异构网络中 SNMP 数据采集监测系统研究

杨 勇[1],李 军[2],张越培[3]
(1. 华中科技大学 网络与计算中心,湖北 武汉 430074;
2. 华北水利水电大学 计算机学院,河南 郑州 450045;
3. 河南卫生职工学院 信息中心,河南 郑州 451191)

摘 要:针对异构网络中数据信息监测的要求,研究在网络异构环境中数据采集的实现方法。分析并使用简单网络管理协议(SNMP)和管理信息库(MIB),设计并实现一个能够对异构网络环境下的主机资源进行实时采集的系统软件。该系统使用关系数据库保存所有主机资源信息,并通过绘图的方式生成实时的主机资源占用曲线,以便对主机在一定时期内的资源占用状况进行综合分析。系统运行结果表明,可以有效地对网络主机资源进行全面监测,及时掌控网络资源运行状况。

关键词:网络管理协议;数据采集;异构网络;资源监测;关系数据库
中图分类号:TP302.1 **文献标识码**:A **文章编号**:1673-629X(2014)06-0087-05
doi:10.3969/j.issn.1673-629X.2014.06.022

Research on SNMP Data Acquisition System in Heterogeneous Network

YANG Yong[1], LI Jun[2], ZHANG Yue-pei[3]
(1. Network and Computing Center, Huazhong University of Science & Technology, Wuhan 430074, China;
2. College of Computer, North China University of Water Resources and Electric Power, Zhengzhou 450045, China;
3. Information Center, Henan Medical College for Staff and Workers, Zhengzhou 451191, China)

Abstract: For requirements of data information monitoring under heterogeneous network, the data acquisition method under heterogeneous environment is researched. Simple Network Management Protocol (SNMP) and Management Information Base (MIB) are analyzed and used to design and implement a host resource acquiring system which has the ability to monitor host resource usage under heterogeneous network at real-time. The system uses relational database to save entire record of network hosts monitered, generating real-time resource usage curve and providing the ability to comprehensively analyze the host resource usage in a long term period. The running result shows that the system has the ability to monitor host comprehensively and effectively, timely control of running status of network resource.

Key words: SNMP; data acquisition; heterogeneous network; resource monitoring; relational database

0 引 言

在异构网络环境中,不同网络应用服务器通常运行的是不同类型的操作系统或不同版本的操作系统,由于互联网应用服务器运行的时效性非常强,要求每个网络应用必须提供 7×24 小时不间断服务。因此,主动对网络应用主机资源进行数据采集和有效监测,第一时间发现、定位故障主机显得尤为重要。同时,通过采集主机资源的记录数据,用编程方式形成各个时段(如:日、周、月、季、年等)每一个主机资源图表信息,即可实现宏观可视信息,又可掌控不同网络应用每一时段的资源占用情况,也可辅助网络防火墙或流控设备,改变防护规则和实施策略,提高或改善网络服务质量。鉴于计算机病毒、网络攻击和网络故障的随机、突发性,如果仅靠人工监测则无论在空间上,还是时间上都是不太现实的,因此,选择一种高效、实用的数据采集工具,设计一套能对 IDC(互联网数据中心)正在运行的主机资源进行实时监测的系统并加以实现,这就是文中研究的核心主旨。

收稿日期:2013-08-19 **修回日期**:2013-11-27 **网络出版时间**:2014-02-24
基金项目:国家 985 工程项目;国家发改委专项(20082198)
作者简介:杨 勇(1960-),男,湖北武汉人,高级工程师,研究方向为网络与计算机通信、数据库;李 军,副教授,研究方向为计算机网络与安全;张越培,副教授,研究方向为计算机网络和信息化建设、规划。
网络出版地址:http://www.cnki.net/kcms/detail/61.1450.TP.20140224.0915.032.html

1 SNMP 协议简介

SNMP[1](Simple Network Management Protocol,简单网络管理协议)具有高可移植性的实现模块,有着不依赖于操作系统平台的支持特性,是网络管理事实上的标准。1990 年 5 月,RFC 1157 定义了 SNMP 的第一个版本 SNMPv1,随后陆续推出了功能更加丰富的 SNMPv2[2]和安全性能更高的 SNMPv3[3]。

SNMP 是工作于传输层之上的一个应用层协议,其主要功能是:在网络实体间获取设备资源数据,提供并传输管理信息,并借助于检查参数和监督特定的网络状态,对网络进行交互式的管理。该协议传输层可以使用 TCP、UDP 等协议作为传输层协议,但通常使用的是无连接的 UDP 协议。其主要组成元素:SNMP Manager 为管理者;SNMP Agent 为 SNMP 代理,运行于被管理对象上;MIB[4](Management Information Base,管理信息库)是指在因特网的网管框架中被管对象的集合,包括与主机资源相关信息。

SNMP 的 MIB 是通过管理信息结构定义的逻辑数据库。MIB 以树状分层结构组织和管理代理支持的各种标量或表格对象,每个对象都有全局唯一的标识,可以通过 OID(Object IDentifiers,对象标识符)查找或数据采集。

SNMP 定义了 5 种基本操作原语:

(1)GetRequest,取请求;

(2)GetNextRoute,取下一个请求;

(3)Set,设置请求;

(4)GetReply,取应答;

(5)Trap,陷阱。

对 MIB 中的数据进行采集、写入等操作可以通过这 5 种操作原语实现。

2 系统设计

2.1 设计目标

实时采集、监测异构网络应用服务器核心数据信息,主要包括:CPU 使用率、物理内存使用率、虚拟内存使用率、硬盘空间使用率、网络丢包率等。

系统具有通用、实时、安全、易部署的特点。通用性是指系统平台能够采集不同厂商生产的服务器以及不同种类的操作系统信息资源;实时性是指对采集的主机资源使用数据,以简洁、直观的图表形式显示,并对出现的异常状况通过声音、颜色、邮件、短信等实时地向管理员发送报警信息;同时,由于服务器主机高可靠、不间断运行的特点,要求系统具有较高的安全性和充分利用现有设置简化部署的特点。

2.2 设计思路

在异构网络环境中,由于存在多种不同类型和不同版本的操作系统,如果针对每种操作系统都开发一种监测工具,无论是前期工作量还是后期维护成本上皆不可取。基于上述原因,系统拟采用基于 SNMP 协议的数据采集和监测方法,理由在于:

(1)SNMP 是目前计算机网络中的标准网络管理协议,在网络管理和监控[5-7]等方面具有广泛的实用性;

(2)SNMP 具有非常强的扩展性和通用性,易部署,主流操作系统都有内置 SNMP 组件可供使用,甚至一些非网络设备如 UPS 电源和物联网设备均带有 SNMP 模块;

(3)SNMP 是一个应用层协议,传输层可以使用不同的协议,可以兼容多种网络;

(4)SNMP 最新版 SNMPv3 提供了全新的安全机制[8-9],使得系统的安全性大大提高。

以上分析不难发现,采用基于 SNMP 协议的数据采集和监测方法能够很好地满足系统的设计目标。

2.3 系统组成

系统 SNMP 部分采用 C/S 架构,有两种基本工作模式:

(1)网络管理器采用轮询的方法获取代理中的 MIB 库信息;

(2)网络管理器采用基于陷阱的方法接收代理发出的 Trap 报文。

系统的整体结构如图 1 所示。图中的监控系统服务器是整个体系结构的核心,它运行着监测系统程序,包括 SNMP Manager 模块、SNMP Trap Receiver 模块以及资源监控管理模块。SNMP 管理模块的功能是向被监测的主机主动采集数据,SNMP Trap Receiver 模块的功能是接收被监控主机发送的 SNMP Trap 信息并传送给资源监控管理模块,资源监控管理模块的功能是:主动调用 SNMP Manager 模块以固定时间间隔轮询所有被监测主机、处理 SNMP Trap Receiver 模块上报的 Trap 信息、根据获取的信息向邮件网关或短信网关发送报警信息、数据保存、数据备份、数据呈现等[10-11]。由于是异构的网络环境,被监控主机包含 Windows、Linux、UNIX、Solaris 等不同类型的操作系统,每个主机都需要包含一个 SNMP Agent 模块。SNMP Agent 模块负责维护本地所有的原始数据信息并响应 SNMP Manager 模块的 SNMP Get 请求。SNMP Trap Sender 模块是一个可选的模块,它的功能是监控本机的资源变化,在发生报警事件时主动向监控服务器发送 SNMP Trap。由于不同类型操作系统在本机采集资源信息的途径不同,因此需要针对每种操作系统编写独立的 SNMP Trap Sender 模块。SNMP Trap Sender 模块并不影响系统整体功能,只在实时性要求特别高的主机上

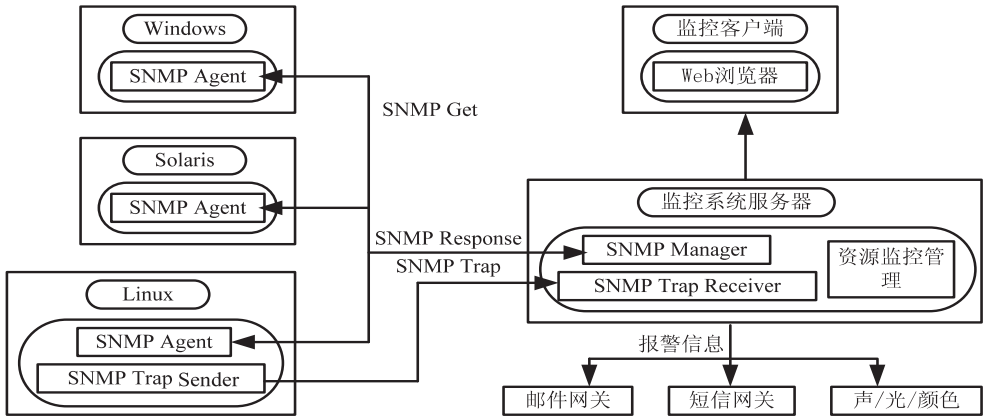

图 1　系统框架

部署。监控数据采用 B/S 架构呈现,监控客户端通过 Web 浏览器查看所有的监控数据,资源监控管理模块将实时数据以图形的方式呈现。

3　系统实现

3.1　开发平台

系统开发运行平台为 Redhat Linux,开发环境为 Linux C 和 Net-SNMP 开发包,后台数据库为 MySQL。

Net-SNMP[12]是一款历史悠久、基于 BSD 及 BSD Like 开源许可的跨平台 SNMP 软件,支持 Windows、Linux、Unix 等操作系统平台,它功能全面,具有高效、高可靠等特点,是多数 Linux、Unix 发行版默认的 SNMP 组件。Net-SNMP 支持 SNMPv1、SNMPv2c 和 SNMPv3,并可应用于 IPv4 及 IPv6 运行环境。Net-SNMP 提供了一个程序集合,包括可用于执行 SNMP 基本原语操作的命令行工具、图形化的 MIB 浏览器、接收 SNMP 陷阱的守护程序、集成了大量 SNMP 模块并响应 SNMP 原语操作的代理守护程序等。Net-SNMP 也提供了可以用于 SNMP 基本原语操作、SNMP 代理扩展以及 SNMP 陷阱的开发包,大大简化了 SNMP 网络管理软件的开发难度。

Net-SNMP 开发包用于开发数据采集模块,MySQL 用于进行数据存储及备份,使用 PHP 来开发 B/S 架构的数据呈现模块。

3.2　监测内容及技术实现

SNMP 服务使用的是 161 号 UDP 端口,系统所需的原始数据通过 SNMP 协议从被检测主机的 MIB 中获取。MIB 中反映主机资源占用情况的信息分别包含在不同的 MIB[13-14]文件中,下面将列出需要监测的原始数据所在的 MIB 文件以及相关资源占用率的计算方法。

CPU 使用率:反映当前主机的负载状况以及软件是否发生死锁等故障。Windows 的 CPU 使用率只有一个综合使用率,而 Linux、Unix 一般有系统 CPU 使用率和用户 CPU 使用率的差别。Windows CPU 使用率只需要通过 HOST-RESOURCES-MIB 中的 hrProcessorLoad 获取即可;Linux、Unix 的 CPU 使用率需要通过 UCD-SNMP-MIB 中的 ssCpuSystem 和 ssCpuUser 分别获取 CPU 的系统和用户使用率。对于多个 CPU 的情况,需要对所有 CPU 使用率取平均值进而得到 CPU 使用率。

存储使用率:包括物理内存、虚拟内存、硬盘空间使用率,反映当前主机所有存储的使用状况。HOST-RESOURCES-MIB 中的 hrStorageTable 包含存储的所有信息,hrStorageTable 以表的形式组织所有信息,其示例表项如表 1(仅包含需要使用的字段)所示:hrStorageIndex 为序号,hrStorageType 为存储类型,hrStorageAllocationUnits 为分配块大小,hrStorageSize 为存储大小(以块为单位),hrStorageUsed 为已使用存储大小(以块为单位)。不难发现,每种类型存储使用率的具体计算公式为:

$$storageUsage = \sum(hrStorageAllocationUnits * hrStorageUsed) / \sum(hrStorageAllocationUnits * hrStorageSize)$$

表 1　MIB 存储信息

hrStorageIndex	hrStorageType	hrStorageAllocationUnits	hrStorageSize	hrStorageUsed
1	hrStorageRam	1 024	379 488	353 152
3	hrStorageVirtualMemory	1 024	776 792	355 140
31	hrStorageFixedDisk	4 096	1 895 859	275 575
33	hrStorageFixedDisk	1 024	233 191	16 422

网络丢包率:主机负载过高就有可能出现丢包。网络丢包率可以分为输入丢包率和输出丢包率,相关数据可以从 IF-MIB 中获取,主要包括输入字节数 ifInOctets、输入丢失字节数 ifInDiscards、输出字节数 ifOutOctets、输出丢失字节数 ifOutDiscards。考虑到单个主机可能配备多块网卡,丢包率的计算公式为:

$$ifDrop = \sum ifDiscards / \sum ifOctets$$

通过以上资源占用信息可以从整体上判断主机资

源使用状况,如需更详细的信息或者其他资源使用状况,比如监控每个应用程序资源占用状况等,可以浏览 SNMP 的 MIB 库,将相关的数据和功能加入监测系统。

3.3 部分数据采集程序

```
int handler( )
{longPackage[4][SIZE];longWinCpuUsed[SIZE]; int StorageNum;intPackageNum;
int WinCpuNum; int i;
char * Target[ ] = {"hrStorageAllocationUnits",/ * 0,各分区簇的大小 * /
"hrStorageSize",/ * 1,各分区的大小,单位为簇 * /
"hrStorageUsed",/ * 2,各分区已用空间,单位为簇 * /
"ifInDiscards",/ * 3,输入丢包数量 * /
"ifOutDiscards",/ * 4,输出丢包数量 * /
"ifInOctets",/ * 5,输入包的数量 * /
"ifOutOctets",/ * 6,输出包的数量 * /
"hrProcessorLoad",/ * 7,Windows 的 CPU 占用率 * /
"ssCpuSystem",/ * 8,Linux CPU 的系统用量 * /
"ssCpuUser",/ * 9,Linux CPU 的程序用量 * /
"hrStorageType"/ * 10,分区类型 * /};
double TotalStorageSize;double TotalStorageUsed;double TotalOctets;double TotalDiscards;
/ * 这个结构体存储所有结果 * /
Data data;
int win_cpu_per;int lin_usr_cpu_per;int lin_sys_cpu_per;float net_per;
FILE * fp;
char line[81];char * position;
char address[25];/ * 被监控服务器地址 * /
char name[25];/ * 服务器名 * /
char os[15];/ * 操作系统类型 * /
char log[200];/ * 日志内容 * /
if((fp = fopen("config.txt","r")) = =NULL)
{fprintf(stderr,"Can't find the config.txt\n");exit(-1);}
while(! feof(fp) && ! ferror(fp))
{init_data(&data);init_storage_data( );fgets(line, 81, fp);
if(line[0] = = '#' || isspace(line[0]))/ * 忽略注释行和首字母为空的行 * /
continue;
sprintf(log,"Start Scan: %s",line);write_log(log);
/ * 获取主机名,地址和操作系统类型 * /
position=getword(line,name);position=getword(position,address);getword(position,os);
/ * 获取 CPU 的使用情况 * /
if(! strcmp(os,"windows"))
{WinCpuNum = walk(address,Target[7],WinCpuUsed);
win_cpu_per = 0;
for(i=0;i<WinCpuNum;i++) win_cpu_per=win_cpu_per+WinCpuUsed[i];
if(WinCpuNum! =0)
{win_cpu_per=win_cpu_per/WinCpuNum;
#ifdef DEBUG
sprintf(log," Calculate WinCpu successfully"); write_log(log);
#endif}
data.win_cpu_per=win_cpu_per;
}
else
{walk(address,Target[8],(long * )(&lin_sys_cpu_per));
walk(address,Target[9],(long * )(&lin_usr_cpu_per));
data.lin_sys_cpu_per=lin_sys_cpu_per;data.lin_usr_cpu_per=lin_usr_cpu_per;
#ifdef DEBUG
sprintf(log," Calculate Linux Cpu successfully"); write_log(log);
#endif}
/ * 计算硬盘,物理内存,虚拟内存的使用情况 * /
calculate_storage(&data.sto_per,&data.mem_per,&data.vir_per);
#ifdef DEBUG
sprintf(log,"Calculate Storage successfully");write_log(log);
#endif
}
```

3.4 数据呈现及备份

系统采集的数据以结构化的方式存储在 MySQL 数据库中,数据呈现采用现在非常流行的 B/S 架构,网络管理员可以随时随地查看实时监测数据。数据呈现模块采用 PHP 开发,将各个主机的各种资源占用情况以二维表的形式绘制出来,通过设置不同时间粒度,可以呈现最近几个小时、天、周、月、年的资源使用曲线。

数据备份通过 MySQL 提供的备份功能实现。使用 MySQL 可以方便实现数据的完整备份、增量备份以及数据的导入和导出,可以非常方便地进行系统数据的备份和恢复工作。

4 系统测试

华中科技大学网络中心是一个典型的异构操作系统网络,包括 Windows、Linux、Unix 和 Solaris 等多个操作系统版本和多种不同的网络应用环境。经过 WEB、DNS、VPN、BBS、EMAIL、FTP 等网络应用服务器上实际部署、运行,实时采集、监控相关网络主机数据资源信息,并在出现故障时发出报警提示,及时解决了网络故障的发现问题、缩短了故障恢复时间、提高了网络服务质量,有着显著的效果和效益。图 2 是 BBS 和 Squid 主机当前的资源使用情况。

图 3 分别是 BBS 主机在一天、一周、一月、一年的时间内的 CPU 使用率曲线,横向坐标表示时间轴,纵向坐标表示使用率,图中上端的曲线和下端的曲线分

别表示用户 CPU 使用率和系统 CPU 使用率。

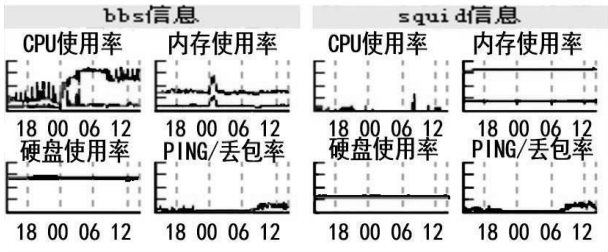

图 2　单个主机资源使用率

图 3　BBS 主机 CPU 使用率

5　结束语

网络应用主机监控是保障网络应用服务的重要手段,监控中的数据采集方法是确保监控效能的关键环节。文中基于一个实际运行的校园网络异构系统平台,采用 SNMP 协议进行数据采集,继而进行分类处理,汇总归纳成图表信息,并根据故障点给出报警提示和短信、Email 讯息,实则为 SNMP 在异构系统平台上进行数据采集和处理的一次很好实践,无疑对网络监控和稳定运行具有很好的示范意义和应用前景。

参考文献:

[1] A Simple Network Management Protocol (SNMP)[S]. RFC 1098,1989.

[2] Introduction tocommunity-based SNMPv2[S]. RFC 1901, 1996.

[3] Introduction to version 3 of the internet-standard network management framework[S]. RFC 2570,1999.

[4] Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)[S]. RFC 3418,2002.

[5] 胡谷雨. 简单网络管理协议教程[M]. 第 2 版. 北京:电子工业出版社,1999.

[6] 叶　春,须自明. 基于 SNMP 协议的分布式网络监控系统的设计实现[J]. 计算机时代,2009(10):30-32.

[7] 柯栋梁,万　燕. 基于 SNMP 协议的流量监测系统的设计与实现[J]. 微计算机信息,2006(3):117-119.

[8] 赵晓峰. SNMP 协议的安全性分析及其攻击与防御[J]. 微电子学与计算机,2006,23(12):132-134.

[9] 包达志. SNMP 协议安全性分析[J]. 计算机安全,2011(6):58-60.

[10] 魏煜欣,李　强. 一种基于 SNMP 网络性能管理数据的采集方法[J]. 计算机工程与应用,2011,47(2):105-107.

[11] 李明江. SNMP 简单网络管理协议[M]. 北京:电子工业出版社,2007.

[12] Net-SNMP[EB/OL]. 2013-02. http://www.net-snmp.org.

[13] 刘雪飞. 基于 SNMP++的 MIB 浏览器研究[J]. 计算机工程与应用,2009,45(3):91-93.

[14] 朱创录. SNMP 网络管理中高效轮询方法研究[J]. 计算机技术与发展,2012,22(12):135-138.

(上接第 86 页)

[2] 杨晓光. 基于 ITS 的高速公路紧急救援管理系统研究[J]. 上海公路,2002(1):4-8.

[3] Liu H X,Wu X K,Ma W T,et al. Real-time queue length estimation for congested signalize intersection[J]. Transportation Research Part C:Emerging Technology,2009,17(4):412-427.

[4] Morales M J. Analytical procedures for estimating freeway traffic congestion[J]. Public Road,1986,50(2):55-61.

[5] Sheu J B,Chou Y H,Chen A. Stochastic modeling and real-time prediction of incident effects on surface street traffic congestion[J]. Applied Mathematical Modeling,2004,28(5):445-468.

[6] 高朝晖,陈里得,黄　卫. 高速公路事件检测与管理系统研究[J]. 交通与计算机,2003,21(5):10-13.

[7] 刘伟铭,管丽萍,尹湘源. 基于决策树的高速公路事件持续时间预测[J]. 中国公路学报,2005,18(1):99-103.

[8] 姬杨蓓蓓,张小宁,孙立军. 交通事件持续时间预测及参数标定[J]. 重庆交通大学学报:自然科学版,2010,29(4):613-615.

[9] 王殿海,金　盛. 车辆跟驰行为建模的回顾与展望[J]. 中国公路学报,2012,25(1):115-127.

[10] 王殿海. 交通流理论[M]. 北京:人民交通出版社,2002.

[11] 丛浩哲,王俊骅,童世鑫. 高速公路网络交通突发事件辐射范围预测模型[J]. 同济大学学报(自然科学版),2012,40(3):414-422.

[12] Greenshields B D,Bibbins J R,Channing W S,et al. A study of traffic capacity[J]. Highway Research Board Proceedings,1934,14(1):448-477.

异构网络中SNMP数据采集监测系统研究

作者： 杨勇， 李军， 张越培， YANG Yong， LI Jun， ZHANG Yue-pei

作者单位： 杨勇,YANG Yong(华中科技大学 网络与计算中心,湖北 武汉,430074)， 李军,LI Jun(华北水利水电大学 计算机学院,河南 郑州,450045)， 张越培,ZHANG Yue-pei(河南卫生职工学院 信息中心,河南 郑州,451191)

刊名： 计算机技术与发展 ISTIC

英文刊名： Computer Technology and Development

年，卷(期)： 2014(6)