

SOFTWARE. HARDWARE. COMPLETE.



ORACLE®

GlassFishで実感! エンタープライズJavaの進化

日本オラクル株式会社

Fusion Middleware事業統括本部 Fusion Middleware第一SC本部
寺田佳央

ORACLE®

Agenda



- ✓ はじめに
- ✓ Java EE 6のご紹介
- ✓ Oracle GlassFish Server 3のご紹介
- ✓ Oracle GlassFish Serverの今後



はじめに

もう一度 Java EE を
見直してみませんか？

- Java EE の開発にこんなイメージを持ってませんか？

- Java EE はXML設定が困難、設定が膨大
- パッケージも面倒 (ear,war,jar...)
- アプリケーションサーバも重い
 - すぐに動作確認ができず開発効率が悪い
 - 再起動を待つのが耐えられない

- Java EE 6 ではかんたん開発がさらに進化しました
- 開発効率が大幅に向上しました
- GlassFish v3はとてもかんたん・軽量です
- GlassFish v3.1は本番環境向けの機能を提供します



Java EE 6の概要

2009年12月10日正式リリース

Java EE 6 のテーマ

- 拡張性
- プロファイル
 - Webプロファイル
 - Enterprise Platform (フルJava EE)
- Pruning
 - 仕様の削減
- かんたん開発
 - 新技術の追加
 - DI, CDI, JAX-RS, Bean Validation
 - 更新された技術
 - JPA2.0, EJB 3.1, JSF 2.0 等

拡張性

- Servlet 3.0
 - アノテーション
 - マルチパート対応
 - 非同期サーブレット
- プラガビリティの向上
 - web-fragments.xml
 - Struts, Spring等
- 管理/設定が容易
 - 設定ファイル(web.xml)の複雑さが軽減

プロファイル

- Java EEの技術を用途毎に分割して提供
 - Java EEのサブセットを提供
- 独自プロファイルの開発が可能
 - 例: 電話会社向けプロファイル
- Java EE 6で最初に提供されるプロファイル
 - Webプロファイル(Webの開発に特化)
 - Enterprise Platform(フルJava EE)

Web プロファイル

- Webアプリケーションの開発に特化した軽量プロファイル
- Webプロファイルに含まれる技術
 - Servlet
 - JSP / EL
 - JSTL
 - JSF
 - Bean Validation
 - EJB Lite
 - JPA
 - JTA
 - DI/CDI
 - Managed Beans
 - Interceptors
 - Common Annotations

Pruning

- 仕様の削減- 2段階プロセス
 - 古く使われなくなったAPIの整理
 - コンポーネントのオプション化
 - 次期バージョン(Java EE 7)で オプション化
 - JAX-RPC(→JAX-WS)
 - EJB Entity Beans(→JPA)
 - JAXR
 - JSR-88

Servlet 3.0

- JSR-315
- 特徴
 - かんたん開発
 - 拡張性
 - マルチパート対応
 - ファイルアップロード
 - 非同期 Servlet のサポート
 - セキュリティ(login/logout処理に対応)

Java SE 5の言語仕様で新たに追加されたアノテーションを使用し宣言的プログラミングモデルを採用。またジェネリクスの利用も可能

ORACLE®

Servlet 3.0

- **Servlet 3.0 で利用可能なアノテーション**
 - @WebServlet
 - - Servletの定義
 - @WebFilter
 - - フィルタの定義
 - @WebListener
 - - リスナの定義
 - @WebInitParam
 - - パラメータの定義
 - @ServletSecurity
 - - セキュリティの制約
 - @MultipartConfig
 - - ファイルアップロード

アノテーションの設定はweb.xmlで上書き設定可能

Servlet 3.0

```
package hello;  
import javax.servlet.annotation.WebServlet;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
  
@WebServlet(name="Hello", urlPatterns={"/Hello"})  
public class Hello extends HttpServlet {  
  
    @Override  
    protected void doGet(HttpServletRequest request,  
                         HttpServletResponse response) {  
        .....  
    }  
}
```

```
<web-app>  
    <servlet>  
        <servlet-name>Hello</servlet-name>  
        <servlet-class>hello.Hello</servlet-class>  
    </servlet>  
    <servlet-mapping>  
        <servlet-name>Hello</servlet-name>  
        <url-pattern>/Hello/* </url-pattern>  
    </servlet-mapping>  
    ...  
</web-app>
```

Servlet 3.0のソースコード

Servlet 2.5まではweb.xmlの編集が必須

ORACLE

EJB 3.1

- JSR 318
- 特徴
 - パッケージの簡略化
 - EJB 3.1 “Lite” の提供
 - ローカルビジネスインターフェースのオプション化
 - 標準化された Global JNDI名
 - Java SEに組み込み可能なEJBコンテナ
 - その他の新機能

EJB 3.1 – パッケージングの簡略化

- Java EE 5 のパッケージング

foo.ear

foo.ear

WEB-INF/web.xml
WEB-INF/classes/
com/acme/FooServlet.class
WEB-INF/classes

foo.jar

com/acme/FooBean.class
com/acme/Foo.class

foo.ear

lib/foo_common.jar

com/acme/Foo.class

foo_web.war

WEB-INF/web.xml
WEB-INF/classes/
com/acme/FooServlet.class

foo_ejb.jar

com/acme/FooBean.class

適切なアーカイブファイル (ear,war,jar) へパッケージ化が必要
面倒なパッケージング/作業負担が大

ORACLE®

EJB 3.1 – パッケージングの簡略化

WEB-INF/classes/com/acme/
FooServlet.class
FooBean.class (EJB)

- かんたんなパッケージング
 - EJBをwarファイルへ含める事が可能
 - WEB-INF/classes:ファイルとして
 - WEB-INF/lib: 分割jarファイルとして
- 今まで同様のパッケージ化も可能
 - ejb-jarファイル
 - 配備記述子はオプション化
 - 記載する必要がある場合別途 WEB-INF/ejb-jar.xmlへ記述可能

EJB 3.1 - “Lite”の提供

Full EJB 3.1機能のサブセットを提供

- Lite
 - ローカルセッションBeans
 - CMT/BMT
 - Declarative Security
 - Interceptors
- Full = Lite +
 - Message-Driven Beans
 - Web Service Endpoint
 - 2.x/3.x Remote view
 - RMI-IIOP Interoperability
 - Timer Service
 - Async method call
 - 2.x Local view
 - CMP/BMP Entity

Bean Validation 1.0

- JSF/JPAと統合
- アノテーションによる制約の表現
 - `@NotNull`
 - `@Size(max=40) String address;`
- カスタム・バリデーター
 - 例: Emailバリデーターをカスタム作成
 - Custom validator classの作成
 - Custom validator methodのBeanへの追加
 - `@Email String recipient;`

Java Persistence API 2.0

- EJBから独立(JSR-317)
- JPA 2.0 = JPA 1.0 + α
 - モデリングの強化
 - JPQL新しい構文の追加
 - Criteria API の提供
 - メタモデルAPIの提供
 - 悲観的ロックの追加
 - バリデーションのサポート
 - 設定オプションの標準化

JPA 2.0 - Criteria API

- JPQLの変わりにプログラミングでクエリーを記載

```
EntityManager em = ...;
CriteriaBuilder cb = em.getCriteriaBuilder();
CriteriaQuery<Person> p = cb.createQuery(Person.class);
Root<Person> person = p.from(Person.class);
p.select(person).where(
    cb.equal(person.get(Person_.name), Taro Yamada));
```

- Metamodel APIとの併用でタイプセーフに
- Person_.nameの有無と型のチェックをコンパイル時に可能
(IDEの強力な型チェック機能でランタイムエラーを抑制)

JavaServer Faces 2.0

- Faceletsの採用
- アノテーション
 - @ManagedBean/@RequestScope/@SessionScope
- faces-config.xmlオプション化
 - ManagedBeanのアノテーション化
 - JSFナビゲーションを改良
 - ボタン/リンク名とXHTMLファイル名のマッチング
 - その他の定義にはfaces-config.xmlが必要
- 標準リソースフォルダ(css/js/images etc)
 - Resourcesフォルダ、warのルートもしくはMETA-INF配下
 - /resources/scripts/, /resources/css/, /resources/img/
- その他: Ajax対応/ブックマーク可能なURL

Dependency Injection

DI 1.0/CDI 1.0

- 新たな@Inject アノテーション
 - @Inject @LoggedIn User user;
- Injection メタモデル
- どんなBeanもInject対象
 - EJB session beans
 - Plain classes with @ManagedBean
 - CDIがモジュール内で見つけたクラス
- デフォルトで無効、有効化する場合は、beans.xmlを配置
 - META-INF/、WEB-INF/に配置



GlassFish v3の概要

2009年12月10日正式リリース

GlassFish v3

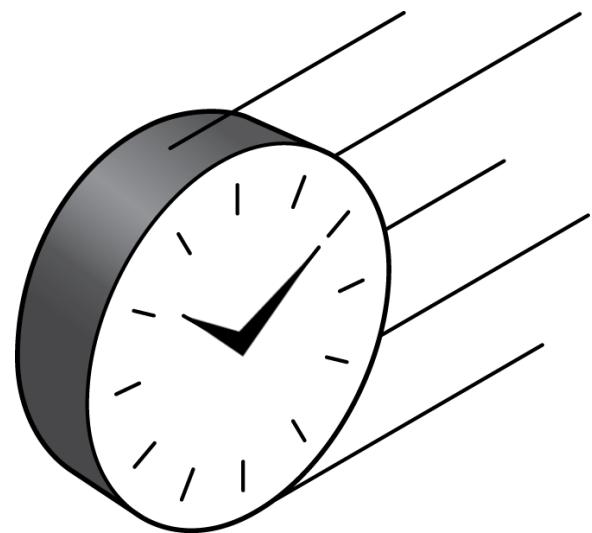
新機能概要

- **Java EE 6 の参照実装**
 - Java EE 6 の仕様に完全準拠
 - 軽量、高速起動
 - かんたん開発／かんたん管理
 - プロファイルに対応
- **先進的なサーバーアーキテクチャ**
 - 非同期 I/O 対応
 - OSGiモジュールサブシステム対応
- **かんたんな操作**
 - unzipによるインストール(Tomcatと同様)
- **Oracle JRockit VM 正式対応**
 - GlassFish v3.0.1より正式サポート

起動時間の劇的な短縮

- **超高速起動**

- 起動時間約 4 秒 (CPU: 2.4GHz Core 2Duo/4GB)
- 軽量コンテナ
- 使われない機能を未初期化
- 開発効率の大幅な向上
- 再起動時間の待ち時間の減少



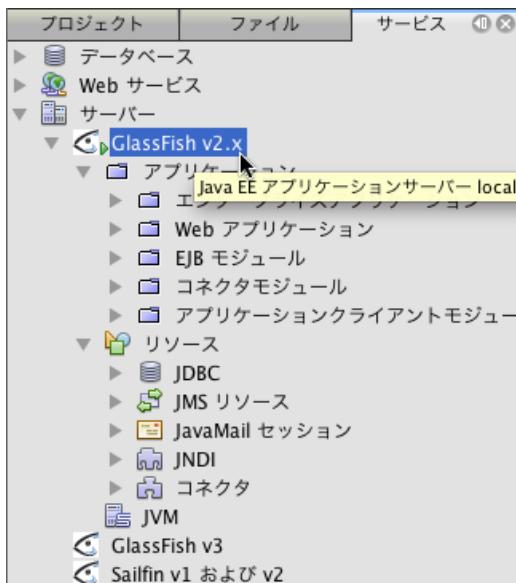
ORACLE®

かんたん開発

統合開発環境連例

- 統合開発環境との連携で開発が容易

- 開発環境からサーバ起動・停止
- Webアプリケーションデバッグ



統合開発環境用プラグイン情報：

<https://glassfishplugins.dev.java.net/>

ORACLE

再配備時におけるセッションの持続

- 再配備時にセッションを保存
 - 開発環境で有効
 - コマンドオプション : -keepSessions=true
- 統合開発環境と連携
 - Eclipse
 - NetBeans



ORACLE®

かんたんな管理機能

- 視覚的に理解が容易な管理コンソール
 - 日本語化可能(更新ツールより)
 - 軽快な操作
 - 管理用コマンドを用意
 - 管理用 REST APIを提供

The screenshot shows the GlassFish v3 Administration Console. On the left, the navigation tree includes 'リソース' (Resources) with 'JDBC' and '接続プール' (Connection Pools), and '構成' (Configuration) with 'JVM 設定' (JVM Settings). The main panel shows the 'JDBC 接続プールを編集' (Edit JDBC Connection Pool) configuration for 'DerbyPool'. It lists '一般設定' (General Settings) with 'JNDI名' (Name) as 'DerbyPool', 'リソースタイプ' (Resource Type) as 'javax.sql.DataSource', and 'データソースクラス名' (Data Source Class Name) as 'org.apache.derby.jdbc.ClientDataSource'. The 'Ping' section has a checked '有効' (Enabled) checkbox. The right panel shows the '利用可能なアドオン' (Available Add-ons) page, listing various components like 'glassfish-web-3.1' and 'glassfish-jersey-2.0' with their details and download links.

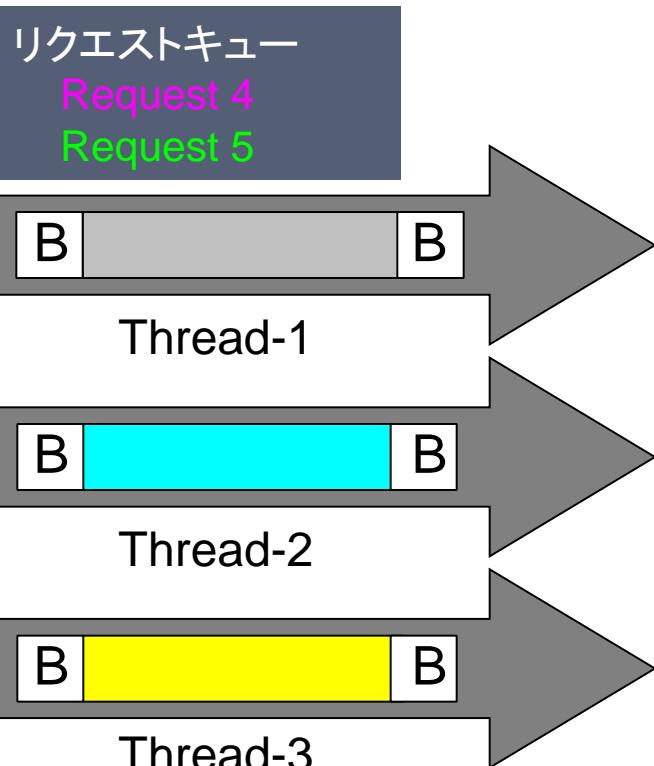
コンポーネント	カテゴリ	バージョン	インストールサイズ	ソース
glassfish-web-3.1	Application Servers	3.0.1-1	57KB	dev.glassfish.org
glassfish-jersey-2.0	Application Servers	3.0.1-1	57KB	dev.glassfish.org
java-gems	Scoring	2.3.6.1.0	38MB	comtib.glassfish.org
python-runtime	Scoring	2.5.11.0	37MB	comtib.glassfish.org
mq-docs	Message Oriented Middleware	4.4.17.2	9MB	mqdocs.glassfish.org
python-container	Scoring	0.5.6-1.1	79KB	comtib.glassfish.org
hibernate	Frameworks	3.5.6.0-2	5MB	comtib.glassfish.org
grails	Scoring	1.1.2-1.0	69MB	comtib.glassfish.org
java-db-doc	Databases and Tools	10.5.3.0-1	15MB	releases.glassfish.org
sun-javase-engine	JBI Component	3.0.11	245KB	dev.glassfish.org
sun-javase-engine	JBI Component	3.0.14	245KB	dev.glassfish.org
sun-javase-engine	Databases and Tools	10.5.3.0-1	104MB	releases.glassfish.org

アップデートツールの統合

ORACLE®

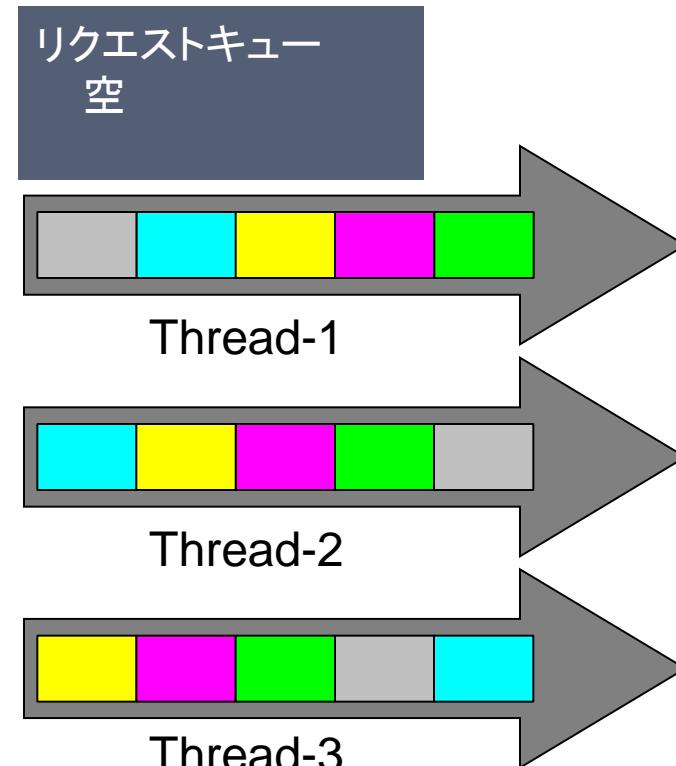
非同期 I/O 対応

同期 I/O(Blocking)



1処理に1スレッドを占有

非同期 I/O(Non-Blocking)

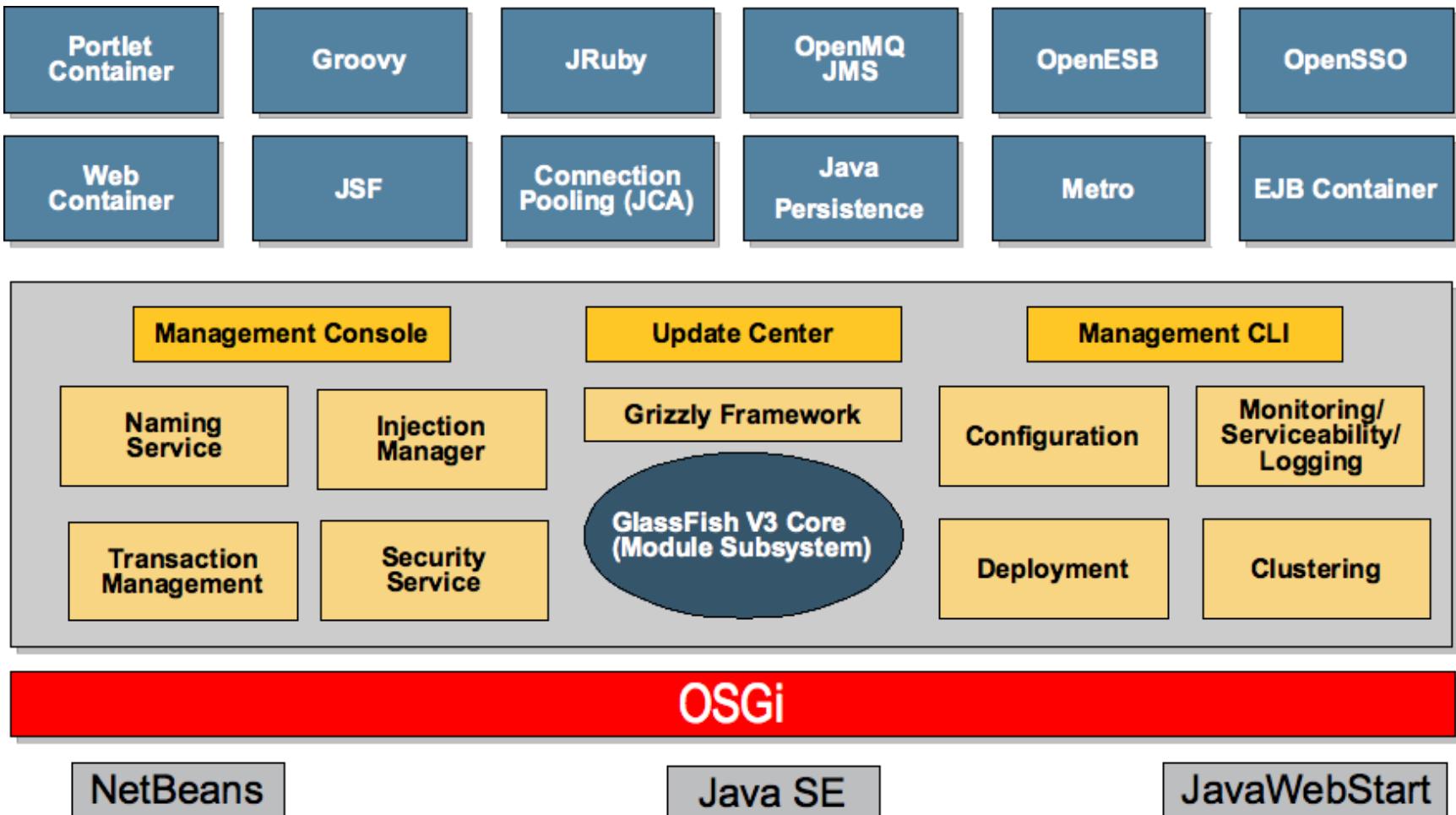


スレッド内で処理を分割

ORACLE®

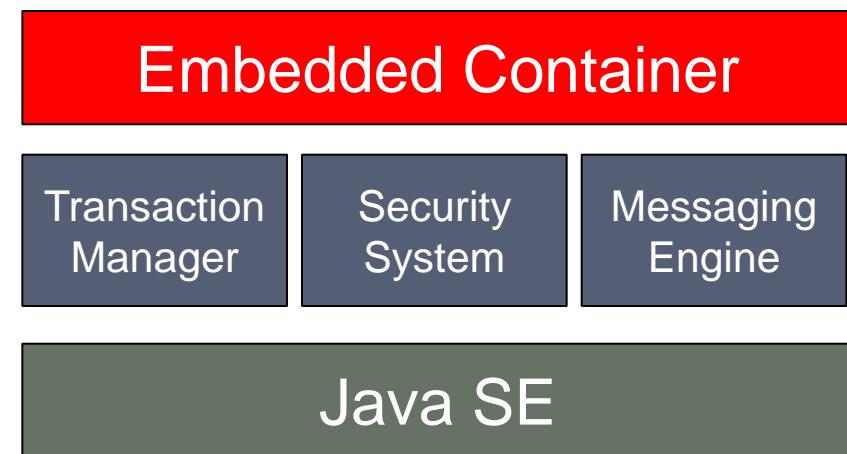
GlassFishのOSGi対応

- GlassFish v3 のアーキテクチャ



GlassFish組み込み可能コンテナ

- GlassFish Embedded Server API を提供
 - EJB 3.1 Embeddable API
 - Java SEプログラム内から GlassFishを起動
- Maven Plug-inの提供
 - Apache Mavenとの連携
 - JUnit による容易なテスト
- デスクトップアプリケーションで EJB 機能を利用可能



組み込みコンテナを利用したJUnitテストコード

```
@Test
public void testSayHello() {
    Map p = new HashMap();
    p.put("org.glassfish.ejb.embedded.glassfish.instance.root",
        "/Applications/GlassFish/glassfishv3-webprofile/glassfish/
        domains/domain1");
EJBContainer container = EJBContainer.createEJBContainer(p);
    try{
        Hello hello = (Hello)container.getContext().lookup
            ("java:global/classes/Hello");
        System.out.println(hello.sayHello());
    }catch(Exception e){
        e.printStackTrace();
    }
}
```



GlassFishの今後

製品のロードマップ

2010年末
Ver 3.1

2011年
Ver 3.2

未定
Ver 4

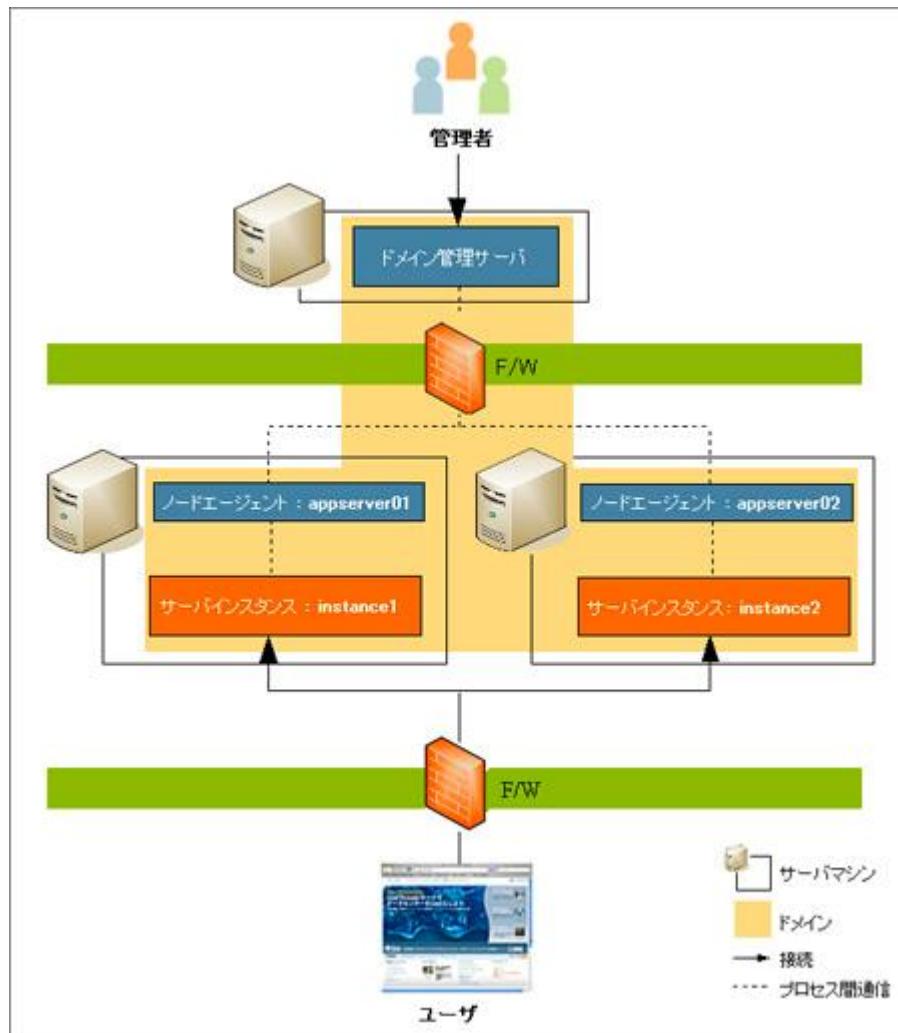
- **GlassFish v3.1 – 2010**
 - クラスタ、中央集中管理機能
 - 高可用性／セッションリプリケーション
 - Coherenceのサポート等
- **GlassFish v3.2 – 2011**
 - クラスタ機能の改良
 - 仮想化サポート
 - Java EE 7のEarly Access 機能を提供
- **GlassFish v4**
 - Java EE 7対応

ORACLE®

GlassFish v3.1の新機能

- クラスタ機能
 - クラスタ対応
 - SSH プロビジョニング(Node Agentの廃止)
 - アプリケーションのバージョニング対応
 - ローリングアップグレード機能の提供
- ドメイン管理サーバの高可用性
- HTML5 WebSocket 対応
- 管理・監視機能の強化
 - DTraceを使用したモニタ
- WebLogic との互換性を提供
 - WebLogicのデプロイメント記述子をサポート

GlassFish v2.1.1のクラスタ機能

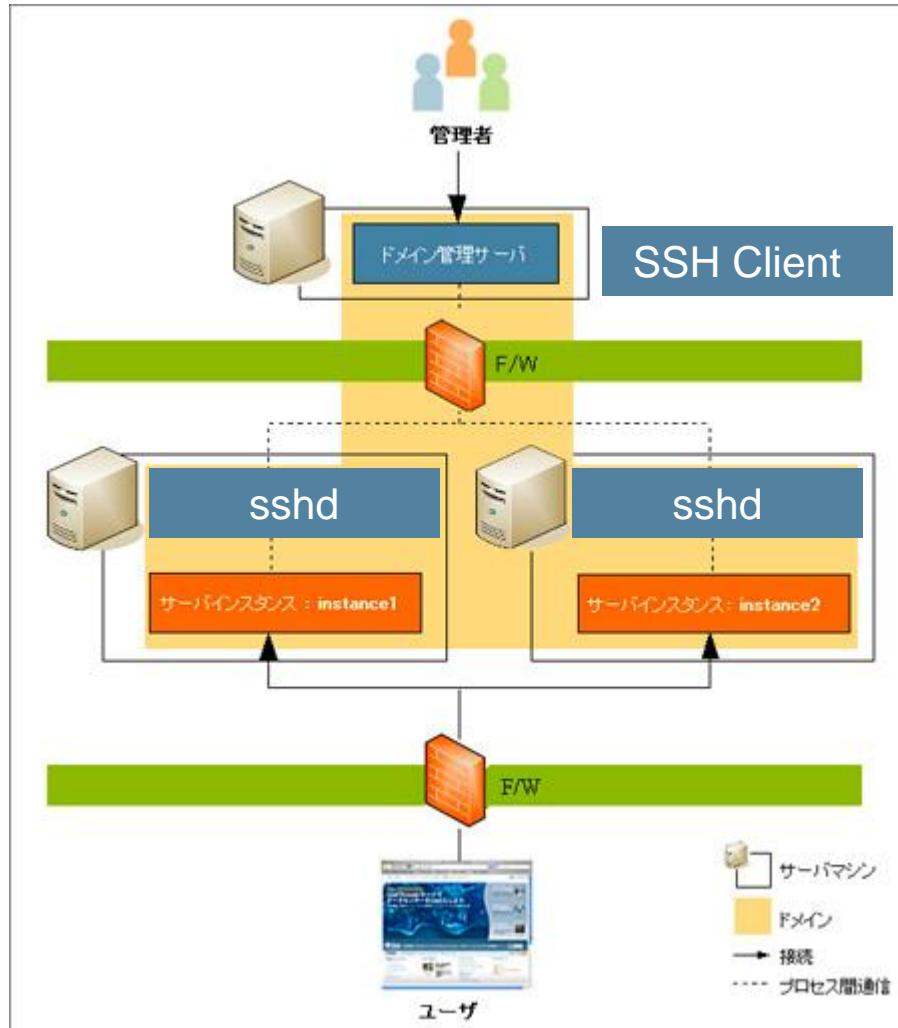


・ノードエージェント方式

- 各物理ノードにGlassFishをインストール
- 各物理ノードにノードエージェントを作成
- ノードエージェントの起動・停止は各マシン上で実施
- ドメイン管理サーバから各ノードエージェントのインスタンスを管理

ORACLE®

GlassFish v3.1のクラスタ新機能



- **SSH プロビジョニング方式**
 - ドメイン管理サーバ内に SSH Client ライブラリを統合
 - Hudson(Trilead-ssh2)の成果物を利用
 - 各物理ノードでsshdを起動
 - システムの初期化とドメイン管理サーバへの登録
- **将来的な実装予定**
 - 各マシンへのインストールもssh 経由で
 - Cloud環境でプロビジョニング
 - Auto Scale対応等

GlassFishの特徴

- 無償ダウンロード、7x24時間サポート
- Java EE 標準の参照実装
- 大規模コミュニティ(100万ダウンロード／月)
- Java EE 6 にいち早く対応(Web Profile対応)
- 軽量、モジュール化(OSGi対応)
- 高速起動、(5秒以下)
- オープンソースアプリケーションサーバで最速(SPECjAppServer ベンチマーク)
- 再配備時のセッション保持
- Microsoft .NET 3.0/3.5との相互運用性
- 各種Webサービス対応
- スクリプト言語(JRuby/Rails,Jython,Django,Scala/Lift, PHP, Server-side Java Script, Groovy/Grails)のネイティブ対応
- GlassFish Enterprise Managerを通じたモニタ、拡張管理

GlassFishの参考情報

- Oracle GlassFish Server
 - <http://www.oracle.com/technology/global/jp/products/glassfish>
- Oracle GlassFish Server 全ドキュメント(英語)
 - <http://docs.sun.com/app/docs/coll/1343.13?l=ja>
- Sun GlassFish Enterprise Server v3 管理ガイド(日本語)
 - <http://docs.sun.com/app/docs/doc/821-1299>
- GlassFish Community
 - <https://glassfish.dev.java.net>
- GlassFish Wiki
 - <http://wiki.glassfish.java.net/>
- ブログ
 - <http://blogs.sun.com/theaquarium/>
 - <http://yoshio3.com/>

まとめ

- GlassFish v3 は先進的なアプリケーションサーバ
 - Java EE 6 の参照実装
 - 軽量／高速起動
 - インストールも容易
 - Java EE 以外のスクリプト言語にも対応
 - Cometアプリケーションの動作環境として最適
- Java EE 6 でさらに進んだかんたん開発
 - XML設定ファイル編集の軽減
 - EJB組み込みコンテナで単体テストが容易
 - 統合開発環境連携でJPAのDBモデリングも容易
- GlassFish v3.1でエンタープライズ機能を提供

本ドキュメントは、弊社の一般的な製品の方向性に関する概要を説明するものです。また、情報提供を唯一の目的とするものであり、いかなる契約にも組み込むことはできません。以下の事項は、マテリアルやコード、機能を提供することをコミットメント(確約)するものではないため、購買決定を行う際の判断材料になさらないで下さい。オラクル製品に関して記載されている機能の開発、リリースおよび時期については、弊社の裁量により決定されます。

Oracle、PeopleSoft、JD Edwards、及びSiebelは、米国オラクル・コーポレーション及びその子会社、関連会社の登録商標です。その他の名称はそれぞれの会社の商標の可能性があります。

ORACLE®

ORACLE®

Copyright© 2010, Oracle. All rights reserved.

44