

基于 Java 的 SNMP 网络管理系统

袁健美

(湘潭大学数学与计算科学学院, 湘潭 411105)

摘要: 将 Web 服务技术应用于网络管理中, 提出一个通用的分布式与集中式相结合的网络管理系统模型。在实现其基本功能的同时运用 Java 技术建立并部署应用程序, 采用封装技术提高系统的扩展性和灵活性。基于用户界面层、中间层和存储层的 3 层体系结构, 设计并实现该网络管理系统。

关键词: 网络管理; 系统模型; Java 技术; Web 服务

SNMP Network Management System Based on Java

YUAN Jian-mei

(School of Mathematics and Computational Science, Xiangtan University, Xiangtan 411105)

【Abstract】 Using the Web service technique in network management, a model system of network management combined by the mode of distribution and centralization is proposed. After implementing the basic applications, the expansibility and convenience of the network management system are improved. A network management system with high efficiency and usability is built based on the Java technology.

【Key words】 network management; system model; Java technology; Web service

1 概述

网络是计算机通信网的重要组成部分, 它通过互连与协同工作来构成大范围的信息处理系统。网络管理指通过监督、组织、控制网络通信服务和信息处理等活动, 确保计算机网络的持续正常运行, 并在运行出现异常时及时响应和排除故障^[1]。如何有效地进行区域内网络的管理是计算机网络能够有效、可靠、安全、经济地提供服务的重要保障。

在 OSI(Open System Interconnect)网络管理标准中定义了网络管理的 5 个功能: 配置管理, 性能管理, 故障管理, 安全管理, 计费管理。Java 技术改变了建立和部署应用程序的方法, 为网络管理提供了新的手段^[2]。本文基于简单网络管理协议(Simple Network Management Protocol, SNMP), 应用 Java 语言设计并实现了具有高效率高可用性的网络管理系统。

2 网络管理系统模型的构造

2.1 系统整体模型

现在的网络主要基于 SNMP 集中式的网络管理^[3]。SNMP 是基于传输层的 UDP 数据报协议, 在管理者和被管设备代理之间传输管理信息。其监管过程为: 在被监控端, 嵌入网络设备或主机设备中的 SNMP Agent 收集设备的各种信息和统计数据, 并不断地把这些数据记录到 MIB 库(管理信息库)中; 在管理端, 通过 SNMP 协议向 SNMP Agent 发出 SNMP 报文, SNMP Agent 收到报文后向 SNMP 管理端响应应答包。

根据 SNMP 协议特性, 本文构造了网络管理系统的设计模型, 如图 1 所示。网络客户端的发送线程与服务器端的接收线程共同负责完成数据的发送、接收和请求命令的解析工作, 服务器端的处理流程控制程序负责完成数据的收集、处理和流程的转向等工作, 数据处理工作在服务器端与数据库端进行。模型建立了从客户端到 Internet, 再到服务器端和数据库端的应用, 构造了将网络平台和管理调度进行

有机结合的软件系统, 实现了具有自动配置和自动调整功能的网络管理, 并形成了用户界面、中间层和存储层的 3 层网络管理体系结构。

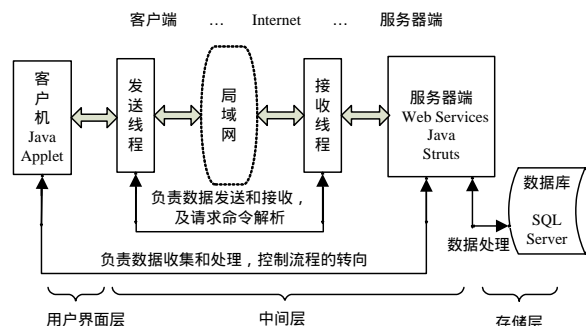


图 1 网络管理系统模型

2.2 3 层体系结构

在 3 层网络管理体系中, 用户界面是基于网页形式和独立运行的 Java Application, 用户通过窗口或 Web 页面来监视管理网络状态; 存储层把关系数据库作为网管信息的存储介质, 这些数据包括网络设备的基本信息、网络性能数据等; 中间层是整个网络管理模型的核心部分, 是管理功能和 Java 技术的主要实现层。考虑到网络通信技术的发展和网络间异构性的普遍, 在此系统模型中进一步引入 Web 服务技术, 把已有的网络管理功能包装起来, 探讨在网络异构环境中不同

基金项目: 国家“973”计划基金资助项目“高性能科学计算研究-大规模并行计算研究”(2005CB321702); 国家第一类特色专业建设点基金资助项目“信息与计算科学”(高教函[2007]31号); 湖南省教育厅划块基金资助项目(07C748)

作者简介: 袁健美(1977-), 女, 讲师、硕士, 主研方向: 并行计算, 计算机网络与软件应用

收稿日期: 2009-02-20 **E-mail:** yuanjm@xtu.edu.cn

操作系统平台间实现组件的互操作，实现 2 个或多个实体之间的协调、通信和合作等，从而提高系统面向管理数据时的实时性和安全性，完成更为复杂和高效的网络管理功能。文中的模型设计克服了单一集中式网络管理中的不足，如管理站信息流量大、管理效率低和灵活性差等，构造了采用分布与集中模式相结合的网络管理系统模型。

3 系统的 Java 设计与实现

3.1 系统设计

系统设计与功能模块具体描述如下：

(1)Java 网络编程：“请求-应答”模式，C/S 与 B/S 混合软件架构，Java 处理 SNMP 报文，Java 反射与 XML 配置文件，数据采集，数据传输，数据库应用。

(2)Web 服务：解析 HTML, CSS, JSP, 信息获取，服务描述，交互数据，用户界面，Web 管理。

(3)Struts 技术：MVC 模式，中间层，JavaBean 与 EJB 组件，JSP 页面产生，Servlet 组件。

(4)Hibernate 与 CVS :应用与数据库间的“持久层”，JDBC 对象封装，copy-modify-merge 机制，存储仓库，目录式模块。

(5)其他技术：存储层，SQL Server，Eclipse 工具，插件 (plug-in)，关系型数据库，进程控制，日志管理，监控分析。

其中，以 Java 网络编程为主，运用 MVC 框架 Struts，引入对象持久层 Hibernate(开放源代码的 O/R Mapping)和项目开发管理工具 CVS(Concurrent Versions System)来提高开发效率。运用多种主流开发技术相结合，可全面实现网络管理系统的功能。

3.2 Java 技术的应用

Java 编程语言适应于互联网的分布式环境，采用“以对象为导向”的方式。使用 Java 编写的应用程序既可在单机上运行，也可分布在网络服务器端和客户端运行，还可用来编写容量很小的应用程序模块或 Applet，提供网页使用者和网页之间的交互式操作。网络管理系统中应用 Java 语言这些特性，设计实现了由客户机发出请求，服务器监听到请求后，为客户机提供响应服务的模型框架，即典型的“请求-应答”模式。

客户机/服务器(C/S)在分布处理过程中,使用基于连接的网络通信模式,在 C/S 之间定义一套通信协议,并创建一个 Socket 类,利用这个类建立一条可靠的链接,C/S 在这条链接上传输数据。其运作过程为:(1)服务器监听相应端口的输入;(2)客户机发出一个请求;(3)服务器接收到此请求;(4)服务器处理这个请求,并把结果返回给客户机;(5)重复上述过程,直至完成一次会话过程。

按照以上 5 个过程,使用 Java 语言分别编写了面向服务器和客户机的应用程序。该程序在服务器上时,程序负责监听客户机请求,为每个客户机请求建立 Socket 连接,从而为客户机提供服务。在设计中为使系统兼有 C/S 的高交互性和 B/S 的高扩展性,采用了两者混合的软件架构。同时基于应用层管理协议,运用 Java 技术处理 SNMP 报文,其中使用信息结构(SMI)、管理信息库(MIB)和管理协议(SNMP) 来对数据库网络资源进行管理。在开发 3 层网络管理体系中的中间层时,系统的 C/S 和 B/S 交互代码部分实现了模型的核心功能,后期功能整合中还运用了 Java 的反射与 XML 配置文件相结合的方法。

3.3 Web 服务与 Struts 技术

Web 技术与网络管理相结合,可在网络异构环境下使计

算机能动态地查找网络管理的接口并把接口集合起来,通过它们的互操作完成网络管理的功能。通过对数据的表示、传输的方式、Web 服务的描述、信息的获取等功能模块的设计,在此系统中网络管理员利用 Web 浏览器可随时随地管理网络,从而使网络管理更加方便快捷。

开放源码的 Struts^[4]是为 Java Web 应用提供的通用框架。Struts 提供了开发 MVC(Model View Controller)系统的底层支持,采用的主要技术有 Servlet, JSP 和 Custom tag library。作为一个 MVC 的框架, Struts 针对 Model(应用的业务逻辑)、View(应用的表示面)和 Controller(应用的处理过程控制)分别提供了 JavaBean 和 EJB 组件、JSP 页面产生和 Servlet 组件来实现。本文运用这种设计把网络管理系统的应用逻辑、处理过程和显示逻辑分成不同的组件实现,且 MVC 模式的应用使程序具有良好的扩展性和维护性。

4 应用实例

4.1 College 网络管理系统

应用上述的系统模型和开发技术,本文在 J2EE^[5]规范上采用 Eclipse 编译环境开发了 College 网络管理系统,用来管理高校机房。所开发的通用的、无人值守和远程控制的网络管理系统平台,在客户端和服务器端分别实现了以下主要功能模块:“客户端功能模块”和“服务器端功能模块”,其中包括[管理人员]模块、[学生|普通用户]模块和[系统管理员]模块等。该网络管理系统以用户进行客户端登录来触发服务器的监控和管理功能,当用户获得客户机应用权限后服务器启动监控线程进行全流程的及时追踪,并通过实时信息交互来实现系统的远程管理。系统应用与管理交互流程见图 2。

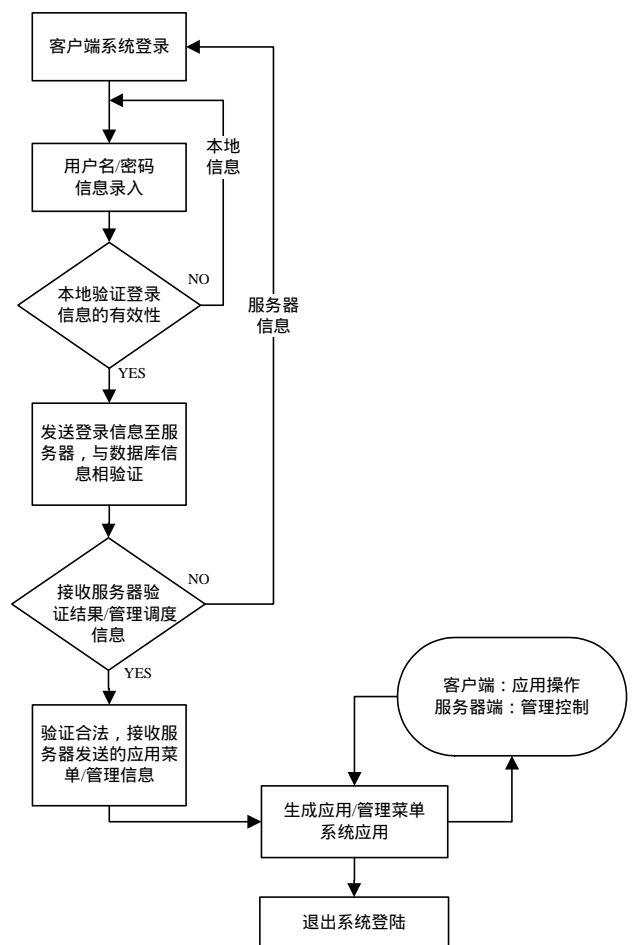


图 2 系统应用与管理交互流程

在服务器端和客户端分别设置了 XML 菜单信息文件，用来记录菜单信息，其中包括菜单名称、功能名称代码、菜单更新时间、完整的类包名、依赖的库文件等信息。对网管人员而言，只要把用户情况、设备情况及用户与网络资源之间的分配关系输入此网络管理系统，系统的“自动配置”和“自动调整”功能将自动地建立图形化的人员与网络的配置关系，并自动鉴别用户身份，分配用户所需的资源(如网络、文档服务等)。

4.2 系统部分代码

(1)服务器端监控客户端请求代码如下：

```
public Server_monitor (int port)
//定义并启动服务器的 Socket 用来监听客户机的连接请求
{if(port == 0) port = Default_Port;
this.port = port;
try {listen_socket = new ServerSocket(port); }
catch(IOException e) fail(e, "Exception creating server
socket");
System.out.println("Server_monitor: listening on port", port);
This.start(); }
public void monitor_run() //服务器监听线程的主程序
{ try { while(true)
//循环执行该线程，监听接受客户机发出的连接请求
{ Socket client_socket = listen_socket.accept();
Connection c = new Connection(client_socket); }}
//每个连接均产生一个连接对象，通过 Socket 通道进行通信
catch(IOException e) fail(e,"Exception while listening for
connections"))
public static void main(String args[]) //启动服务器主程序
{ int port = 0;
if (args.length == 1)
{ try { port = Integer.parseInt(args[0]);
catch(NumberFormatException e)
port = 0; }
new Server(port); }}
class Connection extends Thread
//定义 Connection 类用来处理与客户机的所有通信线程
{ protected Socket client;
protected DataInputStream in;
protected PrintStream out; }
```

(2)客户端信息管理代码如下：

```
User user=new User(); //为客户机端的登录创建用户信息列表
System.out.println("用户名："+text_Number.getText());
System.out.println("用户名："+text_password.getText());
user.setUser_Number(text_Number.getText());
user.setUser_password(text_password.getText());
user.setUser_loginTime(String.valueOf(System.currentTimeMillis
()));
...//进行用户登录验证，判断用户权限，若为普通用户则显示用
//户个人面板
if(validateResult.getAuthorizable() == AuthorizableInterface.
COMMON_USER)
{parent.dispose(); //停止后台监视任务管理器的线程
Observe_taskmgr.stopMultiThreading();
WindowsTray personcontrol=new WindowsTray(); //提供系
//统引用托盘，进行本地资源应用
```

```
}
... //判断用户权限，如为管理员则显示管理员面板
else if(validateResult.getAuthorizable() == AuthorizableInterface.
MANAGER_USER)
{parent.dispose(); //停止后台监视任务管理器的线程
Observe_taskmgr.stopMultiThreading();
WindowsTray personcontrol=new WindowsTray(); //提供系统引
//用托盘，进行本地资源应用
}
... //判断用户权限，如为系统管理员则开放系统所有权限
else if(validateResult.getAuthorizable() ==
Authorizable Interface. SYSTEM_USER)
{... //系统管理员操作
}
```

4.3 系统管理的实现

服务器端启用 Web 服务可随时查询和监控网络内所有客户机的状态，通过分析所采集到的用户登录某客户机信息，提供具体的网络管理操作。如通过获得的机器号、用户学号、用户名、当前应用截屏显示等信息，给出对应的远程控制、管理指令操作等，系统 Web 管理界面如图 3 所示。

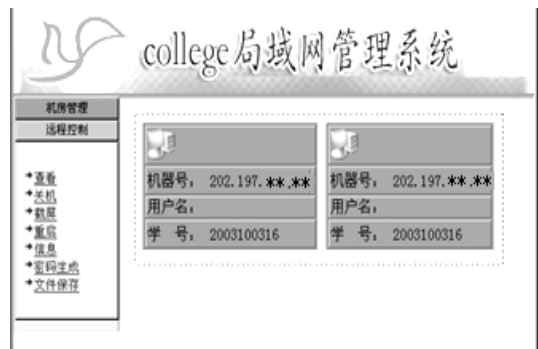


图 3 系统 Web 管理界面

5 结束语

针对网络管理系统的设计研究，提供了一种在实际环境中探索网络特性和实施网络管理的方法。本文将分布与集中模式相结合，提出一种通用的网络管理系统模型，探讨该系统的技术实现，并给出具体应用实例，可作为网络管理研究和软件系统开发的借鉴。

参考文献

- [1] Schonwalder J, Pras A, Martin-Flatin J P. On the Future of Internet Management Technologies[J]. IEEE Communications Magazine, 2003, 41(10): 90-97.
- [2] 赵学伟, 沈旭昆, 齐 越. 基于 Web 的交互式三维发布系统[J]. 计算机工程, 2007, 33(22): 243-245.
- [3] 崔建群, 吴黎兵, 肖德宝. SNMP 协议版本共存机制的研究[J]. 计算机应用, 2007, 27(7): 1117-1719.
- [4] 鲍胜利, 钟 勇. 基于 Struts 框架和 Procedure 的 Web 开发模式[J]. 计算机工程, 2008, 34(9): 67-69.
- [5] Johnson R, Hoeller J. J2EE Development Without EJB[M]. JavaEye, 译. 北京: 电子工业出版社, 2005.

编辑 顾姣健