Final release – Q2 2006



GlassFish Features

- Java EE 5 (formerly J2EE)
- Grizzly
- On-demand initialization
- SOA / Java Business Integration (JBI)
- Callflow
- Extensive self-management framework
- Java Web Start - appclient integration
- Web Services – WS-Management/ebXML registry integration/Fast Infoset



Introduction to Grizzly

What is the Grizzly HTTP Framework?

- Grizzly is a HTTP framework
 - > Uses lower level Java NIO primitives
 - > Easy-to-use, high-performance APIs for socket communications
- Grizzly brings non-blocking sockets to the HTTP processing layer



Introduction to Grizzly (Cont.)

What Is the Grizzly HTTP Framework?

- Grizzly integrates with the current Apache Tomcat HTTP Connector architecture (Coyote)
 - > Coyote is currently used in Tomcat 3.x/4.x/5.x
- Grizzly replaces Apache Tomcat Coyote HTTP front-end and thread pool
 - > Utilizes its high performant buffers and parsing classes

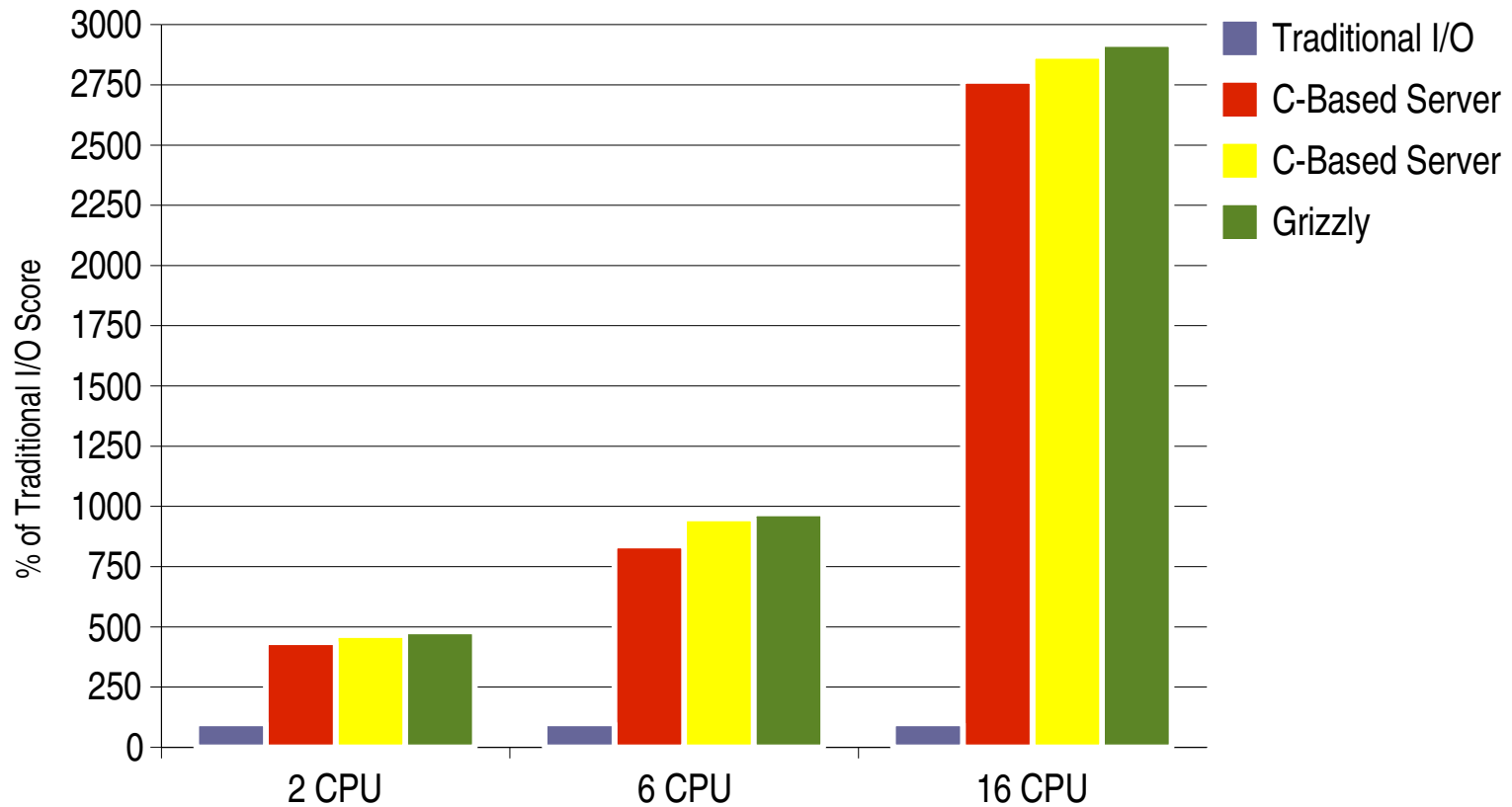


Benchmark Overview

- Typical Servlet/JavaServer Pages™ (JSP™) technology model
 - > Internally developed benchmark
 - > Single Control Servlet
 - > 16 actions
 - > 90% of requests require database interaction
 - > All results formatted through JSP technology
 - > Lots of JSTL 1.1 tags
 - > Session-based, 15 minute average, Keep-alive desirable



Benchmark Results





GlassFish Features

- Java EE 5 (formerly J2EE)
- Grizzly
- On-demand initialization
- SOA / Java Business Integration (JBI)
- Callflow
- Extensive self-management framework
- Java Web Start - appclient integration
- Web Services – WS-Management/ebXML registry integration/Fast Infoset



Java Business Integration

- Enterprise Application Integration is a Big Challenge
 - > JSR 208.
 - > Normalized Message Router
 - > Protocol agnostic integration mechanism.
 - > OpenESB (Sun has open sourced its ESB product based on JSR208)
 - > OpenESB provides out-of-box support for SOAP and JMS.



JBI support in GlassFish

- Java EE webservices (EJBs and Servlets) are automatically enabled in JBI's Normalized Message Router.
- A JAXWS communication from GlassFish can be routed in-process to the Normalized Message Router.



GlassFish Features

- Java EE 5 (formerly J2EE)
- Grizzly
- On-demand initialization
- SOA / Java Business Integration (JBI)
- Callflow
- Extensive self-management framework
- Java Web Start - appclient integration
- Web Services – WS-Management/ebXML registry integration/Fast Infoset



Web service Management

- Webservices are considered first class citizen in GlassFish.
- Management and Monitoring capabilities of webservices are built into the Administration GUI and CLI of GlassFish.



EbXML Registry

- Supports UDDI 3.0
- In built resource adapter to discover and publish web-services descriptors in the registry.
- Supports JAXR APIs.



Fast Infoset

- Optimized Binary XML transport mechanism.
- Webservice communication can transparently use Fast Infoset capabilities.
- <https://fi.dev.java.net>

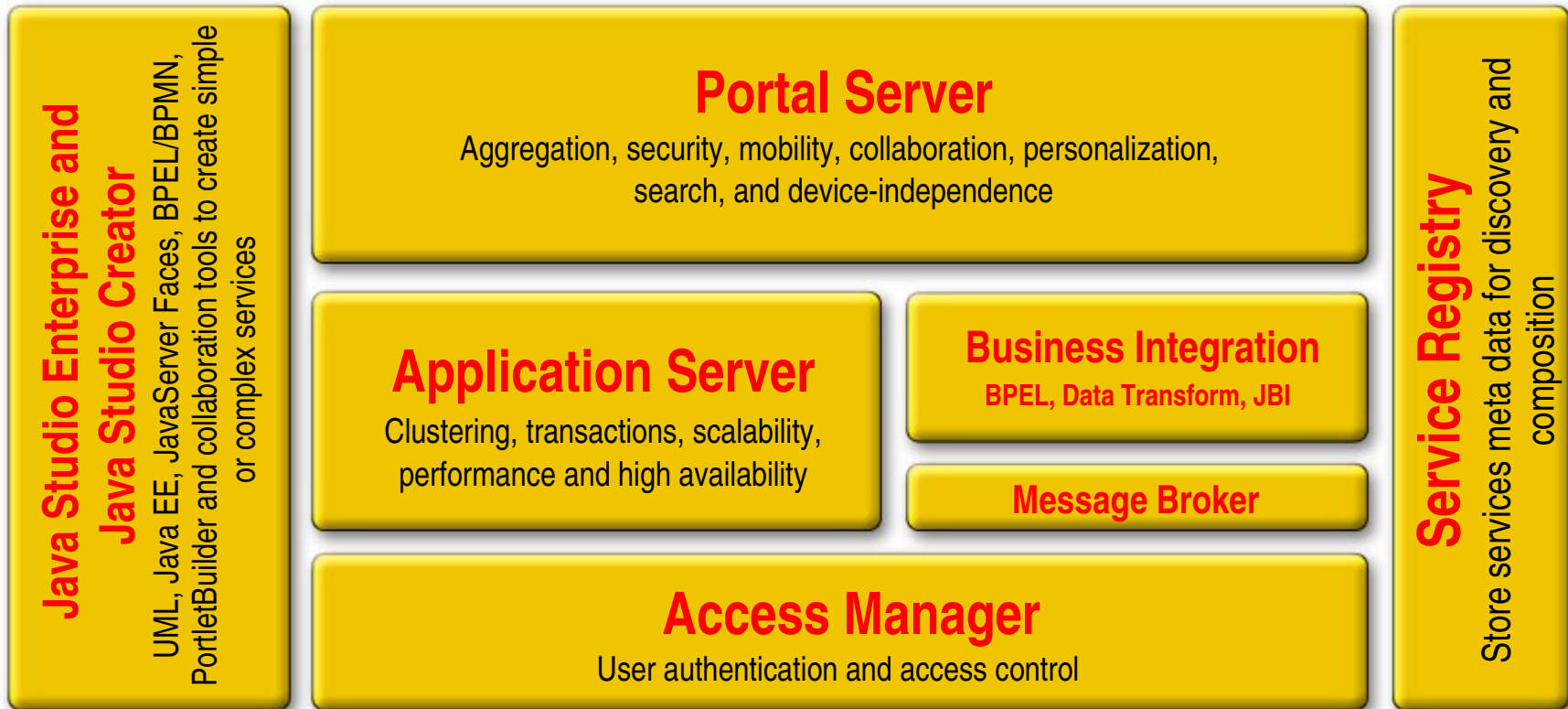


The Future of Java EE Glassfish

- Overview
- Cool/New features in GlassFish
- **Sun Java System Application Server**
- How to get involved
- Resources



Java Enterprise System Application Platform Suite Q4 2005





Sun Java System Application Server

- **Platform Edition**: fast and lightweight
 - > **GlassFish Project**. Basis for Java EE SDK.
 - > Free for development, deployment and (re)distribution
 - > Integrates with NetBeans™, Sun Java Studio Enterprise, Sun Java Studio Creator software, and Eclipse
- **Standard Edition**
 - > Advanced security, out-of-the-box
 - > Self-managing and self-healing
 - > Clustering and multi-machine administration
- **Enterprise Edition supports 99.999% availability**
 - > Very small downtime, no data loss, full recovery



Performance

- Highly **scalable** HTTP path
 - > **Grizzly**: thousands of connections with few threads, all written in Java language (go to TS-3227)
 - > Fast web services using Fast Infoset (go to TS-7187)
- High-performance JMS broker
- Improved Enterprise JavaBeans™ (EJB™) specification performance
 - > Relationship pre-fetching
 - > Read-only EJB specification-based beans
- Optimized XA transactions (LAO)



Sun Java System Application Server Platform Edition

- The first and only **free** application server to publish SPECjAppServer results
- Comparable performance to our SE publication
 - > PE: 298 JOPS on 6 CPUs
 - > SE: 1201 JOPS on 26 CPUs



The Future of Java EE Glassfish

- Overview
- Cool/New features in GlassFish
- Sun Java System Application Server
- **How to get involved**
- Resources



Why Become Part of the GlassFish Project

- Contribute to world class application server
 - > Involved in the implementation of cutting edge Enterprise Java, Web Services, SOA standards, **Grizzly**
 - > **Easy build** environment
- Access to nightly and promoted builds
 - > New Java EE 5 features
 - > Bug fixes
- Interact with a great group of people inside and outside of Sun
- Fully Participative Community



Common Development and Distribution License (CDDL) - pronounced 'cuddle'

- Takes the well known Mozilla Public License (MPL) and makes it reusable without modification
- Clear, consistent, not burdensome for contributors
- As reusable and general as possible
- Meets the definition of [Open Source Definition](#) of [Open Source Initiative](#)
- “...CDDL... Everything is in place for it to work well.”
Linus Torvalds 2/1/05



How to Get Involved

- Join discussions on mailing lists and forums
- **Download** and **tryout** the binary builds
- Submit bugs or request for enhancements
 - > IssueTracker
- Submit a patch to fix a bug or provide an enhancement to issue from
 - > IssueTracker
 - > Web Bugs
- Checkout the GlassFish people page

Building GlassFish

- Build steps – easy, just four !!
 1. Get Apache Maven 1.0.2 from <http://maven.apache.org/>
 2. `cvs -d :pserver:guest@cvs.dev.java.net:/cvs checkout glassfish/bootstrap`
 3. `cd glassfish/bootstrap`
 4. `maven -Dglassfish.os.name="Linux" -Dglassfish.root=".." checkout bootstrap build configure-runtime`
- Full set of instructions on GlassFish site
 - > <https://glassfish.dev.java.net/public/BuildGlassFish.html>
- Instructions to build GlassFish on NetBeans
 - > <http://www.netbeans.org/kb/41/glassfish.html>



The Future of Java EE Glassfish

- Overview
- Cool/New features in GlassFish
- Sun Java System Application Server
- How to get involved
- **Resources**



Resources

- Java.net
 - > <http://java.net>
- Glassfish website
 - > <http://glassfish.dev.java.net>
- Java EE website – Download J2EE SDK here!
 - > <http://java.sun.com/javaee>



Call for Action For GlassFish Project

- Become a user and tell your friends
 - > Download a build
 - > Deploy your application
 - > Learn the new Java EE 5 technologies
 - > Give us feedback
- Become a contributor, we need your help
 - > Bug fixes
 - > Improve build infrastructure and how maven is used
 - > Start with the existing enhancement requests in the Issue Tracker
 - > Use/integrate with your favourite products and share your experience.
- Write tutorials and articles



Questions Please.



Project GlassFish

The Open Source Java EE Application Server

Binod P.G, Sivakumar Thyagarajan

binod.pg@sun.com

sivakumar.thyagarajan@sun.com