体验 GlassFish 的特色功能

蒋健， 王昱
Sun Microsystems, Inc.

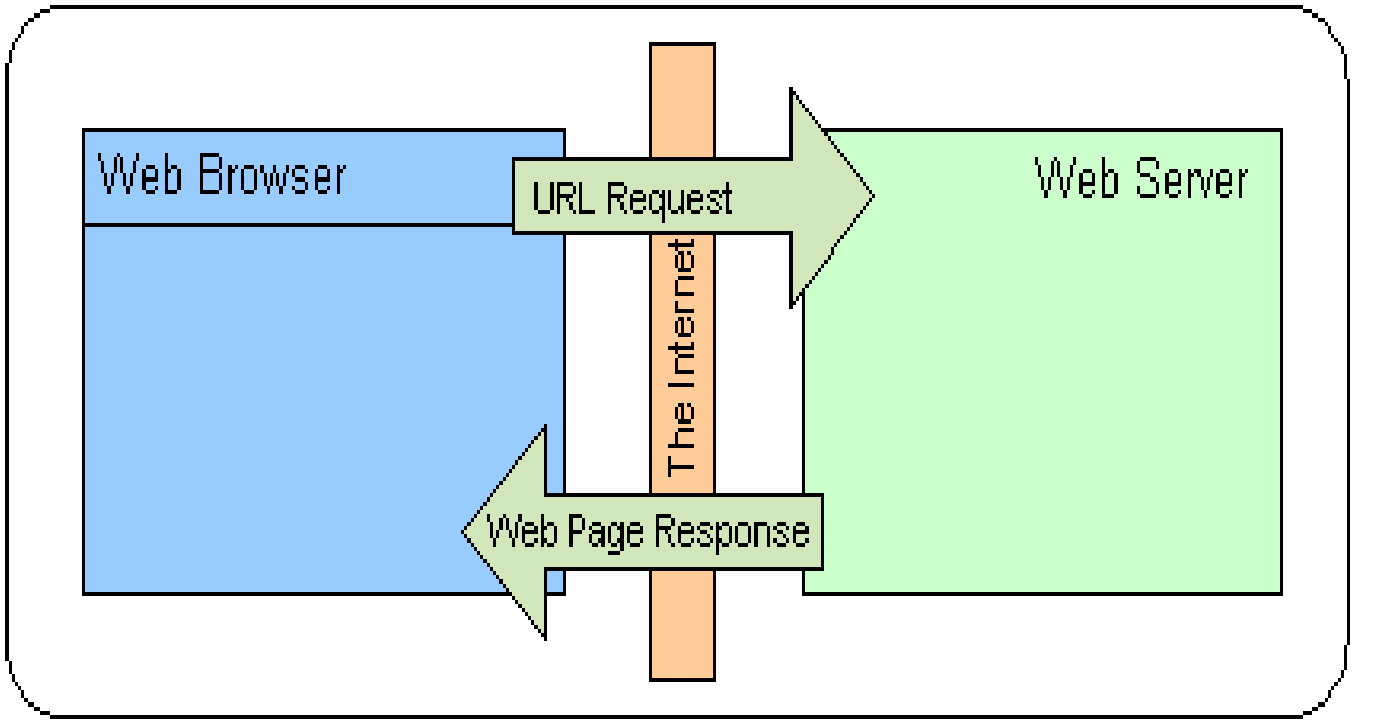
内容介绍

除了实现了 *JavaEE5* 的标准以外，*GlassFish* 还拥有许多非常出众的特点，这些特点在构建未来的企业应用非常有用，特别是对 *Web2.0* 的支持。在这个讲座中，我们来一一了解这些有趣的特点

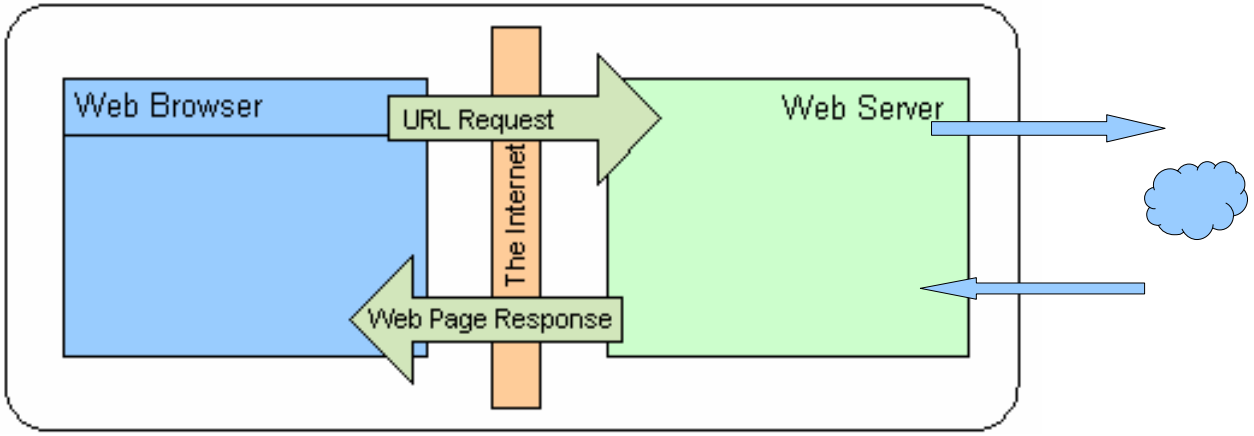
GlassFish 特色功能

- 异步请求处理
- 服务器推送技术
- 虚拟资源管理
- AMX
- 自管理
- CallFlow

HTTP同步请求处理


Web Browser
URL Request
Web Server
The Internet
Web Page Response

HTTP同步请求有时候满足不了...



- 请求处理需要调用外部服务
- 需要人工干预（经理批准）
- 例如：邮件显示

HTTP同步请求有时候满足不了...

- 通常的解决方案为 Polling
- 将业务逻辑拆分为两步
  - 业务执行
  - 状态查询
- Ajax 的应用（自动 polling ）
- Polling 的缺点
  - 白白消耗网络资源
  - 白白消耗 CPU 资源

GlassFish异步请求处理

- Polling 的解决方案的扩展性不好
- GlassFish通过NIO来实现ARP层
- ARP 允许暂时停止一个请求的处理，在某些条件满足后，重新执行
- 保持HTTP连接不断，但是并不是每个连接都会占用一个工作线程，来保证异步处理的性能和扩展性，这一点很重要。
- 例子：邮件显示
  - http://developers.sun.com/learning/javaoneonline/j1lab.jsp?lab=LAB-3360&yr=2007&track=3

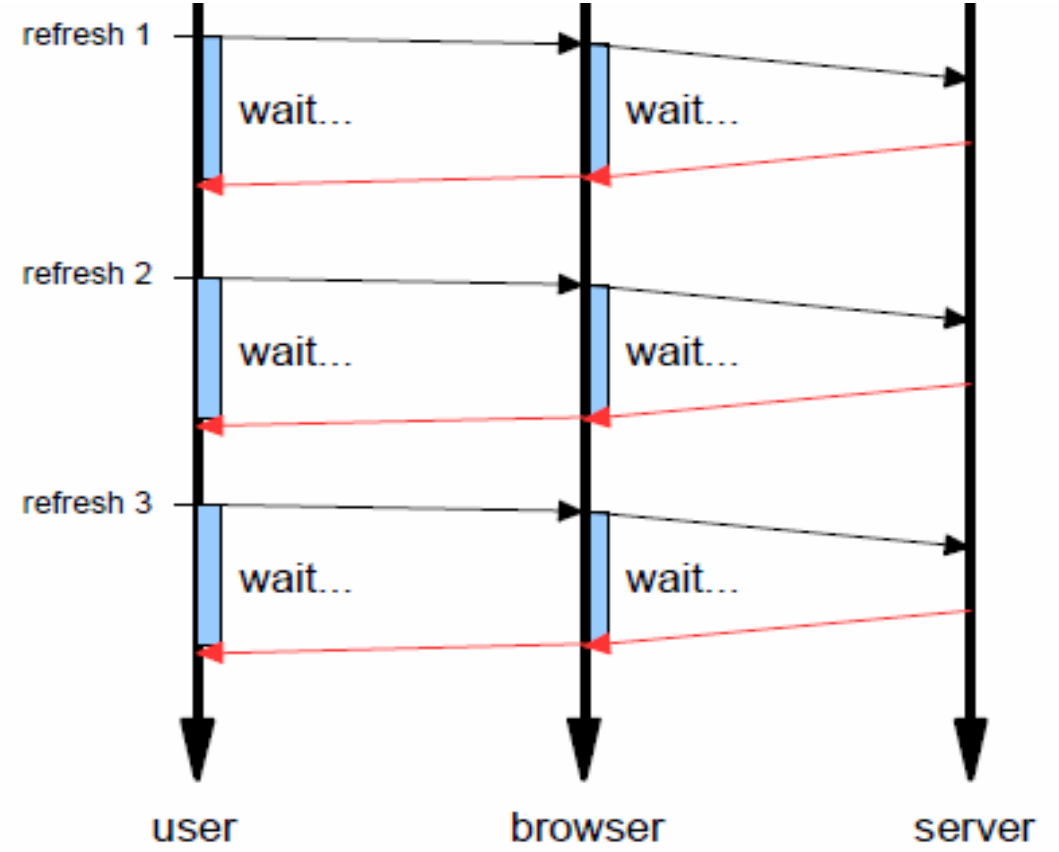
GlassFish 特色功能

- 异步请求处理
- 服务器推送技术
- 虚拟资源管理
- AMX
- 自管理
- CallFlow

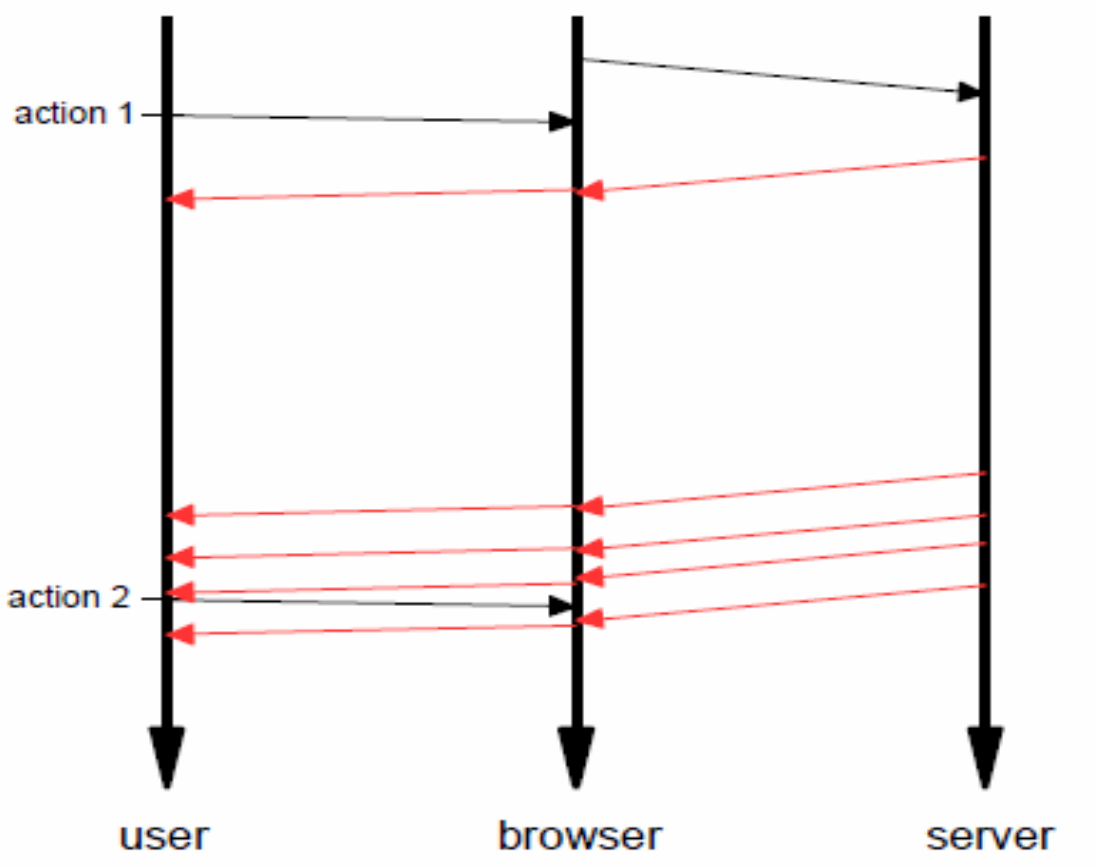
服务器推送技术 (Comet)

- B/S 应用逐步替代 C/S 应用
- Ajax 正在吞噬桌面应用的市场
  - 文字处理器
  - 邮件客户端
  - 相片管理及图形编辑
- 除了和硬件相关的 CAD 软件以及大型游戏
- Web 应用（ Ajax 应用）的一个致命的缺点：服务器端不能主动发消息
- 影响关键应用的响应能力，延迟时间长（股票系统）

传统 Web 的请求响应序列



Comet 请求的响应序列



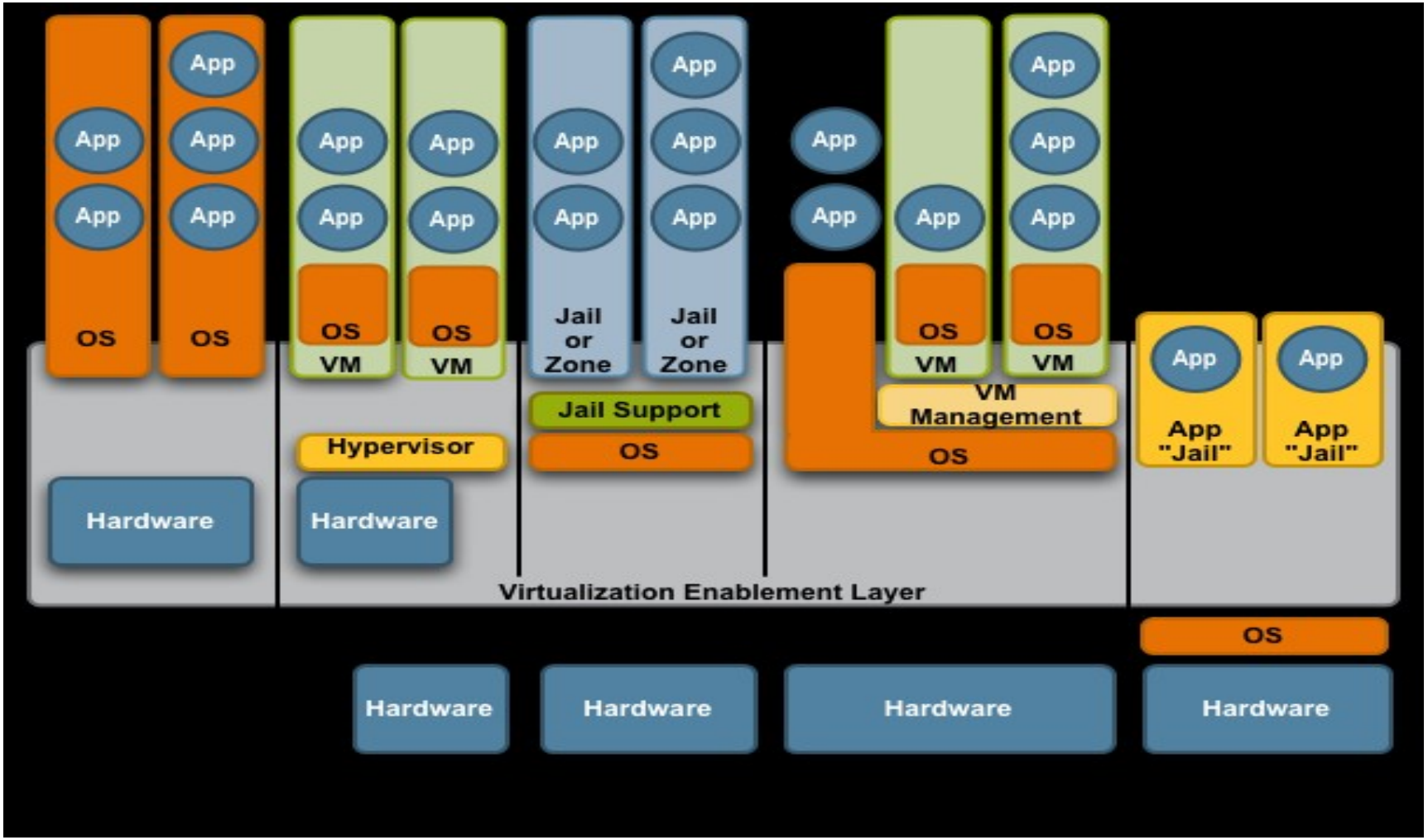
GlassFish数据推送技术

- 服务器端主动发送消息
- 没有使用Polling方法
- 使用了长时间保留的HTTP连接，减少延迟
- 基于 ARP（NIO）的高性能实现.
- 非常容易使用和扩展的API
- 兼容Cometd协议（http://cometd.com)

GlassFish 特色功能

- 异步请求处理
- 服务器推送技术
- 虚拟资源管理
- AMX
- 自管理
- CallFlow

资源虚拟化管理


App
OS
VM
Jail or Zone
Jail Support
Hypervisor
VM Management
App "Jail"
Hardware
Virtualization Enablement Layer

GlassFish的资源管理(RCM)

- GlassFish的 RCM与平台无关
- 应用层的资源分配的策略
- 比底层的虚拟技术更大的灵活性
- GlassFish目前支持两种规则：
  - 保留一定百分比的内存资源
  - 保留一定百分比的处理线程资源

GlassFish 特色功能

- 异步请求处理
- 服务器推送技术
- 虚拟资源管理
- AMX
- 自管理
- CallFlow

JMX — Java 管理扩展

- JMX
  - 管理能被 Java 抽象的各类应用、系统和网络
  - 扩展相应的管理和监控方案，并使之模式化和标准化
- 应用领域
  - 配置查询及更改，各类统计，状态通知
- 好处
  - 条形码，脚手架

JMX 是 GlassFish 管理架构的基础

- 模块化的开发和实现
  - 命令行工具的支持
  - 管理控制台的支持
- AMX 是 GlassFish 对 JMX 的扩展
  - MBean 的门面
  - DCP(Dynamic Client-side Proxy)：映射到客户端的 MBean 代理
  - Dotted Name 支持

特性展示—查看 GlassFish 管理组件

- 通过命令行工具 asadmin 的子命令 get/list 查看
- 通过管理控制台查看
- 通过第三方 JMX 管理应用 JConsole
- 通过 JMX 代码查看
- 通过 AMX 代码查看

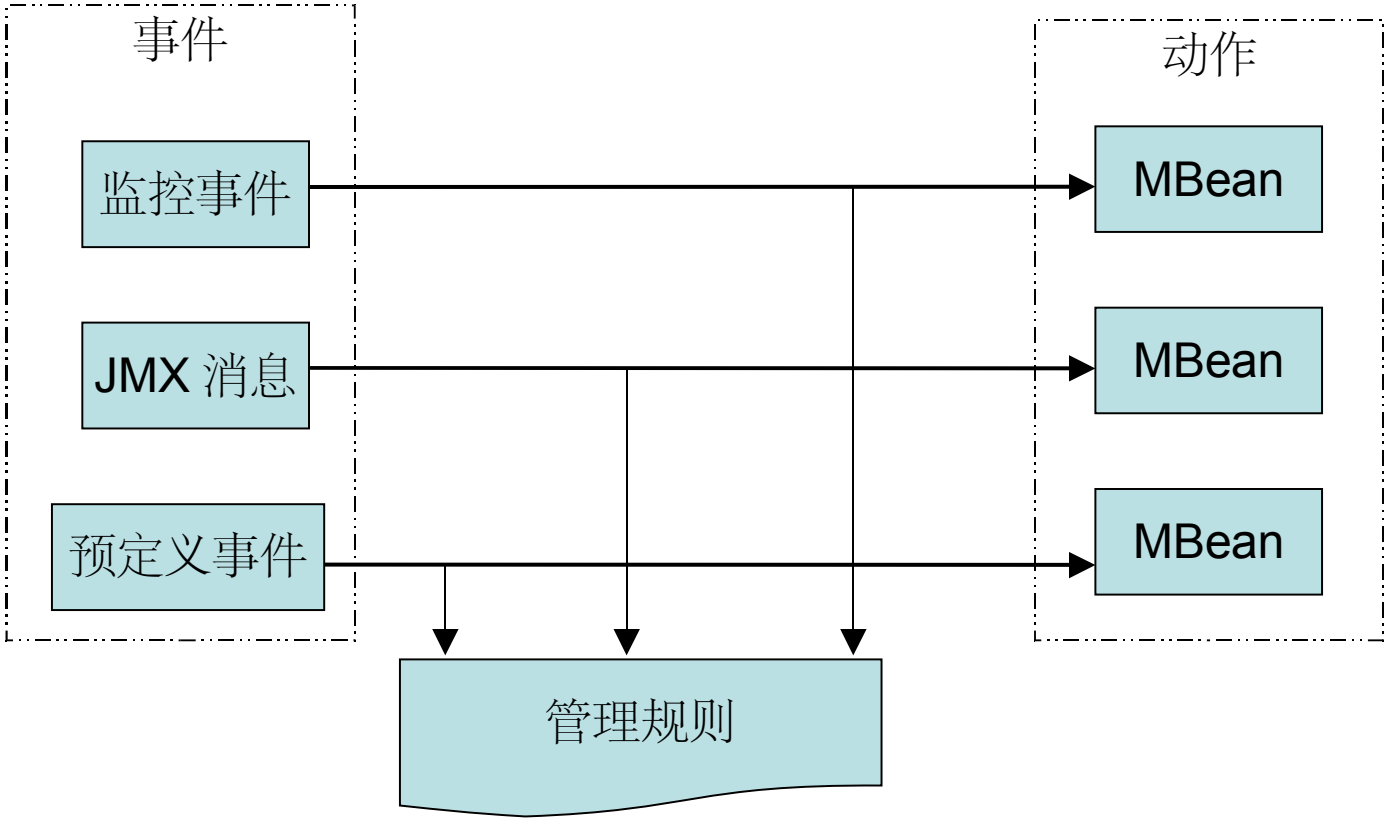
GlassFish 特色功能

- 异步请求处理
- 服务器推送技术
- 虚拟资源管理
- AMX
- 自管理
- CallFlow

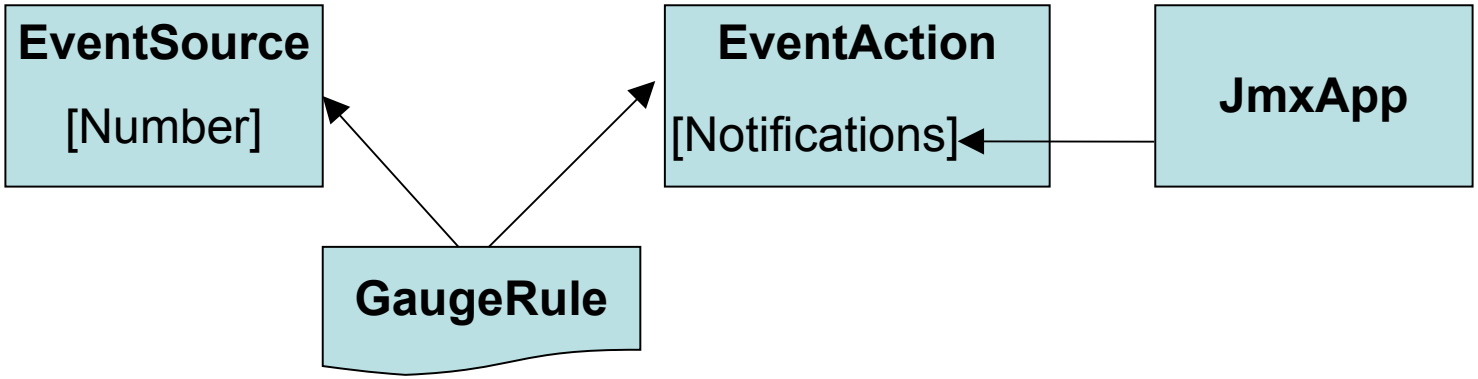
自管理

- 自管理的核心是管理规则的定制和运行
  - 用户自行定义的描述在某种事件触发时完成何种处理的规则
  - 基于 JMX 的事件机制
  - 用于自调整，自配置，自恢复
- 好处
  - 高度可定制化
  - 可以更深入地参与系统的管理

管理规则＝事件＋动作



特性展示—监控数值的自管理应用



- 部署 MBean’EventSource’
- 部署 MBean ‘EventAction’
- 创建数值型监控器的管理规则 ‘GaugeRule’
- 运行客户端监控程序 ‘JmxApp’
- 通过 JConsole 触发管理规则

GlassFish 特色功能

- 异步请求处理
- 服务器推送技术
- 虚拟资源管理
- AMX
- 自管理
- CallFlow

CallFlow —内置的性能探查器

- 收集应用程序运行时的信息，供性能调优和程序调试
- 特点
  - 对系统性能的影响轻微
  - 可以监控到其它 Profiler 无法访问到的信息
  - 可以监控特定的客户端和用户
  - 提供开发接口，方便数据挖掘

特性展示—通过 CallFlow 监测 Web 服务

- 信息收集
  - 各类容器中消耗的时间
  - 被调用的方法及其响应堆栈
  - 关联的应用程序名和模块名
  - 异常出处
- 图形化的结果显示
- 数据查看和过滤

GlassFish 的特别源自开源

- 开发模式
  - 大教堂式的开发 转向集市式开发
  - „“ 如果有足够多的眼睛，所有错误将显而易见
- 开源社区
  - 成为开发载体和平台
  - 公开源代码的同时，也公开了整个开发过程
- 开源生态
  - 提供契机去产业链的上游去获取资源

资源

- 书《GlassFish —开源的Java EE应用服务器》
- JavaOne 2007上的动手实验室 3360 "Taste Special Features of Glassfish"