

基于 Java 代理和 HTTP 协议的 RMI 框架的实现

刘 瑾¹ , 黄龙达²

(1. 南京航空航天大学金城学院信息工程系, 南京 211156;

2. 国电南瑞科技股份有限公司电网分公司电力市场技术部, 南京 210061)

摘 要: 代理模式为其他对象提供一种代理以控制对目标对象的访问, Java 动态代理则为代理模式提供了一种简单易用的框架。基于 HTTP 协议的 Java 对象隧道技术实现对象消息传递, 可以方便地把这个技术集成到应用系统中。在介绍 Java 代理模式和基于 HTTP 协议的 Java 对象隧道技术的基础上, 提出了基于二者的一种 RMI 框架的设计与实现。该框架不但可以穿越企业内外网防火墙, 而且使得部署 RMI 应用与普通 Web 应用一样简单。

关键词: Java 代理模式; HTTP 协议; RMI; Java 序列化

0 引 言

RMI(Remote Method Invocation)是可以使不同的 Java 虚拟机之间进行对象间的通信的一种技术^[1]。RMI 作为一种远程方法调用机制, 借助于 Java 跨平台及高可移植性、安全性等 J2EE 技术架构优势在企业应用程序开发中得到了广泛的应用。但是传统 RMI 需要严格按照 RMI 规则设计程序, 例如: RMI 定义了 Remote 接口后, 还需要使用 rmic 工具生成服务器框架程序和客户端占位程序, 然后在服务器端实现该接口并注册, 才能在客户端使用, 这一系列流程使得使用传统的 RMI 比较繁琐, 同时在穿越企业内外网防火墙时将遇到困难。为了解决上述问题, 本文提出了一种基于 Java 代理模式和依赖 HTTP 协议的 Java 隧道技术的 RMI 框架。

1 Java 代理模式和基于 HTTP 协议的 Java 隧道技术

1.1 Java 代理模式

设计模式是使软件取得灵活性和代码重用的重要方法^[2-3], 而代理模式则为其他对象提供一种代理以控制对目标对象的访问。一般情况下可直接访问一个提供服务的目标对象, 而有时候可能因为目标对象的位置, 目标对象的存在状态等限制条件而不能直接访问目标对象^[4], 此时利用代理对象便可隐藏客户对象和目标对象的交互细节, 方便控制客户对象对目标对象的访问。JDK 提供的 Java 动态代理为 Java 实现代理模式提供了简单易用的框架, 主要由动态代理类和调用处理程序接口两部分组成。动态代理类是一个在创建类时实现指定的接口列表的类, 调用处理程序将 Invoke 方法返回的结果将作为代理实例上方法调用的结果返回。

1.2 基于 HTTP 协议的 Java 隧道技术

利用 HTTP 协议作为通信通道实现客户端对象对服务端对象方法的远程调用, 具体的通信规则是

收稿日期: 2009-12-18 修稿日期: 2009-12-21

作者简介: 刘瑾(1979-), 女, 新疆克拉玛依人, 硕士, 助教, 研究方向为软件体系结构、软件复用

利用基于HTTP协议的Java隧道技术,控制数据发送端输出流写入对象的顺序和数据接收端输入流读取对象的顺序应完全一致。

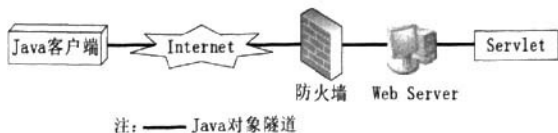


图1 Java对象隧道

2 基于Java代理模式和HTTP协议的RMI框架

2.1 RMI框架设计

HTTP协议是一个无状态的请求/应答协议,每次请求都需要建立一个新的连接。利用Java动态代理可以创建一个远程服务端对象的客户端代理,这样每次调用客户端代理对象的方法时,都会通过客户端的Java动态代理与服务端Java Servlet建立HTTP连接,远程服务对象执行完相关操作后,将结果返回给客户端代理对象作为其方法的调用结果。其中,Java动态代理实例在客户端扮演了RMI占位程序(Stub)的角色,而Java Servlet在服务端扮演了框架(Skeleton)的角色。框架主要由远程对象、服务器端框架、客户端占位程序和HTTP通信线路四部分组成,如图2所示:

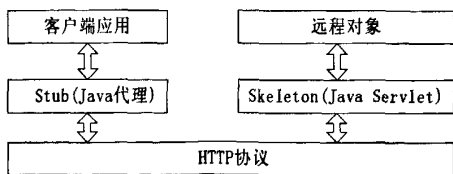


图2 RMI框架结构图

(1)远程对象:即服务器端业务逻辑对象,实现了业务逻辑接口,客户端应用在创建了客户端代理实例后将面向业务逻辑接口编程。

(2)服务端框架:即Java Servlet,在收到客户端代理调用请求后,动态创建业务逻辑对象,并调用其相应的方法,然后将调用结果返回,如果期间抛出异常则将该异常返回给客户端;

(3)客户端占位程序:即客户端代理实例,通过

JDK提供的Java动态代理类Proxy创建客户端代理实例;

(4)HTTP通信线路:客户端调用处理程序通过与服务端Servlet建立HTTP连接,并向连接输出流中写入Java序列化对象及从输入流中读取Java序列化对象,由此完成客户端代理实例与服务端的HTTP通信。

2.2 RMI框架工作机制

框架设计的核心就是通过Java动态代理类创建一个实现HTTP协议远程通信的调用处理程序接口,在此基础上构建代理实例。基于此现将RMI框架的工作流程描述如下:

①客户端代理实例与服务端Servlet建立HTTP连接;

②客户端代理实例将接口名称、接口实现类构造函数个数及参数对象、调用方法的名称、参数个数及参数类型、参数对象写入数据输出流,等待服务端响应;

③服务端Servlet依次接收客户端实例请求输入流并依客户端写入对象的顺序读取对象,并根据服务端的接口名称和接口实现类名的映射配置文件,依赖Java反射机制,动态创建接口实现类实例,并调用相应方法,并将成功标志位及结果对象写入到客户端的输出流。如果其中发生任何异常,则将失败标志位和异常对象返回;

④客户端代理读取标志位,如果成功则返回对象,否则抛出服务端捕获的异常。

2.3 RMI框架的Java实现

基于Java动态代理Proxy创建客户端的代理实例,创建一个实现调用处理程序接口的类,以实现与服务端Servlet的HTTP连接,并向输出流写入发送的对象,等待服务端响应。而服务端Servlet用于接收客户端HTTP请求,主要是解析请求的对象流,调用相应对象方法,并返回执行结果。

①定义业务逻辑接口BusinessInterface,该接口中定义其提供的操作方法。客户端是面向接口的编程,无需关心真正的业务方法的具体实现。服务端的真正业务对象BusinessImpl必须实现该接口,并真

正实现业务接口中声明的方法,该框架中服务端是通过读取一个业务逻辑接口和业务对象类的映射配置文件进行动态加载业务对象类;

② 定义 RMI 调用处理程序接口实现类 `HttpRMHandler`, 该类必须实现 `invoke(Object proxy, Method method, Object[] args)` 方法。在利用 Java 动态代理类 `Proxy` 的 `newProxyInstance` 方法创建客户端的远程业务对象代理实例时, `HttpRMHandler` 类将作为一个参数传入, 代码如下所示:

`Proxy.newProxyInstance (this.class.getClassLoader (), new Class [] {BusinessInterface }, HttpRMHandler);` 由此, 每当客户端代理实例调用相应的方法时, `HttpRMHandler` 类的 `invoke` 方法将会被动态调用, 该方法实现与服务端 `RMIServlet` 建立 HTTP 连接, 建立连接后将业务逻辑接口名称、接口实现类构造函数个数及参数对象、调用方法的名称、参数个数及参数类型、参数对象写入数据输出流, 等待服务端响应; 如果先读取到成功调用标志位 1, 则再次读入调用的结果对象, 如果读取到调用失败的标志位 -1, 则依次读入接收的 `Exception` 对象, 并在客户端抛出该异常;

③ 建立 `RMIServlet`。 `RMIServlet` 初始化时读取业务逻辑接口到业务逻辑对象的映射文件, 接收客户端代理发送的 HTTP 请求, 并按照客户端写入对象的顺序依次解析接收的对象, 动态创建业务对象实例:

`Object targetObj = Class.forName //服务端动态创建业务对象实例`

`(beanClassName). //beanClassName:映射文件读取到业务逻辑对象类全称的字符串`

`getConstructor(conParas). //conParas:业务对象类构造函数的参数类型数组`

`newInstance(constructorArgs); //constructorArgs:业务对象构造函数的参数数组`

`Method method = Class.forName (beanClassName).getMethod(methodName, //调用的方法名称`

`methodArgClass); //methodArgClass:方法的参数类型数组`

`returnObj = method.invoke (targetObj, //返回方法调用`

结果对象

`methodArgs); //调用方法的参数数组`

此时, 如果中间操作未抛出异常则依次将结果标志位 1 和方法调用结果对象 `returnObj` 写入 `RMIServlet` 的 `HttpServletResponse` 数据输出流; 若抛出异常则将 -1 和抛出的异常写入数据输出流。

3 结语

代理模式为分布式环境下调用对象和目标对象的物理位置差异提供了一种控制访问对象的方法, 而 JDK 提供的 Java 动态代理为 Java 实现代理模式提供了简单易用的框架。基于 HTTP 协议的 Java 对象隧道技术实现对象消息传递, 可以方便地把这个技术集成到应用系统中。基于 Java 代理模式和 HTTP 协议的 Java 对象隧道技术实现的一种可以穿越防火墙的远程方法调用框架, 为 Java 平台在因特网上构建部署分布式应用提供了一个简单高效的解决方案。

参考文献

- [1] 余科华, 李杰. 基于 Java-RMI 的移动代理技术的研究[J]. 微计算机信息, 2009, 1-3: 247~248
- [2] James W. Cooper. The Design Patterns[M], 1998
- [3] 张志华, 田英爱, 秦奕青. 设计模式在对象标识管理器中的应用[J]. 微计算机信息, 2006, 11-3: 276~278
- [4] Partha Kuchana. Software Architecture Design Patterns in Java[M]. Auerbach Publications, 2004
- [5] [HTTP://www.w3.org/Protocols/rfc2616/rfc2616.html](http://www.w3.org/Protocols/rfc2616/rfc2616.html)
- [6] Karl Moss. Java Servlet 编程指南[M]. 北京: 机械工业出版社, 2000

(下转第 144 页)

参考文献

- [1] Jan van Bon. 基于 ITIL 的 IT 服务管理基础篇. 章斌译. 华大学出版社, 2007
- [2] 刘宇熹, 陈尹立. 计算机系统服务外包及运行维护管理. 清华大学出版社, 2008
- [3] 石双元, 杨琴, 单进. IT 基础设施运营成熟度模型框架初步研究[J]. 管理学报, 2006(01)
- [4] 万江平, 王云凤, 郑楚卫. IT 服务管理的知识支持结构研究[J]. 计算机应用研究, 2008(03)
- [5] Carrie Higday -Kalmanowitz. E.Sandra Simpson. Implementing Service and Support Management Processes: A Practical Guide. Van Haren, 2005

Design of Workflow Engine of Operation Management

LIU Yu-xi

(Department of Computer Science and Technology, Guangdong University of Finance, Guangzhou 510521)

Abstract: Discusses the operation and maintenance management platform in the design of business development for the constant demands on the system function into the workflow engine. To achieve the creation of documents easy process, without writing any code; easy to build business processes, and can seamlessly integrate into the current some IT service processes; to facilitate the process of configuration, management, through the graphical tool to determine the definition of functional processes to facilitate the process changes.

Keywords: Workflow Engine; Custom Process; Modeling

(上接第 139 页)

Implementation of RMI Framework Based on Java Proxy and HTTP Protocol

LIU Jin¹, HUANG Long-da²

(1. Department of Information Engineering, College of Jincheng, Nanjing University of Aeronautics and Astronautics, Nanjing 211156)

2. Department of Electric Market, NARI Technology Development Co., Ltd., Nanjing 210061)

Abstract: Proxy design pattern provides a proxy for controlling the access of the target object, and Java dynamic proxy is a simple framework of implementation of proxy pattern. The technology of Java object tunnel based on HTTP protocol can realize the transfer of Java object message, and can be easily integrated into application system. Introduces Java proxy pattern and technology of Java object tunnel based on HTTP protocol, then cites the design and realization of a RMI framework depending on Java proxy and HTTP protocol. The RMI framework not only can go through the firewalls of enterprise, but also can be deployed as easily as deploying popular Web application.

Keywords: Java Proxy Pattern; HTTP Protocol; RMI; Java Serialization

作者：[刘瑾](#)，[黄龙达](#)，[LIU Jin](#)，[HUANG Long-da](#)
作者单位：[刘瑾, LIU Jin\(南京航空航天大学金城学院信息工程系, 南京, 211156\)](#)，[黄龙达, HUANG Long-da\(国电南瑞科技股份有限公司电网分公司电力市场技术部, 南京, 210061\)](#)
刊名：[现代计算机（专业版）](#)
英文刊名：[MODERN COMPUTER](#)
年，卷(期)：2010 (2)
被引用次数：1次

参考文献(6条)

1. [Karl Moss](#) [Java Servlet编程指南](#) 2000
2. [查看详情](#)
3. [Partha Kuchana](#) [Software Architecture Design Patterns in Java](#) 2004
4. [张志华;田英爱;秦奕青](#) [设计模式在对象标识管理器中的应用](#)[期刊论文]-[微计算机信息](#) 2006(33)
5. [James W. Cooper](#) [The Design Patterns](#) 1998
6. [余科华;李杰](#) [基于Java-RMI的移动代理技术的研究](#)[期刊论文]-[微计算机信息](#) 2009

引证文献(1条)

1. [刘玉江](#) [Java RMI设计实现的分析与研究](#)[期刊论文]-[电脑开发与应用](#) 2010(9)

本文链接：http://d.g.wanfangdata.com.cn/Periodical_xdjsj-xby201002035.aspx