

基于 SNMP 的计算机性能监控系统设计与实现

陈建国

(福建工程学院 软件学院, 福建 福州 350003)

摘要: 围绕计算机性能数据的监听和控制问题展开, 深入研究计算机性能数据提取技术、SNMP 协议、Sockets 等网络通信技术和远程监控技术, 提出一种基于 SNMP 的计算机性能监控模型并进行系统实现。该系统实现对计算机 CPU、内存等性能数据进行实时监听和控制管理。实验证明, 系统运行良好, 能够有效监听目标计算机的各项性能信息, 及时控制管理目标计算机, 保证目标计算机系统稳定运行。

关键词: 计算机性能; 远程监控; 实时监听; 智能预警

中图分类号: TP 311.13

文献标志码: A

文章编号: 1672-4348(2014)03-0225-07

Design and implementation of SNMP-based computer performance monitoring system

Chen Jianguo

(Software College, Fujian University of Technology, Fuzhou 350003, China)

Abstract: The monitoring and control of computer performance data was focused. A systematic computer performance monitoring model based on SNMP was proposed and implemented, which was based on the study of the extraction technology of computer performance data, SNMP protocol, Sockets network communication technology and remote monitoring technology. The system enables the real-time monitoring and control of the performance data of CPU and memory. Experimental results indicate that the system can function properly, can monitor the performance data from target computer effectively, and can control the target computer in time to ensure the stability of the target computer system.

Keywords: computer performance; remote monitoring; real-time monitoring; intelligent warning

目前越来越多的企事业单位在业务运作和日常办公中采用软件系统, 各单位的软件系统具有数量越来越多、规模越来越庞大、复杂程度越来越高、对性能的要求也越来越高。因此, 要求承载这些软件系统的计算机服务器应保持良好、稳定的运行状态。开发一种能够实时监控企业机房服务器的远程计算机智能监控系统成为当今的热门研究课题。

国外著名的服务器管理软件提供商有艾德威特^[1]、美科利^[2]等公司, 这些软件可以监控服务器硬件、网络设备等设备的运行状态, 以及服务器

软件性能数据的监听。但是由于软件价格较高, 不能被国内中小型企业所接受。国内的服务器管理软件主要集中于利用摄像头进行机房环境监控, 对机房硬件设备的监测等方面。如远程机房火灾预警系统^[3], 何鹏采用 SNMP 协议对远程主机网络状态、网络流量进行监测^[4], 郭萌采用 SNMP 协议对机房网络设备、路由器、交换机等设备进行监测^[5]。现有计算机监控领域的研究主要集中于对计算机外围设备、网络设备等硬件进行监测。

本文提出一种基于 SNMP 的计算机性能监控

模型。该模型采用数据提取技术获取计算机运行过程中的实时性能数据(包括 CPU 性能、内存性能、网络带宽数据、进程数据等)。系统服务器通过 SNMP 协议、Sockets 等网络通信技术对提取的目标计算机性能数据进行远程监听,然后利用智能预警功能和远程控制技术对目标计算机进行远程监控和管理,实现目标计算机系统运行状态良好、稳定。最终保证运行于目标计算机上的各项应用程序和业务系统正常运作。

1 核心技术研究

1.1 计算机性能数据提取技术

WMI (Windows management instrumentation, Windows 管理工具^[6])是内置在 Windows Server 系列操作系统中核心的管理支持技术,主要用于访问和控制计算机系统中的底层数据。程序员通过 WMI 可以管理 Windows 系统中的 CPU、内存、磁盘、操作系统设置、事件日志、文件系统、网络组件、性能数据、进程、注册表设置、安全性等信息。

WMI 使用具有统一性和扩展性的面向对象接口,为各种高级应用程序制定了一系列与计算机底层 API 接口进行交互的标准方法。WMI 的技术框架如图 1 所示。

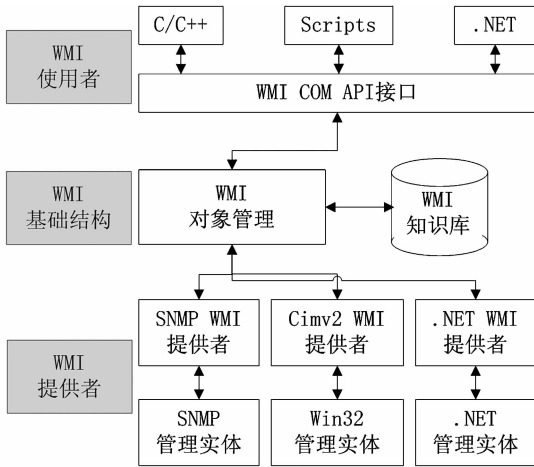


图 1 WMI 技术框架

Fig. 1 WMI technical framework

WMI 使用者是 WMI 技术使用的载体,提供与应用程序交互的 API 接口。使用 .NET 程序的 System Management 类与 WMI 下层进行交互,实现性能数据的查询和读取^[7-8]。

WMI 基础结构是 WMI 技术的核心,包含对象管理和知识库两个模块。

WMI 提供者是一种监控若干个托管对象的 COM 接口,托管对象可以是逻辑或者物理组件,如硬盘、网卡、操作系统、CPU、进程和服务等。

1.2 SNMP 协议

SNMP^[9] (simple network management protocol,简单网络管理协议)是一种基于 TCP/IP 协议栈的互联网应用层协议。利用 SNMP,一个管理工作站可以远程管理所有支持这种协议的网络设备,包括路由器、交换机、防火墙、网络主机、网络应用服务、甚至是温湿度传感器等。

SNMP 报文是指 SNMP 协议实体之间交换的 UDP 报文,由版本号、标识符、SNMP 共同体名和 SNMP 协议数据单元(PDU)组成^[10]。SNMP 报表格式如图 2 所示。

IP 首部	UDP 首部	版本	共同体	PDU 类型 0-3	请求标识	差错状态 0-5	差错索引	名称	值	名称	值	...
-------	--------	----	-----	------------	------	----------	------	----	---	----	---	-----

图 2 SNMP 报表格式

Fig. 2 SNMP report format

SNMP 采用代理/管理站模型,通过管理工作站与 SNMP 代理间的交互操作实现对网络的管理与维护。代理和管理站通过 SNMP 协议中的标准消息进行通信,每个消息都是一个单独的数据报文^[11]。SNMP 的工作原理如图 3 所示。

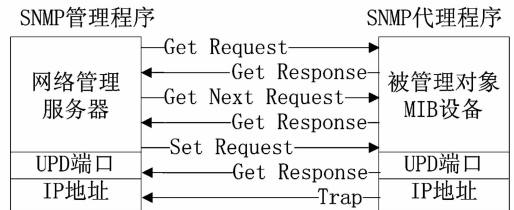


图 3 SNMP 工作原理

Fig. 3 The principle of SNMP

SNMP 协议的基本操作包括 get (获取)、get-next (获取下一个)、set (设置)和 trap (陷阱)。其中,get 和 set 是从节点读取数据和把数据写入节点的基本操作,get-next 用于在 MIB 层次结构上步进,trap 是从服务器(代理程序)到客户机(管理器)的一个主动提供的异步通知。

利用 SNMP 协议监控磁盘 I/O 的核心代码如下:

```
MIBS = " host agentx smux ucd-snmp/diskio disman/event-mib"
```

```
% configure
--enable-static --enable-shared
--with-syslocation = "192.168.0.1"
--with-persistent-directory = "/var/net-snmp"
--with-mib-modules = "$ MIBS"
% if % {tcp_wrappers}
--with-libwrap = % {_libdir}
% endif
--enable-ipv4
--enable-ucd-snmp-compatibility
--with-sys-contact = "admin@localhost" <
< EOF
```

1.3 Sockets 套接字

Socket 是应用层与 TCP/IP 协议族通信的一组中间软件抽象层接口,实现了基于 TCP/IP 协议的网络通信功能^[12]。应用程序通过 Socket 接口通信技术实现客户端和服务器之间数据流发送和接收功能。Socket 通信的具体工作原理如图 4 所示。

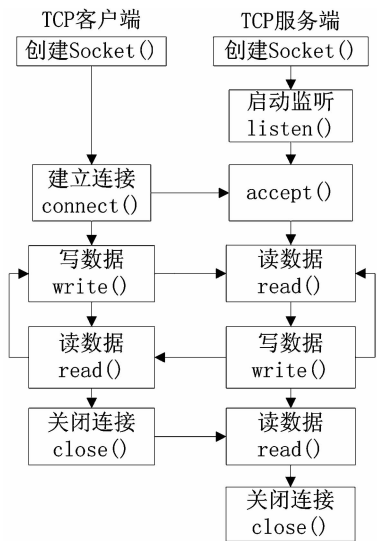


图 4 Socket 通信工作原理

Fig. 4 The principle of Socket communications

客户端工作过程:

(1) 创建一个 Socket 对象,使用 Connect() 方法向指定 IP 地址和端口号连接到目标服务器^[13]。

(2) 使用 write() 方法录入数据,创建 I/O 对象并使用 Send() 方法发送数据。

(4) 使用 Receive() 方法从服务器中接收数据,使用 Read() 从 I/O 对象中读取并显示数据。

(5) 使用 Close() 方法关闭连接和 Socket 对象。

服务器端工作过程^[14]:

(1) 配置一个有效 IP 地址,创建一个 Socket 对象并分配一个有效端口号。

(2) 使用 Listen() 方法监听客户端发送的连接请求,使用 Accept() 方法接收请求。

(3) 使用 Receive() 方法从客户端接收数据,使用 Read() 信息从 I/O 对象读取数据并进一步处理。

(4) 使用 write() 方法录入数据,创建 I/O 对象并使用 Send() 方法发送数据。

(5) 使用 Close() 方法关闭连接和 Socket 对象。

2 系统设计

2.1 系统整体框架

基于 SNMP 的计算机性能监控系统通过对计算机 CPU、内存等性能数据进行实时提取,通过 SNMP 网络协议将数据传送、监听和技术分析,利用智能预警功能和远程控制对远程计算机进行监控和管理,保证计算机系统稳定运行。整个系统由两个子系统组成,分别为目标计算机子系统和服务器子系统。系统整体框架如图 5 所示。

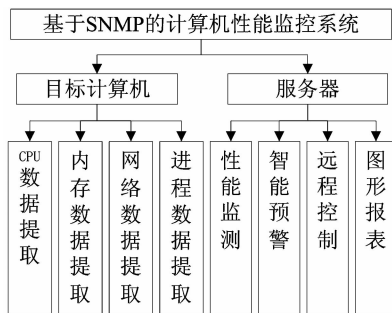


图 5 计算机性能监控系统整体框架图

Fig. 5 The architecture of computer performance monitoring and control system

系统主要功能说明如下:

(1) 性能数据提取:目标计算机子系统主要负责计算机性能数据的实时提取和发送,包括 CPU 性能、内存性能、网络带宽性能、进程实时数据等。

(2) 实时监听模块:服务器子系统主要负责实时监听各个受控目标计算机发送的性能数据,

对这些数据进行存储和显示。

(3) 智能预警模块: 服务器子系统根据具体业务要求为各目标计算机的各个监测对象设置预警阈值, 当目标计算机实时发送的性能数据超出该阈值时, 系统将自动提示报警信息并将信息发送到系统管理员手机。

(4) 远程控制: 系统管理员收到预警信息后, 通过服务器对目标计算机进行远程控制操作, 包括远程登录计算机、远程开关机、远程控制目标计算机的相关应用程序和服务等。

2.2 性能数据提取

目标计算机子系统主要负责计算机性能数据的实时提取和发送, 包括 CPU 性能、内存性能、网络带宽性能、进程实时数据等。本系统性能数据提取模块采用 .NET 平台 C# 开发语言, 结合 WMI 技术进行实现。

2.2.1 CPU 性能数据提取

提取运行过程中的计算机 CPU 信息, 包括 CPU 制造商信息、CPU 当前使用率、CPU 当前温度等。

CPU 性能数据项包括静态数据和动态数据两部分, CPU 性能数据提取的具体流程如图 6 所示。

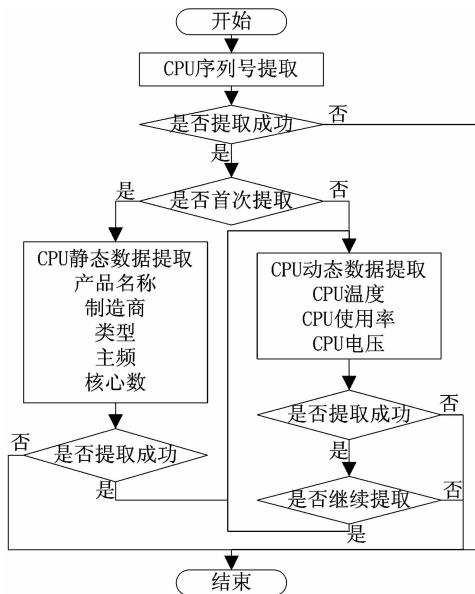


图 6 CPU 性能数据提取流程

Fig. 6 The extraction process of CPU performance data

CPU 静态数据如序列号、产品名称、制造商、类型、主频、核心数等, 静态数据仅需要在计算机连接时提取一次。动态数据如 CPU 温度、CPU 使用率、CPU 电压等, 动态数据需要实时提取, 提取

间隔由系统设定, 可以为 2 - 10 s。

CPU 详细性能数据项如表 1 所示。

表 1 CPU 性能数据项

Tab. 1 The performance data items of CPU

性能数据项	数据项说明
ID	序列号
Name	产品名称
Manufacturers	制造商
BasicFrequency	主频
NumberOfCores	核心数
ProcessorType	CPU 类型
Temperature	CPU 温度
LoadPercentage	CPU 使用率
Voltage	电压
PageSize	CPU 页面大小

2.2.2 内存性能数据提取

提取运行过程中的计算机内存性能数据, 包括物理内存和虚拟内存的总内存容量、当前已用内存等信息。

内存性能数据项包括静态数据和动态数据两部分, 内存静态数据如物理内存总大小、虚拟内存总大小、页面交换总大小等, 静态数据仅需要在计算机连接时提取一次。动态数据如已用和可用物理内存、已用和可用虚拟内存、内存使用率和可交换页面大小等, 动态数据需要实时提取, 提取间隔和 CPU 性能数据提取同样由系统设定。内存性能数据提取的具体流程如图 7 所示。

内存的详细性能数据项如表 2 所示。

表 2 内存性能数据项

Tab. 2 The performance data items of memory

性能数据项	数据项说明
TotalPhysicalMemory	物理内存总大小
LoadPhysicMemory	已用物理内存
AvailPhysicalMemory	可用物理内存
TotalVirtualMemory	虚拟内存总大小
LoadVirtualMemory	已用虚拟内存
AvailVirtualMemory	可用虚拟内存
LoadPercentage	内存使用率
TotalPageFile	页面交换总大小
AvailPageFile	可交换页面大小

2.2.3 网络性能数据提取

提取运行过程中的计算机网络性能数据, 包括网络实时带宽信息、网络状态信息等。

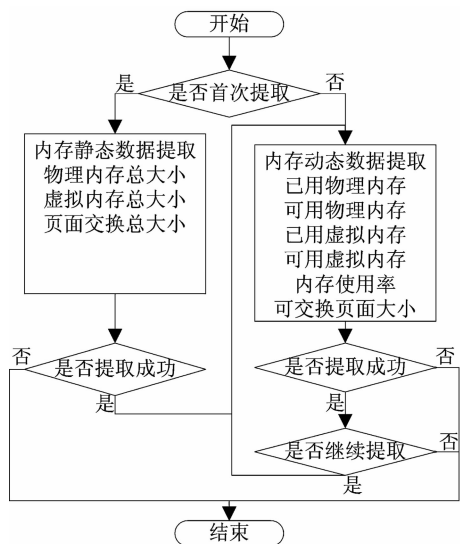


图7 内存性能数据提取流程

Fig.7 The extraction process of memory performance data

2.2.4 进程数据提取

提取目标计算机中当前正在运行的进程信息,包括当前各进程名称、CPU 占用率、内存占用率等。

2.3 智能预警模块

智能预警模块主要负责根据具体业务要求为各目标计算机的各个监测对象设置预警阈值,判断监听的各对象性能数据是否高于预先设定的各项阈值,如果超出预定的阈值,则自动提示报警信息并将信息发送到系统管理员手机。比如,CPU 使用率过高、CPU 温度过高、可用物理内存或可用虚拟内存过小等影响计算机稳定正常运行的关键性能数据项。性能数据智能预警工作流程如图8所示。

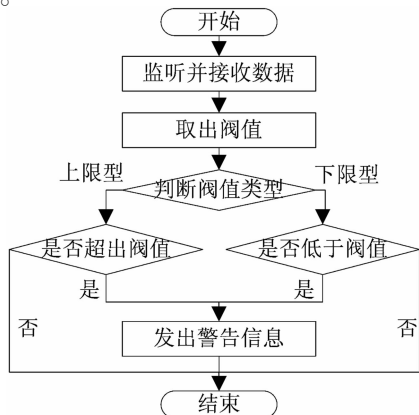


图8 智能预警工作流程

Fig.8 The workflow of intelligent warning

2.4 远程控制模块

远程控制模块由系统管理员进行操作,当系统管理员收到预警信息后,无需前往目标计算机进行实地操作,只需要通过服务器对目标计算机进行远程控制操作,包括远程登录计算机、远程开关机、远程控制目标计算机的相关应用程序和服务等,具体如图9所示。

该模块对于目标计算机处于异地施工状态或非上班时间计算机性能出现问题的情况下十分快捷、方便。

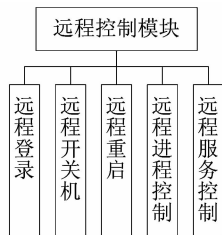


图9 远程控制模块功能结构

Fig.9 The functional configuration of remote control module

3 系统实现

本系统采用 .NET Framework4.0 作为开发平台、使用 C/S 模式进行实现。选用主流的 C#开发语言、XML、WMI 和 Windows 窗体设计技术进行实现。网络通信采用 SNMP 协议和 Socket 技术。系统界面遵循软件界面设计原则,强调界面简洁美观、可操作性强。

3.1 性能数据提取

目标计算机子系统独立于服务器子系统,主要负责计算机性能数据的实时提取和同步发送。本模块使用 C#语言调用 WMI 底层接口,从而提取计算机各硬件数据。WMI 的调用需要引入 System.Management 命名空间。本节以 CPU 和内存性能数据的提取过程为例进行阐述,CPU 性能数据提取功能的核心代码如下。

```
private static CPU_INFO CpuInfo;
ManagementClass mc = new ManagementClass ("W - in32_Processor");
ManagementObjectCollection moc = mc.GetInstance - s();
foreach(ManagementObject mo in moc) //遍历 CPU
{
```

```

cpuInfo = mo.Properties.Value;
x.Add(cpuInfo.ID) //CPU 序列号
x.Add(cpuInfo.NumberOfProcessors); //
CPU 数量
x.Add(cpuInfo.ProcessorType); //
CPU 类型
x.Add(cpuInfo.ProcessorLevel); //
CPU 等级
x.Add(cpuInfo.Temperature); //CPU
当前温度
x.Add(cpuInfo.PageSize); //CPU
页面大小
x.Add(cpuInfo.CpuLoad); //当前
使用率
}

```

CPU 和内存性能数据功能界面如图 10 所示。



图 10 CPU 和内存性能数据提取界面

Fig. 10 The extraction interface of CPU and memory performance data

内存性能数据提取功能的核心代码如下：

```

static MEMORY_INFO MemoryInfo;
static extern void GlobalMemoryStatus ( ref
MEMORY_INFO meminfo );
x2.Add ( MemoryInfo.TotalPhys ); //物理内
存总大小
x2.Add ( MemoryInfo.AvailPhys ); //可用物
理内存
x2.Add ( MemoryInfo.TotalVirtual ); //虚拟内
存总大小
x2.Add ( MemoryInfo.AvailVirtual ); //可用虚

```

拟内存

3.2 实时监听模块

实时监听模块实现服务器对所监控目标计算机的性能数据进行实时监听,目标计算机将实时提取的性能数据采用 XML 格式进行封装,通过 SNMP 协议和 Socket 套接字技术进行传输到服务器。服务器实时监听来自目标计算机的性能数据发送请求,接收数据并利用图形报表方式进行显示。实时监听模块功能界面如图 11 所示。

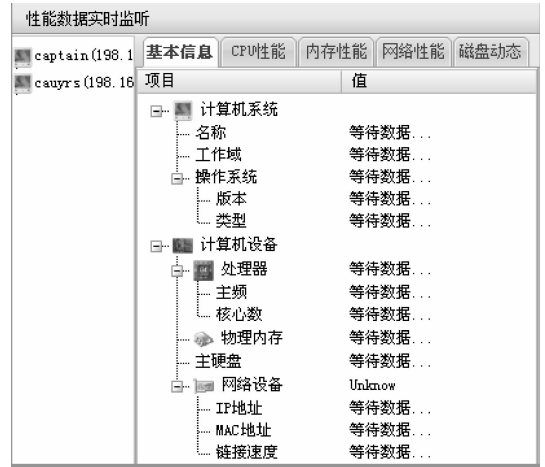


图 11 实时监听模块功能界面

Fig. 11 The functional interface of real-time monitoring module

实时监听模块采用 SNMP 协议进行网络通信,需要事先下载并引入 SnmpSharpNet 插件包,创建 SNMP 的核心代码为:

```

SNMPAgent myAgent = new SNMPAgent
("127.0.0.1", "public", "public");

```

创建一个 SNMP 代理,用于从目标计算机向服务器发送性能数据,参数为服务器的 IP 地址和代理可访问性变量。接着创建一个 SNMP 对象,用于将 XML 格式的性能数据序列化并进行发送。此过程需要引入 system.Xml.Linq 命名空间,核心代码如下:

```

//提取并生成性能数据
string cpulp = "CPU - LoadPercentage:87.5";
//序列化性能数据
StringBuilder buffer = new StringBuilder();
XmlSerializer myser = new XmlSerializer
(cpulp);
using (TextWriter writer = new StringWriter
(buffer))

```

```

}
myser.Serialize( writer, entity );
}
SNMPObject s = new SNMPObject ( buffer.
ToString()); //封装数据
myAgent.Send( s ); //发送数据

```

3.3 智能预警模块

智能预警模块负责为各目标计算机的各个监测对象设置预警阈值,判断监听的各对象性能数据是否高于预先设定的各项阈值。智能预警模块功能界面如图 12 所示。



图 12 智能预警模块功能界面

Fig. 12 The functional interface of intelligent warning module

4 结论

本文提出一种基于 SNMP 的计算机性能监控模型并采用 C#.NET 技术进行系统实现。该系统实现了对目标计算机的 CPU、内存等性能数据进行实时提取,通过 SNMP 网络协议将数据传送、监听和技术分析,利用智能预警功能和远程控制技术对目标计算机进行监控和管理,保证计算机系统稳定运行。

本文所监视的目标计算机主要包括企事业单位机房服务器、高校计算机实验室服务器等。这些目标计算机系统中运行进各种业务系统,本文的创新点在于通过使用 WMI 技术实现对目标计算机在运行过程中的各种性能特征数据进行实时提取和监视,同时使用智能预警功能将超出阈值的性能数据通知系统管理员,以便及时进行处理,保证目标计算机中各种业务系统的稳定运行。

系统测试结果表明:本系统运行稳定、能实时提取目标计算机的各项性能数据,达到预期设计目标,满足企业实际应用要求。本系统模型对企事业单位机房服务器远程监控、高校计算机实验室远程监控、机电和建筑等领域的应用有着重要的探索意义。

参考文献:

- [1] Williams T. Designers looking for the road to distribute real-time systems[J]. Computer Design, 2003(9): 81-83.
- [2] I-Hal L P, Broberg H L. Internet-based monitoring and controls for HVAC applications[J]. IEEE Industry Applications Magazine, 2000, 8(3): 49-54.
- [3] 黄艳阳, 杨学明. 基于 SNMP 的服务器性能监控技术[J]. 广西大学学报, 2009, 31(1): 189-190.
- [4] 何鹏, 邱建林, 顾翔. 基于 SNMP 协议的远程主机监测系统设计与实现[J]. 计算机工程与设计, 2008(13): 185-187.
- [5] 郭萌. 基于 SNMP 的网络性能管理系统设计与实现[D]. 西安: 西安电子科技大学, 2011.
- [6] 勒晓辉. 远程智能监控系统[D]. 天津: 天津大学, 2005.
- [7] 严诚幸. 远程智能抄表系统的设计与实现[D]. 成都: 电子科技大学, 2012.
- [8] 陆璐, 刘发贵. 基于 Web 的远程监控系统[M]. 北京: 清华大学出版社, 2008.
- [9] ARAI-Ali, AL-Rousan M. NET-based home automation system[J]. IEEE Transactions on Consumer Electronics, 2004, 50(2): 498-504.
- [10] 李进. 基于 linux 的远程智能监控系统的设计与实现[D]. 长沙: 湖南大学, 2011.
- [11] Safaric R, Sinjur S, Zalik B, et al. Control of robot arm with virtual environment via internet[J]. Proceedings of the IEEE, 2003, 91(3): 422-429.
- [12] 蒋漪涟. WINDOWS 服务器性能监控的设计与实现[J]. 微型电脑应用, 2010, 26(9): 22-24.
- [13] 郭志波. Lon works 技术在 Internet 远程智能监控中的研究和应用[D]. 南京: 南京工业大学, 2003.
- [14] Aacck W, Szymrensky. Embedded Internet technology in process control device[J]. IEEE, 2004(9): 35-37.