文章编号:1001 - 9081(2007)01 - 0252 - 02

综合性能监控管理模型的设计与实现

戴大蒙, 李虎雄,陈赛娉
(温州大学 计算机科学与工程学院,浙江 温州 325000)
(jsj_ddm@ wzu. edu. cn)

摘 要:分析计算机性能监控平台存在的不足,采用 WMI 和消息中间件技术,构建一种跨操作系统、跨设备的综合性能监控管理模型,通过在金融行业的具体实现证明该模型能高效管理各种 IT 资源,及时报告故障前兆,具备自动修复能力,可扩展性强,适用于信息高度集中的大中型企业。

关键词: 性能监控; Windows 管理规范; 消息中间件
中图分类号: TP393.07 **文献标识码**:A

Design and realization of a model on computer performance monitoring

DAI Da-meng, LI Hu-xiong, CHEN Sai-pin
(*Department Computer Science & Engineering, Whenzhou University, Wenzhou Zhejiang* 325000, *China*)

Abstract: Basing on Windows Management Instrumentation (WMI) and Middleware technology, a model of performance monitoring in multiple operating system environment has been given, for the defects which exist in some computer performance monitoring platforms. The practice in financial business proved that, by this model, the IT resources can be effectively managed and maintained. The platform constructed on this model can report many breakdown omens promptly, and has the automatic maintenance ability. In conclusion, this model is of universal significance.

Key words: performance monitor; WMI(Windows Management Instrumentation); Message Middleware

目前在计算机性能监控领域,基本上采取简单网络管理协议技术(Simple Network Management Protocol, SNMP)和 WMI[1] 对网络设备和 Windows 服务器进行性能监管,但对于综合的 IT 应用环境,国内的计算机性能监控软件基本上处于实验阶段,并普遍存在几大缺陷:1) 不提供跨设备、跨平台监控功能,无法在同一个界面上监控管理不同种类的设备或不同操作系统的服务器;2) 只局限于性能监测结果的显示;3) 排除故障需手工处理。本文构建的性能监控模型,以温州建行为实现对象,在很大程度上弥补了以上不足。该模型能实时采集各项性能数据,包括来自网络、操作系统、数据库和应用系统的运行状态,对超出阈值的性能指标进行分析,确定导致性能下降的原因或故障根源,并根据预警策略通知系统管理员,同时进行简单的自动维护处理。

1 模型的构建与设计方案

信息系统中需要监控的 IT 资源主要是网络设备和服务器,不同应用环境有不同操作平台的服务器。以金融企业为例,柜面业务一般运行在 UNIX/Linux 服务器,办公则采用 Windows 服务器。如何将不同设备、不同操作系统的性能数据集成在一个平台上进行监控管理,是监控模型要解决的首要问题。考虑到 WMI 已逐渐成为 Windows 平台性能监测的事实标准并能很好的支持 SNMP 协议,因此在模型构建中,以 WMI 体系结构为核心,分别利用 SNMP 数据提供者和 WIN32 数据提供者[1] 实现网络设备和 Windows 服务器的监控管理任务,对于 UNIX 服务器则采取消息中间件[2] 的技术,通过 API 接口无缝地集成到管理模型中。当管理对象出现故障前兆时,如何快速定位系统瓶颈,并作出预警提示,也是监控模型急需解决的问题,本模型采用预警平台来捕捉潜在故障,采用多种预警措施进行处理。由图 1 可知,管理应用程序是模型架构的关键,具体内容放在平台实现中重点介绍。

1.1 基于 WMI 的性能监控模块

图 1 左下虚线框是利用 WMI 组件[2] 实现对网络设备和 Windows 操作平台的监控,由数据提供者、CIM[3] (Common Information Model)对象管理器和管理对象构成。其中数据提供者实时监控管理对象,从不同的管理接口提取管理信息,并将这些管理信息和接口映射成对象类,通过 COM API 提供给 CIM 对象管理器以响应管理应用程序的请求(具体实现见关键技术)。模块中的核心部分是 CIM 对象管理器,它通过本身带有的数据库(CIM 数据库)传递和保存管理对象的数据信息,起着桥梁的作用。当管理应用程序发出信息请求时,CIM 对象管理器分析该请求,判断哪些数据提供者具有对应的信息,然后通过 COM 接口将请求发送到相关的数据提供者,从不同的管理对象读取数据信息,将结果返回管理应用程序的同时一并保存在 CIM 数据库中。

1.2 基于消息中间件的 UNIX 服务器性能监控模块

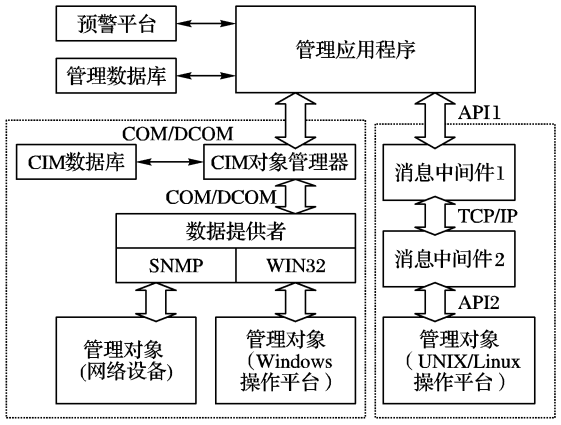

图 1 性能监控管理模型框架

UNIX 操作系统提供了众多的性能检测命令,将这些检测

收稿日期:2006 - 07 - 19;**修订日期**:2006 - 09 - 24

作者简介:戴大蒙(1975 -),女,浙江温州人,讲师,硕士,主要研究方向:分布式数据库、计算机网络; 李虎雄(1973 -),男,湖北孝感人,实验师,主要研究方向:计算机网络、信息安全; 陈赛娉(1980 -),女,浙江乐清人,助理实验师,主要研究方向:信息管理系统.

命令组合到 SHELL 程序中,能方便地生成功能强大的性能检测工具,但要将检测数据传递给运行在 Windows 操作系统下的管理应用程序,急需解决通信问题。一种简单的实现方法是:性能监控平台和管理对象采用 FTP 命令方式,传送检测指令和性能数据文本,再由管理应用程序将性能数据装载到数据库中。但这方法有明显的安全隐患:1) 必须注明对方机器的 IP、用户和口令;2) 传送文件容易被截获和伪装;3) 不能保证文件成功到达等。采用消息中间件能克服这些缺点,实现数据在不同操作系统不同机器间的可靠传输。如图 1右虚线框所示,当管理应用程序需要检测 UNIX 服务器性能时,先构造请求报文,调用消息中间件提供的 API 接口,采用 TCP/IP 协议传送到对应管理对象的消息中间件,同样通过 API 接口与 UNIX 服务器进行交互,并将处理结果打包逐级返回到管理应用程序。

2　综合性能监控管理平台的实现

管理应用程序是模型的核心部分,负责显示、分析和处理从管理对象获取的数据信息,可通过参数设置用户界面调整系统运行环境,主要由四部分构成:采集程序、数据显示分析程序、参数管理程序以及预警处理程序,如图 2 虚线部分所示。

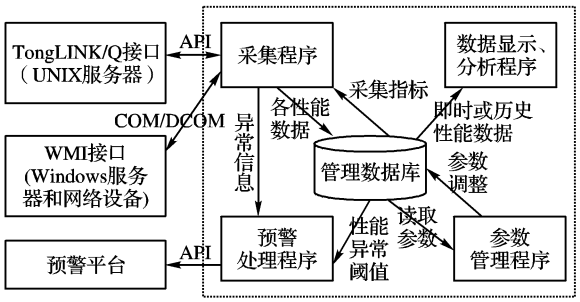

图 2　综合性能监控管理平台

图 3　综合性能信息监测

采集程序以后台服务进程方式启动,根据设定的采样周期扫描管理对象表,提取管理对象的特性信息,若标志为网络设备和 Windows 服务器,直接调用 COM/DCOM 接口,进入 WMI 监控体系;若标志为 UNIX 服务器,则调用消息中间件的 API 接口进行数据采集,并将采集到的不同设备的运行状态信息存入管理数据库,如网络流量、最近 10s 内 CPU 平均使用率、内存使用率、硬盘使用率、业务系统运行情况、备份执行情况等,所以,管理对象相对于用户而言是完全透明的。一旦监测设备出现异常状态(性能数据超过预先设定的阈值),立即启动预警处理程序,生成多种报警信号(如短信、E-Mail 等),传送到预警平台并及时通知指定的维护人员。数据显示和分析程序则根据系统设定的时间间隔从数据库中获取相关的性能信息,既可在一个 Web 页面上集中显示多个设备的即时性能信息,也可以对某台设备的即时或历史性能数据进行具体分析,查找系统的瓶颈,为合理配置系统资源提供决策依据。参数管理程序提供各种参数设置的功能,如预警阈值、数据采样周期、数据显示的刷新频率、需要采集的性能数据指标等等,实现个性化的监控管理。图 3 是监测 IT 运行状态的 Web 页面。

3　系统实现中的关键技术

3.1　管理数据库的设计

管理数据库是模型得以实现的有力保障,除提供各项必需的数据外,如管理对象基本信息、即时和历史的性能检测数据,还提供个性化设计的信息,如管理命令、检测指标、预警控制信息和处理流程的定义,这些设计极大地提高了系统的扩展性和适应性,以下为系统中关键的几张表:

(1) 管理对象表(IP、管理对象名称、设备类型、分类、设备位置、操作系统),存放和维护管理对象的节点信息;

(2) 监控参数表(数据采样周期、数据采集目录、最早可以报警时间、最迟允许报警时间、Web 页面刷新频率等);

(3) 监控指标参照表(指标编号、说明、执行方式、对应的操作系统、详细命令);

(4) 检测指标分类表(IP、指标编号、量化分档、上限、下限),其中 IP 为 0 表示所有机器,量化分档共四类,分别以绿、黄、红、黑表示;

(5) 故障预警表(IP、指标编号、量化分档、预警方式、通知对象、预警内容、连发间隔时间);

(6) 历史检测数据表/实时检测数据表(IP、指标编号、检测时间、检测值)。

3.2　性能数据的采集

3.2.1　UNIX 操作系统的数据采集策略

从模型上可以看出,UNIX 服务器性能数据必须通过消息中间件才能传送到管理机,因此对应的数据采集分两步进行,首先借助于 UNIX 强大的 SHELL 命令实现原始数据的采集,部分采集命令见表 1;然后将运行结果(性能数据)重定向到文本文件中,通过 TongLINK 消息中间件的 API 接口与管理应用程序进行交互。UNIX 服务器端,调用 cmfpsjjh 函数进行数据交互;在管理平台则通过 cmfpjssj 和 cmfpfssj 函数分别进行数据的接收与处理结果的返回。

表 1　SHELL 数据采集命令

命令名称	作用	举例说明
sar	CPU 的使用率	sar 1 10 //查看最近 10s 的 CPU 平均使用率
sar –r	内存使用率	sar –r 1 3 //查看最近 3s 的内存平均使用率
swap –l	交换区使用率	swap –l //查看实时虚拟内存大小和使用率
df –v	硬盘使用率	df –vlawk '{print $2 " " $6}' //查看硬盘各文件系统的使用率
ps	进程运行情况	ps –u rootlgrep –c cron //查看 root 用户启动 cron 进程个数

3.2.2　WMI 数据采集策略

对 Windows 服务器的数据采集,系统通过 WMI 体系结构中的 WIN32 数据提供者负责从 Windows 系统访问数据,并将这些性能数据映射在相应的类上,网络设备则通过 SNMP 数据提供者进行访问,道理相同。当管理应用程序发出信息请求时,COM API 直接查询这些类(见表 2)并访问相应的属性,获知Windows服务器的状态,包括CPU信息、存储信息、磁盘

(下转第 256 页)

3）中断服务程序：当 PDIUSBD12 向微控制器发出中断请求时，读取 PDIUSBD12 的中断传输来的数据，设定事件标志，交由主循环程序来处理；

4）标准请求处理程序：对 USB 的标准设备请求进行处理；

5）厂商请求处理程序：对用户添加的厂商请求进行处理；

6）主循环程序：发送 USB 请求、处理 USB 总线事件和用户功能处理等。

3.3 上位机程序设计

上位机的程序主要包含 USB 的驱动程序及应用程序两部分。其中驱动程序部分直接使用周立功单片机公司提供的驱动程序，在 Windows 2000 或 Windows XP 下安装后即可使用。应用程序部分则调用周立功单片机公司提供的动态链接库 EasyUSB.dll 来读写端点的数据[5]。

如图 6 所示，点击“上传”按钮，PC 在与下位机握手后就会读取端点 2（对应 D12 的主端点）的数据，并以二进制文件的形式保存，接受完毕后会弹出对话框提示接受完毕，否则会提示握手不正确。点击“清除”按钮，向下位机发送清除命令，下位机完成后会回复一个清除完毕命令，PC 接收到这个命令后会弹出对话框提示清除完毕，否则会提示清除失败。点击“打开文件”按钮，会把本次上传的数据文件打开。

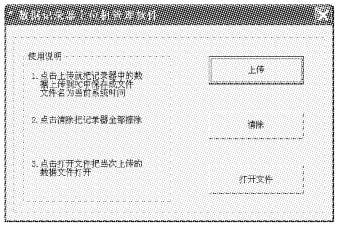

图 6 记录器上位机管理软件界面

4 结语

在设计并实现了数据记录器后，按照设计要求在实验室对它进行了测试，步骤如下：

1）将一个文件通过 RS-422 串口下载到数据记录器中保存；

2）通过 USB 接口将记录器中的数据通过数据记录器上位机管理软件，上传到 PC 并以文件的形式保存；

3）设计一个文件比较软件，将源文件和上传的二进制数据文件进行比较，比较结束后如果完全正确会弹出对话框，告诉你两个文件完全一样，否则提示文件不同，并计算误码率。

按照以上步骤，对各种大小的文件，RS-422 使用多种波特率，对记录器进行了大量测试，测试结果表明上传文件全部正确，误码率为零，从而证明数据记录器设计方案正确和可行。现在该记录器已经在工业现场中实际使用，完全满足设计要求、使用方便。

参考文献：

[1] 周立功. LPC2210 ARM 微控制器数据手册 1[Z]. 广州：广州周立功单片机发展有限公司，2004.

[2] 王田苗. 嵌入式系统设计与实例开发[M]. 北京：清华大学出版社，2002.

[3] 周立功. ARM 嵌入式系统基础教程[M]. 北京：北京航空航天大学出版社，2005.

[4] 周立功. ARM 嵌入式系统实验教程（一）[M]. 北京：北京航空航天大学出版社，2004.

[5] 周立功. PDIUSBD12 USB 固件编程与驱动开发[M]. 北京：北京航空航天大学出版社，2003.

（上接第 253 页）

I/O 信息、网络信息、进程信息、服务信息、用户信息、应用活动状态信息和系统日志信息等。

表 2 WIN32 数据提供者映射的常见类

类名称	属性	对应的性能数据
win32_Processor	MaxClockSpeed	CPU 的主频
Win32_LoginMemory Configuration	TotalPagefileSpace	内存的虚拟值
Win32_PerfRawData_PerfOS_Memory	AvailableMbytes	内存的可用值
Win32_ diskDrive	Size	硬盘的总量
Win32_LogincalDisk	FreeSpace	逻辑硬盘的空闲值
Win32_PerfRawData_Tcpip_NetworkInterface	CurrentBandwidth	网络的带宽
	BytesTotalPersec	网络的流量

某些性能数据并不能从相关的类中直接获取，需要进行一定的计算，如 CPU 使用率、内存空闲率、硬盘空闲率、硬盘碎片空间等等，下面以内存空闲率为例，简要介绍性能数据获取的实现原理。

```
Set colItems = objWMIService.ExecQuery_("Select * From
  Win32_PerfRawData_PerfOS_Memory")        '查询内存使用值类
For Each objItem in colItems
avmem = objItem.AvailableMbytes            '获取可用内存(kB)
Next
set wbemObjectSet = objWMIService.ExecQuery_("Select * From
  Win32_LogicalMemoryConfiguration")       '查询虚拟内存类
For Each wbemObject In wbemObjectSet
tomem = wbemObject.TotalPhysicalMemory/1024
                                           '获取物理内存(KB)
Next
data_mi = INT(avmem * 100/tomem) '计算空闲内存率
```

3.3 预警处理机制

预警处理程序启动后，首先从“故障预警表”中读取所有记录，并与采集程序中传递的性能信息进行比对，对超出阈值的性能数据，根据预警方式的不同调用不同的预警接口程序，发送相应的警报信息。对于原因确定的故障，可运行预先设计好的修复程序自动排除。其中，短信通知是最常用预警措施，本文采用东方软件公司开发的 SMIAS（短信网关系统）[4] 提供的 SP 接口，实现短信预警，鉴于篇幅，仅给出关键代码。

```
cmpp_connect_to_ismg("134.130.51.68",7890,&conn);
          //向短信网关 134.130.51.68 发送连接请求
cmpp_login(&conn,&cl);   //向短信网关登录注册用户
cmpp_submit(&conn,&cs);      //发送短信到短信网关
cmpp_recv(&conn,&cp,0);
                  //接收短信网关返回的处理信息
```

4 结语

本文利用 WMI 标准和消息中间件通信技术，实现了跨操作平台、跨设备的性能监控管理系统，提供故障发现、预警以及自动修复等多项功能，并在温州建行得到成功应用。实践证明，该模型使用价值高，通用性强，尤其适合于金融、电信、电力、民航和铁路等信息系统集中化程度高、高可靠性运行环境的应用场合。

参考文献：

[1] MATTHEW L, ASHLEY M. Windows Management Instrumentation (WMI)[M]. Indianapolis: New Riders Publishing, 2001.

[2] 徐晶，许炜. 消息中间件综述[J]. 计算机工程，2005，31(16)：73－76.

[3] 周宇，赵成栋，康建初. 应用性能管理中的相关标准和技术[J]. 计算机工程，2004，30(18)：101－102.

[4] 刘荣辉，刘光昌. 短消息代理网关系统的设计与实现[J]. 电信科学，2005，(3)：23－26.