

智能化网络管理工具的设计研究

王 瑛

(上海中医药大学 现教中心,上海 201203)

摘要:文中主要研究在当前信息化革命背景下的高效网络管理工具设计。传统的网络管理工具中存在过分依赖预先配置信息,导致维护工作量巨大以及学习理解过于复杂的不足。针对这些问题,文中提出了基于端口扫描和工厂模式的智能化检测方法。并采用这种检测方法设计出一种智能化的网络管理工具。这种工具基于开源平台开发,采用面向对象的设计技术,面对日新月异的网络环境有更好的维护性和扩展空间,明显降低了网络管理人员的工作强度,提高了网络管理的工作效率。

关键词:网络管理;端口扫描;工厂模式;面向对象

中图分类号:TP31

文献标识码:A

文章编号:1673-629X(2014)08-0063-04

doi:10.3969/j.issn.1673-629X.2014.08.015

Research on Design of an Intelligent Network Management Tool

WANG Ying

(Modern Education Technology Center, Shanghai University of Traditional Chinese
Medicine, Shanghai 201203, China)

Abstract: Mainly research on an efficient network management tool design in the present background of informatization revolution in this paper. The traditional network management tool has some deficiencies about large maintenance work and exceedingly complex learning comprehension caused by depending on configuration information greatly. Aiming at these problem, a new smart detection method based on port scanning and factory pattern is put forward. The new way is used to design an intelligent network management tool. This tool is development based on open source platform, using technology oriented object, with better maintenance and extension space for the changeable network environment, which can reduce the administrators' working hours and improve efficiency of network management work.

Key words: network management; port scanning; factory pattern; object oriented

0 引言

随着信息化革命的到来,数据中心中的新技术新设备不断地涌现,使得网络管理的难度不断增加。很难通过简陋的手段获得对网络环境直观的认识。传统的基于 Excel 表格、Word 文档记录静态数据,用 Telnet 远程命令获取动态数据的管理方式已经很难跟上信息技术日新月异发展背景下的管理需要。虽然厂商通常会随产品提供一些方便客户日常管理使用的软件工具,但是仍然不能解决众多不同设备高效统一管理的问题^[1]。为了解决在随时变化的复杂网络环境中进行有效管理的问题,需要引入新的解决思路和设计方法。目前常规的一些管理工具和软件往往存在功能比较单一、适用面窄的问题。即使有一些软件工具能够支持

较多类型的设备,也往往存在部署配置复杂,成本高昂的问题^[2-4]。

文中针对现实环境下网络管理的需求,以及现有的网络管理软件工具中存在的一些不足,提出了一种新的网络管理工具设计方法。

1 工具的整体架构设计

1.1 开发平台与语言的选择

为了降低开发成本,应当尽量选择开源平台及工具。文中研究设计的网络管理工具选择 Redhat Linux 操作系统作为运行平台。Redhat 是目前主流的开源系统之一,功能强大,稳定可靠。开发语言选择 Perl。Perl 是美国人 Larry. Wall 于 1987 年发布的,遵守 GNU

收稿日期:2013-10-28

修回日期:2014-01-24

网络出版时间:2014-05-21

基金项目:上海市高校选拔培养优秀青年教师科研专项基金项目(2011JW30)

作者简介:王 瑛(1985-),女,上海人,硕士,工程师,研究方向为软件工程、计算机应用技术。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20140525.1242.021.html>

通用公共许可证。Perl 内置了正则表达式的功能,使得处理报表工作变得容易。后台数据库采用 IBM 的 DB2 Express-C。这是 DB2 数据库系列中的一款免费产品,提供最基本的常规功能,省去其中的一些高级特性,能够满足小型工具的开发需要,设计更加紧凑,运行更加高效。

1.2 对象模型的设计

Perl 语言本身并没有面向对象的特性,为了设计出比较结构清晰的对象模型,本管理工具采用 CPAN (Perl 综合程序库)提供的第三方程序模块 Object::InsideOut 作为支持面向对象特性的框架。

1.2.1 设备抽象层设计

考虑到需要提供一个高效统一的管理工具,那么就需要将过去分散孤立的脚本和软件提供的功能整合在一起。整合并不是简单地将原有功能罗列在一起,而应该是有机结合,结构清晰,设计风格统一,能够便于复用。

根据对当前数据中心内网络设备管理的需求分析,设备抽象层的类大体可以分为主机、交换机、存储三大类^[5-6]。

再具体细分,主机类又可以根据其操作系统的不同,分为 Windows, Redhat, Suse, Unix 等等;交换机类根据厂商产品系列的不同可以分为 Brocade, Cisco, Qlogic 等等;存储也可分为 EMC 的 Symetrics, vMax, vPlex, IBM 的 DS8000, XIV, SVC 等^[7-8]。无论封装实现的是什么设备,无论底层有多么大的不同,为了便于上层集成调用,应当对上层提供一致的接口。

从上层来看,无论底层封装的是什么设备,对上层无外乎两个操作,信息收集与数据录入。因此,所有的设备类都需要实现两个方法,collect 与 import_to_database^[9-10]。

collect 方法实现从设备收集信息的逻辑,而 import_to_database 实现向数据库录入的逻辑。

1.2.2 协议适配层设计

由于文中设计的是一个集中统一网络管理的工具,必须通过一些网络协议来远程操控设备。常见的网络管理协议有 SSH, Telnet, Winexe 等。现实中,不同设备可能共用同一种协议,同一种设备可能在不同协议之间切换的现状,也就是说,特定协议与特定类型的设备之间是没有必然联系的^[11]。这就决定了网络协议的逻辑不能在设备抽象层内实现,而应该独立出来,单独设计一个协议适配层。设计良好的协议适配层类能够在不同设备之间轻松调换使用,提供了极大的开发灵活性^[12]。

对于一个网络协议,常见的用户场景就是,建立连接,发送命令并接收返回的文本,关闭连接。所以,一

个协议适配层类应当实现三个方法,open,cmd,close。open 方法用于建立连接;cmd 方法用于发送命令并接收文本;close 方法用于关闭连接。在设备抽象类的对象实例化的时候,只需要通过目标设备的对象的属性将协议对象动态传递进去即可。由于协议适配层的类具有相同的接口,更换协议对设备抽象层的类来说是完全透明的。

1.2.3 自动识别模块设计

管理众多的设备的静态登记信息,如果只是依赖管理员的记忆力,或者是 Excel 表格,费时费力,结果也并不可靠^[13]。首先要设计一个能够从网络当中自动抓取的活动 IP,并识别出运行在该 IP 地址上的设备类型的模块。

当活动 IP 地址被从一个子网中扫描出来以后,进一步识别设备的类型可以分为网络扫描和远程命令试探两个阶段。

第一阶段的网络扫描主要是为了下一步的命令试探缩小范围。无论是主机、交换机,还是存储设备,都会对外打开一些固定的网络服务,这些网络服务通常都监听特定的端口,通过扫描端口可以大致了解设备打开了哪些网络服务,进一步可以猜测设备可能对应的类型^[14]。在扫描端口之后,进入第二阶段,发送前期锁定的几种可能的设备类型上支持的命令,再通过 Perl 语言内置的正则表达式搜索特征关键字。如果成功捕获到了响应的关键字,就成功地识别了设备类型。类型识别成功后,才能调用响应类型的设备抽象类,对设备进行信息收集与管理。

2 自动识别模块的具体实现

2.1 网络扫描部分

在第一阶段的网络扫描中,首先由 Subnet 类负责活动 IP 的获取,之后将活动的 IP 地址实例化为 IP 类的实例,之后 IP 类会扫描该 IP 地址上的哪些端口已经被监听了。这两个类依赖于 Nmap 提供底层功能的支持。Nmap 是一款用于网络安全审计的工具,可以列举主机,监控网络服务运行情况。Nmap 不仅可以扫描单个 IP 地址,还可以对整个网络进行大规模扫描。同时,Nmap 也是一款遵守 GNU 通用公共许可协议的软件,可以免费地在文中要设计的工具中集成使用。

Subnet 类和 IP 类的接口如图 1 和图 2 所示。

Mgt. Network. Subnet
ip
mask
unicast_ips

图 1 Subnet 类的接口

Mgt. Network. IP
address
opened_ports
refresh_opened_ports()
is_port_opened(int port)

图 2 IP 类的接口

Subnet 类在获取活动 IP 地址的过程中需要使用 Nmap 提供的 Ping 扫描,需要执行命令如下:

```
nmap -sn 192.168.1.1/24
```

其含义是,扫描网络 IP 为 192.168.1.1,而子网掩码为 255.255.255.0 的网络中有哪些可以 Ping 通的 IP 地址。

IP 类在扫描单个 IP 地址上被监听端口过程中需使用 Nmap 提供的 Intense 扫描,需要执行命令如下:

```
nmap -Pn 192.168.2.1
```

其含义为,扫描 IP 地址 192.168.2.1 上所有已经被监听的端口号。

命令执行完后返回结果:

```
PORT      STATE SERVICE
80/tcp    open  http
1900/tcp  open  upnp
MAC Address: 14:E6:E4:8D:2B:E2 (Unknown)
```

2.2 远程命令试探部分

这部分逻辑通过一组工厂类被实现。当设备类型被检测完成后就会实例化响应类型的对象来完成后面的信息收集与数据录入工作。考虑到设备类型较多,都在一个工厂类内实现太过繁杂,可以将具体工作分解到若干个类中完成,每个工厂负责某一大类设备的识别与实例化工作,形成一个三层的工厂树。处理具体大类的工厂类可分为处理服务器的 ServerFactory,处理交换机的 SwitchFactory,处理存储的 StorageFactory 等等。

首先由顶层的基本工厂类接收一个 IP 地址作为参数,然后将 IP 地址像放上流水线一样,依次交给每一个负责具体大类的工厂类处理,当流水线上的某一个工厂类成功地处理了这个 IP 地址时,就用识别得到的类型为该 IP 地址实例化一个设备对象出来;如果当前工厂类无法处理,则交给流水线上的下一个工厂类继续处理。

下面给出顶层工厂类的实现:

```
package Mgt::Device::Factory::BaseFactory;

use strict;
use Carp;

sub NewDevice
{
```

```
my $ip = shift;

my $ip_obj = Mgt::Network::IP->new('address'=>$ip);

my $device = undef;
$device = Mgt::Device::Factory::StorageFactory::NewStorage($ip_obj);
if(defined $device)
{
    return $device;
}

$device = Mgt::Device::Factory::StorageFactory::NewSwitch($ip_obj);
if(defined $device)
{
    return $device;
}

$device = Mgt::Device::Factory::StorageFactory::NewServer($ip_obj);
if(defined $device)
{
    return $device;
}

confess "Failed to detect device type for IP address ".$ip."!";
}
```

要发送远程命令首先涉及到选用什么网络协议的问题。网络协议一般都会监听固定的端口号,这就是所谓知名端口(见表 1)。通过查找知名端口表就可以知道网络设备打开了哪些网络协议的服务。

表 1 常见网络协议知名端口号

服务名	端口号
FTP	21
SSH	22
Telnet	23
DNS	53
HTTP	80
SNMP	1503

通常,打开 22 号端口意味着设备打开了 SSH 协议的服务器,这是一台 Linux 或者 Unix 系列操作系统的服务器的常见配置。这种情况下,工厂类就会调用 SSH 协议,发送 Linux 或者 Unix 系统上的命令,然后尝试从返回的文本捕获预期的关键字。

下面给出 ServerFactory 处理打开了 SSH 服务的设备的具体实现:

```
if($ip_obj->is_port_opened(22))
{
```

```
my $ssh = Mgt::Network::Protocol::SSH->new( $ip_obj );  
$ssh->open();  
  
#Try Linux  
@lines = @ { $ssh->cmd("uname") };  
foreach my $line ( @lines )  
{  
if( $line =~ /linux/i )  
{  
$server = Mgt::Device::Server::Linux->new( $ip_obj );  
return $server;  
}  
elsif( $line =~ /AIX/i )  
{  
$server = Mgr::Device::Server::AIX->new( $ip_obj );  
return $server;  
}  
}  
  
$ssh->close();  
}
```

可以看到,ServerFactory 在处理的时候,首先检查 22 端口是否已被监听。发现 22 端口已被监听之后,则通过 SSH 协议连接目标 IP 地址,发送 Linux 和 Unix 系列操作系统上的 uname 命令,如果从返回的文本里捕获到“Linux”,表明目标服务器是一个 Linux 服务器,如果捕获到“AIX”,表明目标服务器已经安装了 IBM 的 AIX 操作系统。

3 网络管理工具的完整运行流程

整个工具运行时的完整流程如图 3 所示。scan_network 是整个程序的入口点,通过 -n 选项传递一个网络地址及其子网掩码,Subnet 类会找出这个网段里的所有可用的活动的 IP 地址,然后为每一个 IP 地址调用 scan_ip。在 scan_ip 中,BaseFactory 会被调用,它会依次尝试使用 StorageFactory, SwitchFactory, ServerFactory 来处理这个 IP 地址。每个工厂类都会检查该地址上有哪些端口被打开了,然后使用命令尝试识别设备的类型。类型识别完成后,就会为该地址实例化一个设备对象,调用它的 collect 方法收集信息,然后调用 import_into_database 完成数据录入。

4 结束语

文中网络管理工具基于端口扫描和工厂模式设计实现,具有自动化程度高、易学习、易配置的特点。面向对象的设计风格使其代码逻辑更为清晰,更加易于复用。一改过去管理工具功能单一、分散、成本高、维

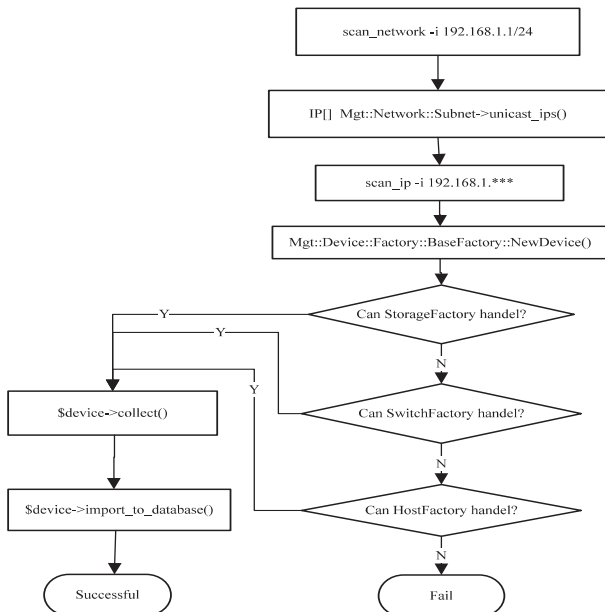


图 3 网络管理工具执行的完整流程

护困难的局面,可大量节省人力,使管理人员更多地把精力集中在对网络整体拓扑的关注上面,而不必过多将精力耗费在细节问题上。管理人员可以基于该工具本身数据库所包含的信息,自行开发一些分析工具,满足自己个性化的需求。信息收集与分析的分离也带来了极大的灵活性,使得该工具的适应范围更广。

参考文献:

- [1] 徐 革,李 宁.基于 FCSAN 和 IPSAN 架构的数字图书馆综合网络存储应用[J]. 计算机应用研究,2005,22(6):168-170.
- [2] 罗 睿,李双庆.中小型网络管理平台的设计[J]. 电脑知识与技术:学术交流,2007(7):51-53.
- [3] 何新江,王 晶.浅谈计算机网络管理技术[J]. 电脑知识与技术,2008(34):1805-1806.
- [4] 魏 晓,胡金初.基于 Linux 系统的分布式网络管理系统[J]. 计算机技术与发展,2007,17(6):60-63.
- [5] 陈 坚,申山宏,成卫青.网络管理发展及其关键技术[J]. 计算机技术与发展,2011,21(4):214-218.
- [6] 胡 丹,龚 俭.基于 .NET 的分布式网络管理软件框架[J]. 计算机技术与发展,2006,16(3):89-91.
- [7] Robison T R, Zhang Du, Mitchell W. Discovery agent deployment in diverse SAN environments[C]//Proc of IEEE international conference on information reuse and integration. [s. l.]: IEEE, 2005:439-444.
- [8] Brothers T J, Muknahallipatna S, Hamann J C. Fibre channel switch modeling at fibre channel-2 level for large fabric storage area network simulations using OMNeT++: preliminary results[C]//Proc of 32nd IEEE conference on local computer networks. Dublin: IEEE, 2007:191-202.
- [9] 魏 勇.一种专用网络管理平台的设计思路[J]. 通信技

像效果。按照图 3 和图 4 所示,两种算法都能产生所需求的高质量图像,两者并没有明显的差别,都有很好的图像效果,说明改进的算法保证了图像质量。表 1 为文中改进算法与传统算法、基于包围盒算法和基于光线跳跃算法在绘制时间和旋转时间上的比较。

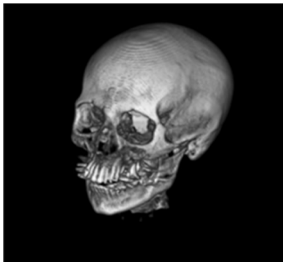


图 4 改进光线投射算法

表 1 文中算法与传统方法绘制时间比较 s

绘制算法	绘制时间	旋转时间
传统光线投射算法	5.23	5.06
基于包围盒算法	3.96	3.83
基于光线跳跃算法	3.04	2.94
改进的光线投射算法	1.89	1.83

根据表 1 的结果显示,相较于传统光线投射算法,基于包围盒的光线投射算法、基于光线跳跃的光线投射算法和文中改进的光线投射算法在绘制时间和旋转时间上分别有 1.32 倍、1.73 倍、2.76 倍的提高,可见文中改进的算法绘制速度最快。

4 结束语

文中提出了一种基于碰撞检测包围盒技术和光线跳跃的加速算法。首先利用包围盒技术隔离体数据周围无效的空体素,减少投射光线的数目;然后利用光线跳跃技术,在包围盒内快速进行光线的合成,跳过透明的空体素,减少大量体素的重采样。实验结果表明在图像质量不受影响的前提下,与传统光线投射算法、基于包围盒算法和基于光线跳跃算法相比,文中改进的算法在很大程度上减少了体素采样的计算,提高了绘制速度,同时又不降低图像质量。未来工作希望能改进包围盒技术和进行动态步长大小的设置,以适应不

同形状大小的体数据,进一步完善算法。

参考文献:

[1] 唐泽圣. 三维数据场可视化[M]. 北京:清华大学出版社, 1999.

[2] Lakare S, Kaufman A. Light weight space leaping using ray coherence[C]//Proc of visualization. [s. l.]; IEEE, 2004: 19-26.

[3] 樊 鹏,郭宝龙. 一种改进求交的自适应光线投射体绘制算法[J]. 计算机工程与应用, 2008, 44(4): 70-72.

[4] Tao Ling, Qian Zhiyu. An improved fast ray casting volume rendering algorithm of medical image[C]//Proc of 4th international conference on biomedical engineering and informatics. Shanghai; IEEE, 2011: 109-112.

[5] 何 晶,陈家新,黎 蔚. 基于 GPU 的实时光线投射算法[J]. 计算机工程与应用, 2008, 44(9): 40-42.

[6] 周志光,陶煜波,林 海. 一种有效显示隐藏特征的光线投射算法[J]. 计算机学报, 2011, 34(3): 517-525.

[7] 梁承志,高新波,邹 华,等. 空间跳跃加速的 GPU 光线投射算法[J]. 中国图象图形学报, 2009, 14(8): 1684-1688.

[8] Levoy M. Display of surfaces from volume data[J]. Computer Graphics and Applications, 1988, 8(3): 29-37.

[9] 边园飞,杨 光,杨红雨. 改进的飞行模拟碰撞检测算法研究[J]. 计算机技术与发展, 2013, 23(1): 5-8.

[10] 赵奇峰,郭宝龙,张 磊. 基于 VTK 的医学图像可视化三维重建[J]. 计算机技术与发展, 2009, 19(6): 39-42.

[11] Hubbard P M. Approximating polyhedra with spheres for time-critical collision detection[J]. ACM Transactions on Graphics (TOG), 1996, 15(3): 179-210.

[12] 杨 晓,廉静静,张新宇. 基于 OSG 的虚拟场景中包围盒碰撞检测的研究[J]. 计算机技术与发展, 2011, 21(9): 32-34.

[13] Mascolo S, Casetti C, Gerla M, et al. TCP westwood: bandwidth estimation for enhanced transport over wireless links [C]// Proceedings of the 7th annual international conference on mobile computing and networking. [s. l.]; ACM, 2001: 287-297.

[14] Çelebi Ö C, Cevik U. Accelerating volume rendering by ray leaping with back steps[J]. Computer Methods and Programs in Biomedicine, 2010, 97(2): 99-113.

[12] 郭其标. 一种综合网络管理方案及其实现[J]. 科技信息, 2007(20): 360-361.

[13] 张德文,徐孟春,杜晓宁. XML 技术在分布式网络管理系统中的研究与应用[J]. 信息工程大学学报, 2007, 8(2): 231-234.

[14] Kim Song-Kyoo. Stochastic design of enhanced network management architecture and algorithmic implementations[J]. A-merican Journal of Operations Research, 2013, 3(1): 87-93.


(上接第 66 页)

术, 2008, 41(9): 98-100.

[10] Cavalas D, Greenwood D, Ghanbari M, et al. Advanced network monitoring applications based on mobile/intelligent agent technology[J]. Computer Communications, 2000, 23(8): 720-730.

[11] 刘 波,罗军舟,李 伟. 大规模网络管理中的任务分解与调度[J]. 通信学报, 2006, 27(3): 64-72.

智能化网络管理工具的设计研究

作者：[王瑛](#), [WANG Ying](#)
作者单位：[上海中医药大学 现教中心, 上海, 201203](#)
刊名：[计算机技术与发展](#) 
英文刊名：[Computer Technology and Development](#)
年, 卷(期): 2014(8)

引用本文格式: [王瑛](#), [WANG Ying](#) [智能化网络管理工具的设计研究](#)[期刊论文]-[计算机技术与发展](#) 2014(8)