

JMX 动态管理服务的研究与设计*

孙明** 周明天 詹瑾瑜

(电子科技大学计算机科学与工程学院 成都 610054)

【摘要】 在 JMX 规范 1.0 的基础上, 分析了 JMX 的技术背景及其体系结构, 提出了一种 MBean 对象的实现方法和 Java 管理服务核心模块的设计模式。讨论了远程管理通信接口以及基于 Java 事件模型 Notification 机制的设计, 并对标准管理服务器进行相关扩展, 实现持久化对象管理和远程对象管理。

关键词 Java 管理扩展; 管理服务器; 会话; 通知机制

中图分类号 TP393

Research and Design of Dynamic Management Service Based on JMX

Sun Ming Zhou Mingtian Zhan Jinyu

(College of Computer Science and Engineering, UEST of China Chendu 610054)

Abstract Conforming to the JMX specification 1.0, this paper analyzes the technology background and the architecture of JMX, and discusses the implementation method of the Mbean object and the design of the core of dynamic management service. In particular, the paper also discusses the design of remote management communication interface and Notification mechanism based on Java event model, and extends the standard MBeanServer to be a persistent and remote management service.

Key words java management extension; mbean server; session; notification

Java Management Extensions(JMX, Java 管理扩展)是 SUN 基于 Java 技术提出的一种管理策略^[1], 它将 Java 语言的动态性、灵活性以及可移动性引入了管理服务的范畴中。使得 JMX 有构建第三代管理解决方案的基础^[2], 因为第三代管理技术在本质上是动态, 能够满足在各种各样的网络和平台上的服务驱动的管理请求。JMX 的提出保持了 Java 平台在信息技术领域中的领先地位, 满足动态管理的市场请求, 为管理系统的设计者和开发者提供方便的工具。通过设计, 这种新的标准能够很好的适应以前的系统, 实现新的管理方案, 同时保持对所有的工业应用开放。本文主要论述管理服务系统中 Managed Bean(Mbean)对象定义以及 JMX Agent 动态管理核心 MBeanServer 和远程管理通信接口的设计。

1 JMX 管理服务体系结构

JMX 是一种致力于解决分布式系统管理问题的应用编程接口, 目前 JMX 管理体系结构基本可以划分为以下三个层次:

1) Instrumentation level 提出了实现 JMX 可管理资源的规范。这些资源可以是 Java 应用、服务

2002年6月24日收稿

* 信息产业部预研基金资助项目; 四川省重大科技攻关基金资助项目, 编号: SG95.17.1

** 男 24岁 博士生

实现、设施等，它们是通过一个或多个 MBean 来实现。

2) Agent Level JMX Agent 是管理服务的实体，运行在 JVM 上起着连接管理应用和 MBean 的桥梁作用。它直接控制被管理资源，使其可用于远程管理。

3) Manager Level 提供了管理组件的实现，它可以作为管理者或管理代理通过 Agent 层操纵分布式管理服务。

此外，JMX 还通过 Additional Management Protocol API(附加管理协议 API)，为目前大多数流行的管理协议提供了扩展接口，目的在于管理系统开发通用性以及提供一套整合现有管理技术的解决方案，其管理服务的体系结构如图1所示。

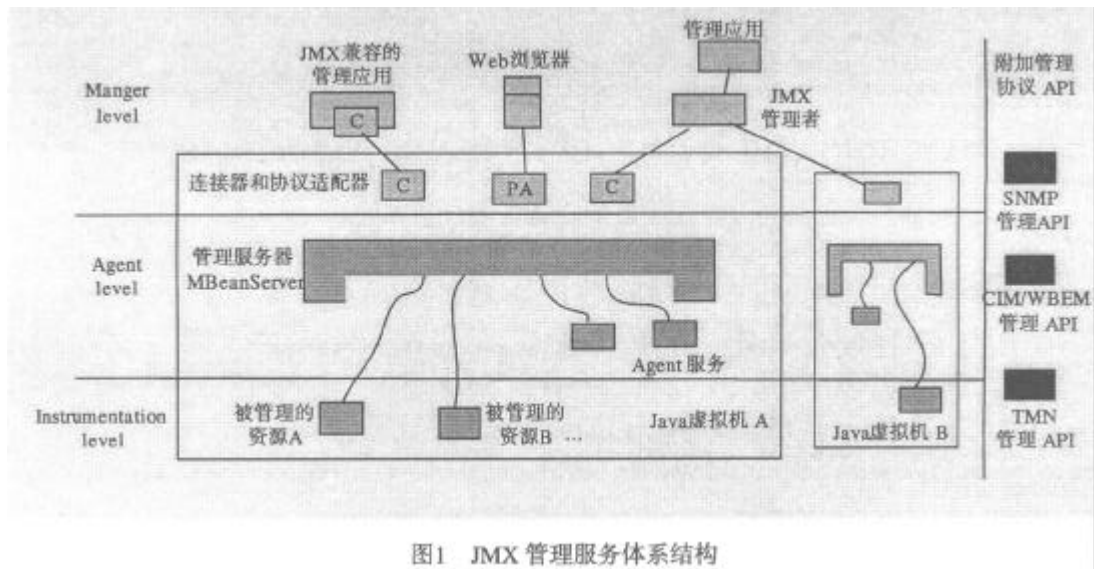


图1 JMX 管理服务体系结构

2 动态管理服务核心的设计

2.1 管理对象 MBean 的定义

MBean 采用了包装类(Wrapper)的设计模式：将 Java 应用、组件以及分布式网络中的各种资源包装为类似 JavaBean 的组件，称为 Managed Bean 或 MBean，实现了 Java 对象的可管理性。

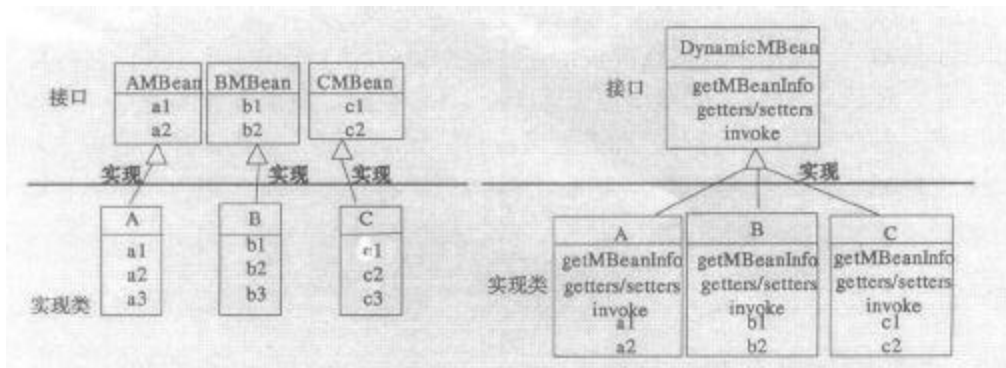


图2 Stander MBean 的实现

图3 Dynamic MBean 的实现

目前定义了两类 MBean 来封装被管理的资源：Stander MBean 和 Dynamic MBean。每个 Stander MBean 都有自己 MBean 接口，该接口详细地定义了 MBean 可被访问的属性和操作，适用于直接管理结构：被管理对象已预先定义好，且不需要经常改动。在这种情况下，开发一种新的可被管理的资源使用 Stander MBean 更加容易和直接。但是在很多情况下，被管理的对象需要经常引发和改动，

这就需要通过 Dynamic MBean 在运行时动态管理这些资源的属性以及操作。Dynamic MBean 直接实现特定的接口，不需要定义专属的 MBean 接口，为管理资源提供了一种更加灵活的方式。两种 MBean 的实现过程如图2和图3所示。

2.2 管理服务器 MBeanServer 的实现

Java 对象或其他被管理资源的 Java 封装在实现了 MBean 之后，必须在 MBeanServer 中注册，才能被 JMX 核心所管理。MBeanServer 的设计和实现是管理服务的重点，它在整个管理体系结构中起着承上启下的重要作用。

管理核心由以下一些模块构成：标准 TongMBeanServer 接口^[3]，为管理者提供标准的管理 API，屏蔽管理服务的低层细节，使管理者可以很方便地通过管理工具配置和监控 MBean；扩展接口 ExtendedMBeanServer 实现了持久化对象管理以及 MBean 的自动注册^[4,5]；RemoteMBeanServer 对 TongMBeanServer 进行 RMI 封装，用于在分布式网络环境的对象管理；MBeanServerFactory 实现了对象工厂的功能——创建以及管理所有的 MBeanServer；MBeanServerManager 是系统的入口点^[6]，负责整个管理系统的生命周期。MBeanServer 模块部分的设计思想如图4所示。

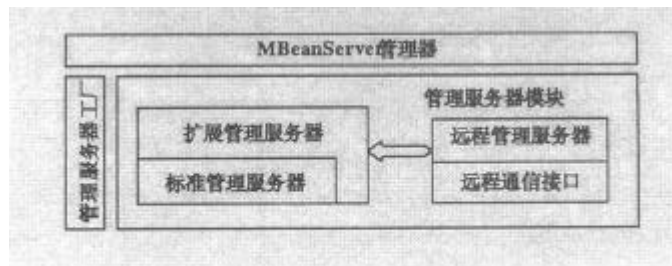


图4 MBeanServer 模块结构

1) 管理服务器 TongMBeanServer

TongMBeanServer 是整个管理服务的基础，主要负责创建，注册，注销以及删除 MBean，并提供管理者访问 MBean 对象的接口或方法，对已经注册的 MBean 进行监控和管理。根据管理应用的实际情况，TongMBeanServer 包含两种注册方式：根据系统运行情况，手动将所需管理的对象注册到管理服务器中；或者先将管理对象绑定在管理服务的名字空间(naming space)中，在启动管理服务的同时，系统自动将它们取出并在 MBeanServer 中注册。

2) 扩展管理服务器 ExtendedMBeanServer

ExtendedMBeanServer 继承 TongMBeanServer，实现了名字空间中 MBean 对象名字持久管理以及名字空间 MBean 对象自动注册的功能：建立 persistedMBeans Hash 表保存对象名 ObjectName 到名字空间中持久 MBean 对象信息之间的映射。启动管理服务的同时查找并取出名字空间中绑定的 MBean 对象，存放在 PersistedMBeans 表中并在 TongMBeanServer 中注册。当 MBean 从管理系统中注销后再次注册时，系统通过查询 PersistedMBeans 就可以获取它注销前最后的被管理状态，实现了 MBean 对象的持久化管理。

3) 远程管理接口 RemoteMBeanServer

RemoteMBeanServer 模块通过 RMI 封装实现了 TongMBeanServer 远程对象管理的版本。它提供了与 TongMBeanServer 模块相同的功能，但它并没有直接实现 MBeanServer 接口，因为该接口中定义的方法均没有抛出 RemoteException(远程访问异常)，而这正是使用 RMI 远程通信的机制所必需的。基于这点，将所有标准 MBeanServer 接口中定义的方法进行 RMI 扩展，那么远程对象就可以通过访问 RemoteMBeanServer 实现分布式系统的对象管理功能。

4) 工厂接口 MBeanServerFactory

采用面向对象设计模式中对象工厂的思想, MeanServerFactory 在系统中主要负责创建以及管理 MbeanServe: 根据具体的管理应用请求创建相应的扩展或远程 MBeanServer, 并将其对象应用保存在一个数组列表中, 通过对该列表的查询可获知当前系统中 MBeanServer 的实例情况。

5) MBeanSeverManag

作为特殊的管理资源 MBeanServer 本身没有实现任何 MBean 接口, 故不能直接被系统所管理。因此设计 MBeanServerManager 对象, 对 MBeanServer 的生命周期进行管理和配置。

2.3 远程管理通信机制

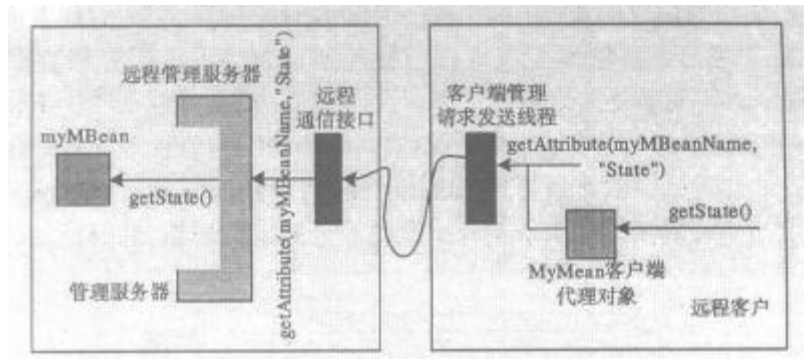


图5 远程管理通信机制

分布系统中的远程对象管理是通过 RemoteMBeanServer 和远程管理通信接口来实现的^[7,8], 因此设计一套如图5所示的管理系统必需的远程管理通信机制。以 RMI 协议的方式, 远程客户通过 MbeanToken(令牌)与本地主机的管理服务建立一个 session(会话), 以此来访问 MBeanServer。在这种情况下, RemoteMBeanServer 将建立一张 hash 表 currentTokens 保存当前所有 Token 和其时间对象之间的映射关系。下面是客户请求通过 RMI 访问 RemoteMBeanServer 时, 远程管理通信接口使用的辅助对象的实现:

- 1) AutoDiscoveryPacket 重新定义了 RMI 管理端口能够自动识别的包的包头, 如图6所示。
- 2) AutoDiscoveryPacketFactor 将接收到的网络数据流, 按照 AutoDiscoveryPacket 数据包的格式进行封装。
- 3) RMIPortQueryPacket 继承 AutoDiscovery Packet 类, 实现一个 RMI 端口查询包, 其包头如图7所示。
- 4) RMIPortResponsePacket 继承 AutoDiscoveryPacket 类, 实现一个 RMI 端口响应包, 其抱头如图8所示。
- 5) CommandSender 线程, 位于远程客户端, 实现 RMI 请求的发送: 将客户管理请求发送到管理服务器的监听端口。同时也负责取回服务器的响应信息。
- 6) CommandListener 线程, 位于管理服务器端, 负责监听网络数据包及服务器响应信息的回送。

'E'	'M'	'A'	'D'
Command		Packet Length	

图6 AutoDiscoveryPacke

'E'	'M'	'A'	'D'
0	1	0	12
Sender Port			

图7 RMIPortQueryPacket

'E'	'M'	'A'	'D'
0	2	0	16
RMIMgmtPort			
Mgmt Interface Version Number			

图8 RMIPortResponsePacket

2.4 管理通知机制 Notification

Mbean 的管理接口允许 MBeanServer 在其管理的对象上执行配置和管理的操作, 然而这些接口

Mbean 的管理接口允许 MBeanServer 在其管理的对象上执行配置和管理的操作,然而这些接口一般只定义了管理一个复杂的分布式系统所必需功能的一部分。更多的时候,管理应用需要对一个状态的改变或者一个特定的条件进行响应。这种情况下,使用 Notification 机制对管理接口进行灵活的扩展,通过广播接口广播管理事件,使得管理应用可以很及时的被管理对象的状态或属性的改变。

管理服务中的通知机制 Notification 建立在 Java 事件模式的基础之上,需要发送的广播 MBean 或 MBeanSever 实现 Broadcaster 接口,而管理应用或其它管理对象作为接收者实现 Listener 接口。同时,接收者通过设置过滤器 NotificationFilter 可以屏蔽掉那些不感兴趣的管理事件,提高系统处理管理事件的效率。Notification 机制使得一个 Listener 只需注册一次就可以接收 Broadcaster 中发生的所有不同类型的事件。

3 结束语

根据 MBean 对象的定义及 MBeanServer 模块的设计,本文论述了一种基于 JMX 动态管理服务核心部分的设计,该设计适用于 J2EE 以及其他分布计算平台对象的管理。目前,集成了该管理服务模块的 J2EE 应用服务器广泛地应用在联通165计费系统上,经过实际应用证明了这种管理服务具有可动态扩展、开发应用简单以及接口可通用等特点。同时对 JMX 管理服务中 JMX Agent 服务的设计、存储访问点的控制以及 Notification 机制的完善的研究打下了良好的基础。

参 考 文 献

- 1 Sun Microsystems, Inc. JMX Instrumentation and Agent Specification, V1.0. 2000,7. <http://www.java.sun.com>
- 2 Sun Microsystems, Inc. Java Management Extensions White Paper. 2001, 5. <http://www.java.sun.com>
- 3 Asensio J, Villagr V, Lopezde V J, *et al.* Experiences in the management of an EJB-based e-commerce application. The 8th HP OpenView University Association Plenary workshop, Berlin, Germany, 2001
- 4 Uehara, K, Inohara, S, Miyazawa H, *et al.* A framework of customizing transactions in persistent object management for advanced applications. Fourth International Workshop on , 1995: 84 -93
- 5 Richard H. Managing semantic heterogeneity in databases: a theoretical prospective. 1997
- 6 Erich G, Richard H, Ralph J 著. 设计模式. 李英军, 马晓星 译. 北京: 机械工业出版社, 2000
- 7 Cavallaro C, Raciti A, Torrisi A, *et al.* Reliability improvement of photo voltaic power conversion systems by an optimal remote-management controller. Proceedings of the Fourth IEEE International Caracas Conference on , 2002: 486-491
- 8 Shu-Yao Chien, Vassilis J, Carlo Z, *et al.* Efficient Management of Multiversion Documents by Object Referencing. 2001, 291-300