JBoss AS 7 简介

Jeff Zhang (Senior Engineer, JBoss AS)

jeff.zhang AT jboss.org

日程

- 历史
- 为什么开始开发 AS7
- 基础知识
- 结构
- 组件
- 举例

JBoss AS 历史

业界市场份额最大的开源 JavaEE 应用服务器

AS 4

- JavaEE 1.4, EJB 2.1
- 最为成熟的版本，直到如今依然很多人使用
- JMX作为内核

AS 5

- JavaEE 5, EJB 3
- 重新设计的MicroContaner作为内核(Adrian Brock), 开发历时时间很长
- EAP 5.1 基于 AS5, 会继续维护

JBoss AS 历史

AS 6

- JavaEE 6, EJB 3.1
- 在AS5基础之上的改进版，期望能大幅度提高启动时间，以及对domain支持
- 如今 AS6 CR1 tag，预计年内Release(Then die)

AS 7

- JavaEE 6
- 重新设计内核，原有的MC的相关组件几乎全部抛弃
- AS作为主容器，其他子系统作为extension的思路保持不变

为什么开始开发 **AS7**

- 原有AS非常臃肿 >170M
- 启动时间非常缓慢 (TODO 有一个业界应用服务器比较图)
- 部署时间缓慢，受到MC相关组件设计的限制
- Domain特性缺失
- 扩展性受限，复杂的内核机制
- 新人新思路

资源

People

- David M Lloyd (MSC, JBoss-modules, 架构师)
- Jason Greene (Team leader， 领导)
- Brian Stansberry (Domain, HA等高端特性负责)

URL

- source code (https://github.com/jbossas/jboss-as)
- discussing forum (http://community.jboss.org/community/jbossas/dev/jboss_as7_development?view=all)
- download (http://sourceforge.net/projects/jboss/files/)

基础知识

Java相关

- Stax (替换原有JBossXB)
- Classloader
- OO 面向对象

工具

- Maven 3 编译，包管理
- Git, github.com

原有AS相关知识

- 子系统，系统组件，JavaEE相关规范实现
- 部署模型,(借鉴原有的Deployer框架，代码重写，思路基本一致)
- Sar, Thread, LogSystem

编译和启动

编译

```
* git clone https://github.com/jbossas/jboss-as.git
  (如果参与项目，需要在自己的帐号下clone项目，然后编译)
* cd jboss-as
* ./build.sh install
```

启动

编译或者下载解包后

```
> cd $AS_HOME/bin
> ./standalone.sh
```

会看到

```
10:15:28,382 INFO  [modules] JBoss Modules version 1.0.0.Beta11
10:15:31,217 INFO  [server] Starting standalone server
10:15:31,447 INFO  [server] Activating core services
...
10:15:35,366 INFO  ... Starting JCA Subsystem (JBoss IronJacamar 1.0.0.Beta3)
10:15:35,408 INFO  ... Started FileSystemDeploymentService for directory ...
10:15:35,411 INFO  ... JBoss AS 7.0.0.Alpha2 "Halloween" started in 4028ms.
```

结构

两种启动模式

- Standalone 独立进程
- Domain 域模式，多个JVM进程连接成一个域

目录结构

- bin 启动脚本
- docs 文档，包含范例，schema, license
- modules 按照一定目录结构组成，classloader可以按照组件化的方式加载其中的模块。
- domain 域模式的配置文件，数据，日志等
- standalone 独立模式的配置文件，数据，日志等

举例， 独立模式配置文件

standalone.xml

扩展模块

```
<extensions>
    <extension module="org.jboss.as.connector"/>
    <extension module="org.jboss.as.jmx"/>
    <extension module="org.jboss.as.logging"/>
</extensions>
```

子系统

```
<extensions>
    <subsystem xmlns="urn:jboss:domain:logging:1.0">
        <console-handler name="CONSOLE">
            <level name="INFO"/>
            <formatter>
                <pattern-formatter pattern="%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n"/>
            </formatter>
        </console-handler>
</extensions>
```

目前(1.0.0.Alpha1)的子系统

- connector (JCA, Ironjacamar)
- jmx
- logging (JBoss Logging)
- managed-beans
- messaging (JMS, Hornetq)
- naming
- osgi (JBoss Osgi)
- remoting (JBoss remoting)
- sar
- threads (JBoss thread)
- transcations (JBoss Transaction)
- web (JBossWeb)

很快还会加入

- ejb (ear)
- security (PickBox)
- webservice (JBossWS)
- cdi (Weld)
- rs (Resteasy)
- …

核心组件

MSC

- “微内核”组件
- 纯Java，不用xml中bean定义和Annotation
- 目标，小而快
- 注入，依赖性，生命期管理

JBoss-Modules

- 管理扩展组件
- classloader管理器
- 借鉴Osgi

核心接口和概念

核心接口和类

- org.jboss.msc.service.Service 服务，定义生命期
- org.jboss.msc.service.BatchBuilder 构造服务和依赖性
- org.jboss.msc.service.ServiceController 服务控制器
- org.jboss.msc.value.InjectedValue 值注入包装器，以及相关的类，构成值注入

核心概念和关键类

- Extension 扩展点 META-INF/services/org.jboss.as.Extension
- Parser 解析器 解析xml配置文件 (XMLElementReader)
- SubsystemAdd 添加子系统 (AbstractSubsystemAdd)
- SubsystemElement/EnvironmentElement/Namepace/Element/Attribute 定义schema解析出来的元素
- Services 扩展的自定义服务 (org.jboss.msc.service.Service)
- Processor 组件内容解析和处理，类似原来的Deployer (DeploymentUnitProcessor)
- DeploymentChainSelector 部署类型选择器

举例一，Transcation

- TransactionsExtension initialize方法，注册解析器
- TransactionSubsystemElementParser readElement方法，解析Transaction相关属性
- TransactionSubsystemAdd applyUpdate方法,通过BatchBuilder加入服务和依赖性
- TransactionManagerService start/stop方法

举例二, Sar

- 启动服务过程和上页类似
- SarSubsystemAdd applyUpdate方法中，加入处理器，

```
addDeploymentProcessor(batchBuilder, new ServiceDeploymentParsingProcessor(),

addDeploymentProcessor(batchBuilder, new ParsedServiceDeploymentProcessor(),
```

- ServiceDeploymentParsingProcessor processDeployment方法，处理jar/sar，读入META-INF/jboss-service.xml文件
- ParsedServiceDeploymentProcessor processDeployment方法，加载对应类，启动服务

休息...

End