

基于 Web 的分布式网络实时监控系统的设计与实现^{*}

杨家海¹ 吴建平¹ 李钟辉²

¹(清华大学计算机科学与技术系 北京 100084)

²(中国教育和科研计算机网络中心 清华大学 北京 100084)

摘要 随着大规模网络的建设和应用,网络管理问题变得越来越重要.由于传统的集中式的网络管理模式已越来越难以胜任这项工作,人们及时地提出了分布式网络管理的概念.该文论述了一个基于 Web 的分布式网络实时监控系统(WebRMonitor)的总体结构及系统实现的关键技术;数据采集模块的设计与实现和实时数据的可视化表示等.

关键词 计算机网络,网络管理,客户/服务器结构,Web 服务器/浏览器,公共网关接口 CGI,Java.

中图法分类号 TP393

随着计算机及通信技术的飞速发展,计算机网络已经渗透到社会经济的各个领域,对社会经济发展起着越来越重要的作用,也使人们的工作甚至生活方式发生巨大的变革.今天人们对网络的依赖就像过去人们对电话的依赖一样,哪怕几分钟的中断也是难以容忍的.与此同时,随着网络规模的扩大,影响网络服务的因素也增多了,如网络设备(交换机、MODEM、路由器以及通信线路等)的失效问题、主机和终端的故障问题、软件中出现的问题、安全保密问题等等.因此,网络管理的复杂度也大大增加.如何进行有效的网络管理,确保信息网络的可靠而稳定的运行已经成为一个迫切需要解决的问题^[1].

在以开放标准为基础的网络互连领域中,网络管理也应通过标准化来实施^[2].国际标准化组织 ISO 和 Internet 组织在这方面都做了大量的工作,并提出了各自的协议标准和服务,如 ISO 的公共管理信息服务及协议 CMIS/CMIP^[3,4],Internet 的简单网络管理协议 SNMP v1/v2^[5,6].

由于历史和客观的原因,ISO 的 CMIS/CMIP 虽然定义得很完备,但却没有得到工业界的广泛支持.而 Internet 的 SNMP 一经提出就得到工业界的广泛响应,许多公司相继推出了基于 SNMP 协议标准的网络管理系统和产品,如 Sun 公司的 SunNet Manager,HP 的 OpenView,IBM 的 NetView/6000 等等^[7].这些产品的推出,一度缓解了网络管理的矛盾.

然而,在早期所开发出的网络管理系统中,大都是基于单一主机的 Manager-Agent 模式,由运行网络管理软件的 Manager 从各 Agent 收集网管信息,在本地完成对信息的分拣和处理工作,网络各项参数的绘制与显示亦由网管主机完成,这是一种高度集中的网络管理模式.这势必会造成网管主机的超负荷运转,极大地降低网络管理系统的运行效率.而且相关参数的绘制与显示也要依赖于相应的网管平台,没有一个统一而高效的监控界面,给网络管理员的操作和使用带来了一定的困难.

因此,人们及时引入了分布式网络管理的概念.这不仅体现在低端网管信息的收集上,还表现在用户端的监控和管理界面上^[8].Web/Java 技术的快速发展和广泛应用使得分布式网络管理从概念变成了现实^[9].近期推出的一些新型网络管理系统,如 Advent 公司的 NetMonitor 等^[10],将用户界面直接实现于当今最为流行的 Web 浏览器之上,使用户可以直接通过 Web 浏览器来完成对网络的管理和监控;一些传统的网络管理产品供应商,如

* 本文研究得到国家“九五”科技攻关项目基金资助.作者杨家海,1966年生,副教授,主要研究领域为网络协议工程学,网络体系结构,网络管理及应用.吴建平,1953年生,教授,博士生导师,主要研究领域为网络协议工程学,网络体系结构,网络管理.李钟辉,1974年生,硕士生,主要研究领域为网络管理.

本文通讯联系人:杨家海,北京 100084,清华大学计算机科学与技术系

本文 1998-01-13 收到原稿,1998-04-15 收到修改稿

HP, IBM 等也都在其新版的网管产品中加入了 Web 的支持^[1], 这无疑是在统一管理界面和分布式管理上进行的一种成功的尝试. 它使用户不必再重新学习复杂的界面操作方法, 只需像浏览 WWW 一样, 方便而轻松地完成网络管理工作, 也不必局限于某一特定平台, 最大限度地实现了跨平台管理的功能.

当然, 基于 WEB 的网络管理刚刚开始, 一些厂商提供的产品其功能也都有限. 尤其是我国目前正在进行一系列全国性、行业性计算机网络的建设和, 购买了大量传统集中式的网络管理平台. 为了自主扩充现有功能, 也为了保护原有的投资, 我们在国家“九五”攻关项目的资助下进行了利用现有网管平台的、基于 WEB 的分布式网络管理方面的研究. 本文讨论了一个基于 WEB 的分布式网络实时监控系统(WebRMonitor)的总体结构及系统实现的关键技术, 数据采集模块的设计与实现和实时数据的可视化表示等.

1 WebRMonitor 的总体结构

本监控系统以 IBM NetView/6000 网络管理系统(运行在 IBM RS/6000 工作站上)为基础, 利用其收集的网管信息, 在客户端的 Web 浏览器上实现对网络流量及其他信息的实时监控, 并能提供与 NetView/6000 系统较为相似的操作界面, 以方便网络管理员的使用. 系统实现的目标是尽可能最大限度地实现分布式监控模式, 同时还要尽量减轻网管主机的负担, 以提高网络管理效率.

根据上述要求, 结合现有网络管理系统的特点, 可以将所要设计的流量监控系统分为 3 大模块: 流量数据获取模块、统计图形绘制模块和绘制特性控制模块. 其基本的设计思想是在满足系统实现的目标的前提下, 降低由管理而引起的额外带宽开销. 基于客户/服务器模式的通信机制为实现上述设计思想提供了可能, 它既能利用 NetView/6000 提供的先进网络管理功能特性, 又能利用 Web 的强大图形功能. 在服务器方, 只由网管服务器 NetView/6000 统一采集实时数据, 服务器进程只接收由网管服务器采集的数据, 并进行数据格式的分析转换和传送工作; 网管主机上的实时数据传送到客户端以后, 由客户端的 Web 浏览器根据相应数据完成实时图形的绘制, 并实现对图形特性的本地控制. 这样就可以使网管主机从同时要绘制大量实时曲线的沉重负担中解脱出来, 而把主要精力用于数据的采集和故障的发现等工作上, 这将大大提高网管服务器的运行效率.

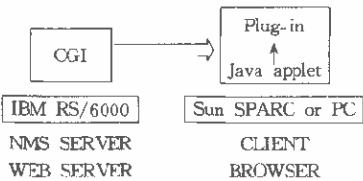


图1 WebRMonitor的总体结构

WebRMonitor 的总体实现结构如图 1 所示.

(1) WebRMonitor 系统以 Client/Server 模式工作, 客户与服务器之间的通信采用 HTTP 协议进行, 服务器端程序负责实时数据的采集与格式的转换等工作, 客户端程序负责实时曲线图形的绘制、由线特性的控制及其他一些附加功能的实现.

(2) 由于 Plug-in 具有实时运行的高效性, 并且可以存取客户端的本地文件, 因此决定用 Plug-in 来实现统计图形绘制功能模块; 利用 Java applet 对多线程的支持及其平台无关性, 将使用 Java applet 来实现绘制特性控制功能模块.

(3) Web Server 和网管服务器运行在同一台主机上, 在服务器端, 通过标准的 CGI 接口实现对由网络管理服务器收集的网络实时数据的存取、访问与传送. 其文档类型为客户端 Plug-in 在浏览器中登记的 MIME 类型.

(4) 在客户端, 以 Plug-in 方式完成流量数据的读取与存储, 并完成实时数据的图形绘制工作, 同时, 为了与 NetView/6000 网络管理系统的相应功能一致, 还提供对绘图区内相应鼠标事件的处理功能(指示流量数据对应的时间).

(5) 在客户端, 由 Java applet 完成图形绘制特性的实时动态控制, 并可直接调用 Plug-in 内实现的本地方法进行图形更新, 同时还建立一个线程用以定时调用 Plug-in 内实现的另一本地方法, 从而完成对 Server 端 CGI 的调用, 以实时获取流量数据.

(6) 考虑到一般的网络都需同时监控多个被管设备和多条通信线路, 因此, 界面的设计还应该考虑到可根据用户需要进行定制的功能. 作为附加部分, 本系统也实现了这样的功能.

2 实时数据采集

实时数据采集是 WebRMonitor 系统的基础, 如前所述, 该功能在服务器一端(也是网管服务器). 实时数据

采集需要解决两个问题:一个是如何将数据从被管对象系统收集到网管服务器上来,另一个是网管服务器收集的数据如何传递给 Web 服务器。

2.1 数据采集

网管服务器和被管对象之间的通信利用标准 SNMP 协议进行,因此,解决第 1 个问题有两种方法可以选择,一个方法是利用 NetView/6000 系统已经实现的功能来进行,另一个方法是自己实现一套 SNMP 协议。由于重新实现一套 SNMP 协议的工作量较大,同时也考虑到 NetView/6000 系统在许多方面还是很优秀的,因此,我们决定利用 NetView/6000 已有的功能来实现实时数据的收集。

当然,这种选择在一定程度上增加了解决第 2 个问题的难度,NetView/6000 的 XnmGraph 绘图命令可以直接接收 SnmpCollect 实时收集的动态数据(二进制格式)来完成绘图工作,而由于 NetView/6000 是 IBM 公司的商品化软件,所以未公开这些相应的数据接口。为了利用 NetView/6000 收集的数据来实现实时数据的分布式图形显示,首先必须分析这些数据的结构和格式。

通过大量查阅 NetView/6000 的相关技术手册和大量的实验,我们清楚了实时数据的格式,并实现了随机地将二进制数据转换成 ASCII 格式数据的工作,这为后面将实时数据传递给 Web 服务器,并由 Web 服务器传送到客户端进行显示与绘图提供了保证。

2.2 数据处理与传送

标准 CGI 接口为外部数据接入 Web 服务器提供了标准接口。CGI 编程的核心是接口参数的确定及处理。由于从网管服务器收集的数据先暂存在一个文件中,因此需设置一个文件名参数 filename。另外,为了能绘制任意一点的曲线,还需设置一个位置参数 pos,用于指示上次的文件读取位置。除此之外,为了解决分布式数据存储的一致性问题(即保证 Client/Server 两端的流量数据文件的一致性),在每次客户端的流量监控系统初启时,若本地数据文件存在(本地数据文件的长久保存是为了减轻系统初启时网络的传输量),则还应向 CGI 附加一个 test 参数作为一致性测试标志,CGI 由此读取相应历史文件的第 1 组数据的结束时间并传回客户端。这样,客户端将传回的时间值与本地文件中的第 1 个时间数据相比较,若两者相等,则说明两端的历史数据一致,可只传送新数据;若两者不相等,则说明 Server 端(网管主机)的历史数据文件已发生了变动(例如,删除过历史数据文件,造成文件的重新建立,第 1 组的结束时间即发生变化),此时就要重传整个历史数据文件,从而保证了 Client/Server 两端流量数据的一致性。

在数据的处理上,由于同时存在测试与读取出入口流量文件共 4 种数据类型,故应在传输数据的头部加入 1 字节的标识位来指示后续数据的类型,以保证客户端在网络传输异步性的情况下,可以通过该标识位确定数据类型,正确完成数据的读取与处理工作。具体定义是:“0”与“1”分别表示入/出口测试数据,后面只有 8 字节的十六进制 ASCII 码字串(是由二进制长整型转换而来的),代表相应文件的第 1 组结束时间;“2”与“3”分别用于说明后续数据为入/出口流量数据,其后的第 1 组 8 字节十六进制 ASCII 字串是此次读取相应历史数据文件的最后位置,以便客户端进行记录,供下次 CGI 调用中 pos 参数的使用,随后即是以 16 字节为单位的数据流。

CGI 程序由 cgi 提供的 cgiHeaderContentType 库函数来指定传输文档的 MIME 类型,在此,特标志为“application/x-netmon-plugin”的新的 MIME 类型,以便客户端已向浏览器登记此类型的 Plug-in 进行读取和处理,从而完成本地数据文件的记录和统计图形的绘制工作。数据处理与传送模块的工作流程如图 2 所示。

3 实时数据的图形化显示

在 Web 浏览器中实现对一般图形的绘制并不困难,Java 技术或 Plug-ins 技术都可以实现,这里的关键问题是要体现实时性、效率和灵活性,即如何定时刷新实时曲线,并能灵活、有效地对图形进行控制。为了实时刷新数据,我们在客户端实现了一个定时发送查询信息的进程(或线程),即由客户端而不是服务器端来实现数据的更新,在定时间隔的选取上,应与网管服务器采集数据的间隔保持同步。

考虑到 Plug-in 具有实时运行的高效性,并且可以存取客户端的本地文件,故而决定用 Plug-in 来实现统计图形绘制功能模块,利用 Java applet 对多线程的支持及其平台无关性,使用 Java applet 来实现绘制特性控制功

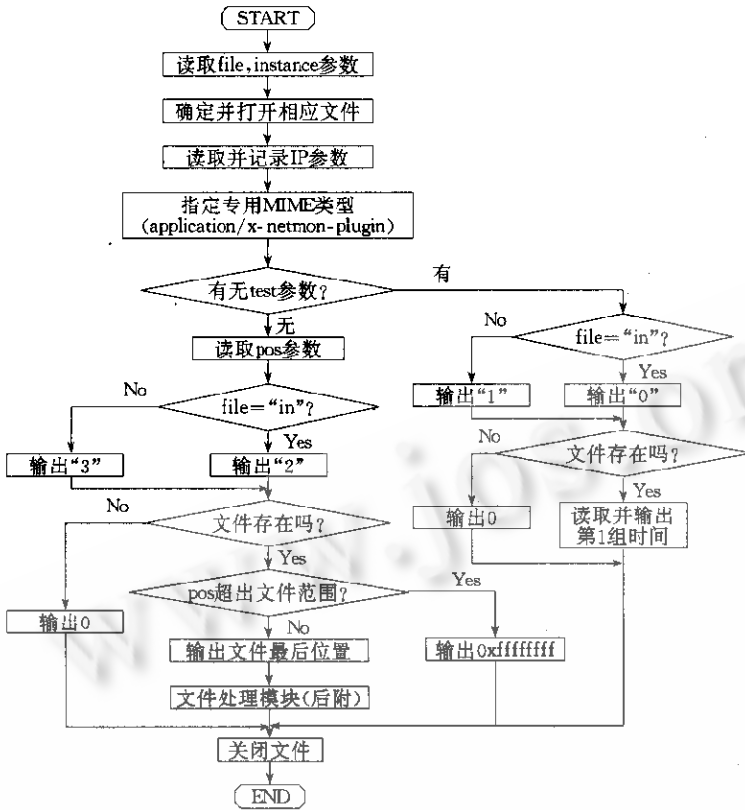


图2 实时数据的处理与传送

能模块。但这就要涉及到 Java 与 Plug in 的接口问题,所幸的是 Netscape 公司开发了 Live Connect 技术,利用该技术可以实现同一主页内的 Java, Plug-in 和 Javascript 对象之间的相互调用,使三者有机地结合在一起,从而实现更为强大的功能。通过详细阅读 Live Connect 的技术资料,我们发现,它正好适合流量监控系统客户端两大功能模块设计方案的技术要求,而且 Netscape 公司还在其网址上提供了一套 Live Connect/Plug-in SDK 软件开发环境,因此,我们决定以 Live Connect 技术为核心来构造流量监控系统在客户端的两大功能模块。

Live Connect 技术是用于实现同一主页内 Java, Plug-in 以及 Javascript 各对象之间的联系和调用,其核心部分为 Java 类,它可以直接与主页内的 Plug-in 和 Javascript 对象进行

相互调用,而 Plug-in 与 Javascript 之间的关联则是通过 Java 类间接实现的。Java 类与 Plug-in 相互调用的实现手段是将主页内的 Plug-in 封装成一个新的 Java 类,即 netscape. plugin. Plugin 类,它是从 java. lang. Object 类派生而成的,由 Netscape 提供的 Java Package 包含。这样,Java 类就可以通过对 Plugin 类的操作来完成对 Plug-in 中实现的本地方法的调用,而 Plug-in 也可以通过 JRI(Java Runtime Interface, Java 运行时间库接口)来实现对 Java 类相应域的存取及其方法的调用。

Java applet 仅提供控制界面和定时线程,它将用户的设定值直接写入 Plugin 类之中的相应域(即成员变量),同时通过调用 Plugin 类的本地方法完成相应的控制功能。Java applet 是通过 Netscape 公司所提供的 Java Package 中包含的另一个新的 Java 类 netscape. javascript. JSObject 里的一些方法来实现与同一主页内 Javascript 对象的联络(JSObject 类是 Netscape 对主页内的 Javascript 对象进行封装而得来的)。

对 Server 端 CGI 的调用以及本地数据文件的生成与更新都是由 Plug-in 来完成的,为尽可能减少系统初启时的网络数据传输量,客户端本地文件将长久保存并允许多实例共享(通过建立和清除共享标志文件,实现多个相同监控实例共享同一本地数据文件,从而减少重复的流量数据传输)。为了保证客户端本地数据文件的有效性,在系统初启时将检测 Client/Server 两端数据文件的一致性,若一致,则仅需传送客户端本地数据文件所没有的新数据,否则,将删除客户端本地数据文件,全部进行重传。由于网络数据传输所存在的异步性,为 Plug-in 的数据读取处理加入了一个文件锁,以防止发送数据乱序现象的出现。此外,在 Plugin 类中也添加了一个互斥锁,用以实现 Java applet 方法与 Plug-in 本地代码对关键参数存取的互斥性,保证了相关数据的一致性。

4 结束语

目前,WebRMonitor 系统的开发和联调已经完成,系统已交由 CERNET 网络中心测试运行。

由于系统 WebRMonitor 使用了 Netscape 公司所独有的 Live Connect 技术,所以,目前只支持 Netscape Navigator 3.0 (及更高版本)的浏览器。而且,由于网络中心目前大量使用的是运行 Solaris UNIX 操作系统的工作站,因此客户端的流量监控系统平台暂定为 Solaris 的 UNIX 系统平台。而 Server 端(网管主机)的操作系统是运行于 IBM RS/6000 上的 AIX 3.2,因而 Web 服务器及相应的 CGI 程序则是基于 AIX 3.2 系统平台来实现的。

在系统的扩充和完善方面还有许多工作要做,我们准备在以下两个方面进一步完善系统。一方面是支持更多的客户端系统平台,如广泛使用的 Windows 95 及 Windows NT 等,以期使本系统具有更广泛的应用范围和环境。另一方面是增加监控的内容。目前的实时监控主要集中在几条主干网线路的实时的进出流量,实际上,用户想监视的不仅仅是这些内容,用户也许还想知道某些特定线路的丢包率、出错率;关键网络互连设备的运行状况,如路由器 CPU 的利用率、内存的利用率、设备环境温度等等。由于系统采用了分布式客户/服务器结构及模块化设计,为系统的扩展带来了极大的方便。比如,为了监控路由器 CPU 的利用率,只需在服务器数据采集方改变采集对象即可。目前我们已经着手上述工作,以期尽快使本系统实用化和通用化。

参考文献

- 1 ISO/OSI. Basic Reference Model for Open System Interconnection. IS7498, Oct. 1984
- 2 Lynch Daniel C, Rose Marshall T. Internet System Handbook. Reading, MA: Addison-Wesley Publishing Company, Inc., 1993
- 3 ISO/OSI. Management Information Service Definition, Part 2: Common Management Information Service. IS9595-2, ISO, Switzerland, May 1988
- 4 ISO/OSI. Management Information Protocol Definition, Part 2: Common Management Information Protocol. IS9596-2, ISO, Switzerland, May 1988
- 5 Case Jeffrey D, Fedor Mark *et al.* A Simple Network Management Protocol(SNMP). RFC1157, Internet Network Working Group, May 1990
- 6 Galvin James M, McCloghrie Keith. Administrative Model for Version 2 of the Simple Network Management Protocol (SNMPv2). RFC1445, Internet Network Working Group, Internet Engineering Task Force, Apr. 1993. ftp://nic.ddn.mil
- 7 IBM Corporation. NetView for AIX Administrator's Reference Version 3. Sep. 1994
- 8 Gottfried Luderer, Hosoon Ku *et al.* Network Management Agents Supported by a Java Environment. In: International Symposium on Integrated Network Management (ISINM'97). San Diego, CA, 1997. http://www.eas.asu.edu/~netsys
- 9 Sun Microsystems. The Java Language Environment—A White Paper. http://www.sun.com
- 10 Advent Network Management, Inc.. Delivering Web Based Network Management Tools. http://www.adventnet.com
- 11 HP Corporation. HP Openview Bridging the Gap. http://www.hp.com/openview

Design and Implementation of a Web-based Distributed Network Monitoring System

YANG Jia-hai¹ WU Jian-ping¹ LI Zhong-hui²

¹(Department of Computer Science and Technology Tsinghua University Beijing 100084)

²(Center of China Education and Research Network Tsinghua University Beijing 100084)

Abstract With the construction and application of lots of large-scale internetworks, it becomes more and more important to manage these internetworks. Since the conventional centralized network management mode becomes less competent, a distributed network management architecture is proposed in this paper. The structure of a Web-based distributed network realtime monitoring system—WebRMonitor is discussed, and its key implementing techniques are given, such as the gathering, decentralized presentation of the realtime data and so on.

Key words Computer network, network management, client/server architecture, Web server/browser, CGI (common gateway interface), Java.