基于 WMI 的 Windows 服务器监视系统的研究与实现

Research and Implementation of Windows Server Monitor System Based on WMI Technology

周中雨 （中国民航信息网络股份有限公司 100010）

摘要：建设 Windows 服务器监视系统可以减轻系统管理员的工作压力，提高管理水平。本文讨论了 Windows 服务器监视系统的解决方案，并给出了基于 WMI 技术的 Windows 服务器监视系统的功能、系统结构及实现技术。

关键词：WMI WINDOWS 服务器 监视系统

1 概述

随着 Internet 的飞速发展和广泛应用，网络应用系统越来越多。这些应用系统都需要服务器来承载，因此服务器的数量也越来越多。服务器采用的操作系统可分为 Windows 和 UNIX(LINUX)两类，Windows 服务器数量占一定比重。

服务器是网络应用系统的关键设备，需要保证 7×24 小时不间断运行，因此需要有专门的系统管理员来负责管理。系统管理员为了掌握服务器的运行状况，经常需要查看服务器的 CPU 运行情况、内存使用情况、文件系统使用情况、进程信息、在线用户情况等服务器运行状态参数。

对于 UNIX 服务器，管理员可以远程登录通过 Shell 命令查看系统运行信息。而对于 Windows 服务器，没有简洁统一的界面显示系统信息，对实时监视造成困难。为改变系统管理员工作中的被动状态，减小系统管理员的工作压力，需要建立服务器监视系统，使得系统管理员能够在任何时间、任何地点都可以查看服务器系统的运行状况，要求监视系统发现故障后及时通知系统管理员，并且尽可能不增加服务器的负担，保证服务器安全运行。

2 解决方案

开发 Windows 服务器监视系统的常规思路是采用 Client/Server 结构。在被监视服务器上安装管理代理(Agent)，采集服务器运行状态参数并发送到监视服务端。监视服务端接收运行状态信息并保存数据供查询使用。自行开发管理代理的工作量比较大，并且可能产生安全隐患，所以大部分的监视系统基于 Windows 提供的功能组件进行开发。比较常见的解决方案是基于 SNMP(Simple Network Management Protocol，简单网络管理协议)服务的监视系统。本文介绍基于 WMI(Windows Management Instrumentation)技术的监视系统，此系统功能更加强大，开发工作量更小。

WMI[2] 是 Windows 2000 操作系统的一部分，也可用于 Windows NT 4.0 和 Windows 98/95。WMI 有两个主要部分组成，一个是实际的 WMI 部件(WINMGMT.EXE)，另一个是 WMI 库。CIM(Common Information Model)库是存储可管理的静态数据的中心数据库。当请求 WMI 数据时，如果请求信息是静态的，那么静态数据将从 CIM 库中取出。如果请求信息是动态的，那么使用特定的 WMI Provider 获取信息。WMI 的结构如图 1 所示。

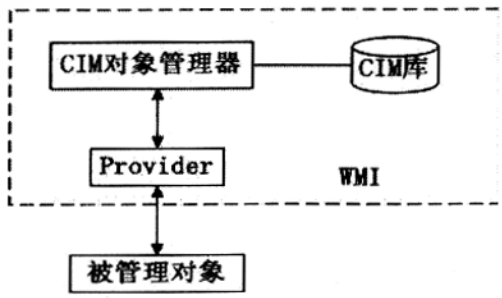

图 1 WMI 结构

基于 WMI 的 Windows 服务器监视系统如图 2 所示。

在这样的系统中，监视程序通过 COM/DCOM 获取本机或者远程计算机的运行状态信息。系统管理员通过配置管

理程序设置对 Windows 服务器的状态监视参数;参数设置后,配置管理程序将设置信息存放在数据库中,并通知监视

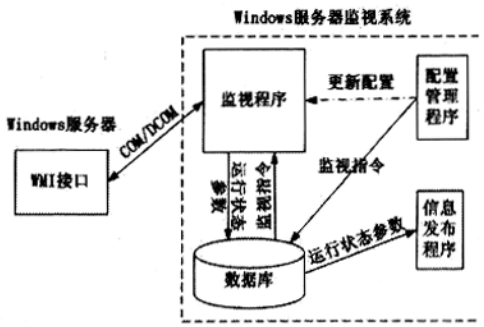

图 2　基于 WMI 的 windows 服务器监视系统

程序。监视程序获取运行状态信息并存储在数据库中。系统管理员通过信息发布程序可以查看 Windows 服务器当前的或各个历史时间的工作状态。

此方案的优点是对 Windows 系统提供了统一的管理接口,WMI 允许应用程序的开发者,使用简单的、一致的机制,去查询企业中的任一台 Windows 计算机上的信息,或是进行系统配置。并且通过 WMI 可以取得 Windows 系统中大部分的信息,功能强大。缺陷是 WMI 技术依赖于 NETBIOS 服务,可能引起安全问题,本文第三部分将提出解决方案。

3　实现

3.1　功能设计

Windows 服务器监视系统的主要功能是根据系统管理员的指令收集各服务器的运行状态信息,并以表格形式提供给系统管理员,使得系统管理员能够方便地查看各服务器的运行状况。从系统管理员关心的实际情况出发,对 Windows 服务器的监视主要包括以下内容[1]:

(1) CPU 信息:记录 CPU 的使用率;

(2) 存储信息:记录内存使用率、剩余空间、错误页/秒;

(3) 磁盘 I/O 信息:记录逻辑磁盘的文件系统类型、磁盘容量、剩余容量等;

(4) 网络信息:记录网络适配器的最大速度、接收字节/秒,发送字节/秒、错误数据包数等;

(5) 进程信息:记录每个进程的详细情况,如进程名称、占用 CPU 百分比、占用虚拟内存数量、占用内存数量、优先级、线程数目、运行累计时间等;

(6) 服务信息:记录 Windows 服务信息,如是否启动、启动类别、登录身份等;

(7) 用户信息:得到计算机中用户和组的信息,得到文件目录权限;

(8) 应用活动状态:记录一些应用的运行信息,如 IIS 中错误的请求总数、POST 请求速率、发送字节数/秒、接收字节数/秒、GET 请求速率、当前匿名用户数、最大连接数、当前连接数、总字节数/秒等信息;SQL Server 的数据文件大小、用户连接数、总使用内存大小等;

(9) 系统日志监视:取得 Windows 事件日志和性能日志。

对于不同的信息,采集的频率不同,对每一类设置不同的采集时间间隔。服务器运行状态参数被采集到数据库中,系统管理员通过浏览器就可以查看各服务器的运行状态,并且当发现数据超出系统管理员设置的阈值时,系统会主动报警。

3.2　实现技术

Windows 服务器监视系统是基于数据库的网络应用,数据库采用 ORACLE,WEB 服务器采用 Apache,所有系统都运行在 Windows 环境。在具体实现技术上,监视服务器程序可以采用 C++ 语言或其他语言实现,配置管理程序与信息发布程序采用 JSP 实现。

WMI 提供了强大的功能取得远程服务器上的信息。例如通过以下代码可以取得远程机器上的进程列表。

```
……
ConnectionOptions oConn = new ConnectionOptions();
oConn.Username = "test"; /远程计算机用户
oConn.Password = "1234"; /远程计算机口令
ManagementPath p = new ManagementPath("\\ServerName\root\cimv2");
ManagementScope ms = new ManagementScope(p,oConn);
ObjectQuery oq = new ObjectQuery("SELECT * FROM Win32_Process");
ManagementObjectSearcher query1 = new ManagementObjectSearcher(ms,oq);
ManagementObjectCollection queryCollection1 = query1.Get();
foreach (ManagementObject service in queryCollection1){
……
}
```

3.3　网络安全分析

使用 WMI 技术会产生潜在的安全问题,但是这种配置很容易避免。一般情况下,Windows 服务器放置于防火墙

内部，放置于防火墙外的监视服务器对 Windows 服务器进行监视。由于远程监视 Windows 服务器时需要服务器开放 Netbios 服务，这样就需要防火墙开放 139 端口，以便 Netbios 服务能够顺利通过防火墙，这样就产生了安全问题，如图 3 所示。

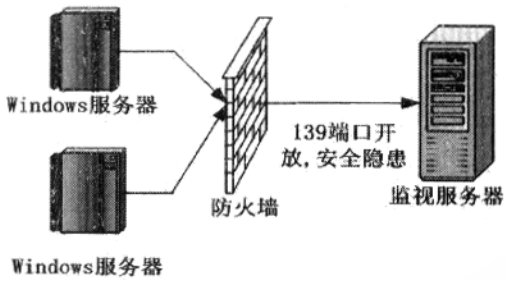

图 3　存在安全隐患的网络结构

可以通过下面的配置来避免安全隐患，在下面的环境下，将监视服务器放置在防火墙内，不需要防火墙开放 139 端口，防火墙外根本无法扫描到 Windows 服务器上的 Netbios 服务，Netbios 服务完全被封闭在防火墙之内，因此不会存在安全隐患，管理员可通过 Web 浏览器查看运行状态参数。消除安全隐患后的方案如图 4 所示。

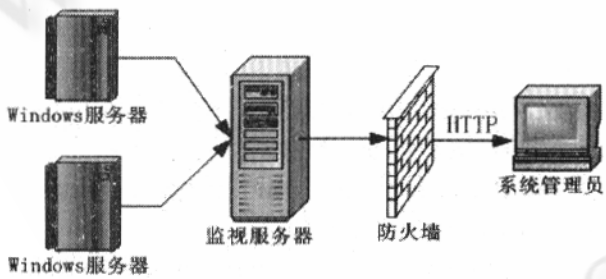

图 4　消除安全隐患的网络结构

3.4 后续工作

WMI 提供了对 Windows 服务器进行控制的功能，为开发 Windows 服务器控制系统提供了技术保障。例如通过以下代码可以对 Windows 服务器进行重新启动的操作。

```
……
Set obj = Locator.ConnectServer("ServerIP", "root\cimv2","user", "password")
Set objj = obj.ExecQuery("select * from Win32_OperatingSystem where Primary=true")
for each x in objj
x.reboot()
next
……
```

4 结束语

WMI 是复杂的、功能强大的工具，利用 WMI 技术可以完成对 Windows 服务器的监视和控制任务。基于 WMI 技术的 Windows 服务器监测系统正处于研究阶段，经过不断进行改进和完善后，将应用于清华大学的 Windows 服务器管理中。

参考文献

1 秦刚、李俊，基于 SNMP 的网络设备监视系统的设计与实现[J]，计算机工程与应用，2003,13:197－199。

2 张建章、苗宏，创建基于 Web 的 Windows 管理器方法[J]，计算机应用研究，2001,09:130－133。