

一种通用的拓扑发现系统框架

赵中伟

(中兴通讯南京研发中心 网管业务开发一部, 江苏 南京 210012)

摘要:随着信息网络的飞速发展,计算机网络在工业发展中承担着越来越重要的作用。越来越多的设备运行在网络上,越来越多的新的网络不断涌现,使得网络的管理和维护变得更加的复杂和困难。为了更好地管理和维护网络,需要一些自动化的软件系统帮助管理。网络拓扑自动发现作为网络管理系统中的重要功能,能够对网络拓扑进行自动和准确的发现。不同网络有不同的拓扑发现算法。文中运用 JAVA 和 XML 技术设计和实现了一种能够在不同网络中方便移植的拓扑发现系统框架,大大减轻在不同网络中开发拓扑发现系统的复杂度。

关键词:网络管理;拓扑发现;XML;JAVA

中图分类号:TP301

文献标识码:A

文章编号:1673-629X(2011)02-0131-05

A Compatible Topology Discovery Framework

ZHAO Zhong-wei

(Network Management Application Development Department 1, Nanjing R & D
Institute of ZTE Corporation, Nanjing 210012, China)

Abstract: As the fast development of information network technology, computer network is going to be a more and more important role as the development of industry. There are more and more equipments operating on the network, which makes more and more new network emerge. Hence, it is complicated and difficult to manage and maintain the network. For better manage and maintain the network, the automatic software system is needed to help manage and maintain the network. As an important function in the network management system, automatic discovery of network topology can discover the network equipments precisely and automatically. Different network need take different algorithms. This paper designed and implemented a topology discovery architecture which is portable among different networks based on JAVA and XML. The architecture can greatly reduce the complexity of developing new topology discovery system.

Key words: network management; topology discovery; XML; JAVA

0 引言

随着网络技术的不断发展,网络规模的越来越大,如何有效地管理整个网络,继而达到对网络设备的发现和管理越来越重要,拓扑发现模块在网络管理系统中处于重要地位,同时也是故障管理、配置管理、性能管理、安全管理的前提。如何尽可能地提高拓扑发现模块的可移植性,降低拓扑发现模块的开发难度越来越成为网管开发工程师的首要选择^[1,2]。

为此,拓扑发现系统需要满足以下5条要求:

1) 框架系统的可重用与可扩展,本拓扑发现系统框架需要独立于具体的网络,具有能够方便支持各种不同网络的快速开发的特性,因此可扩展性显得尤为重要。

2) 客户端显示的国际化,统一对客户端的显示风

格,显示语言,进行国际化。同时,对于客户端的语言和显示风格可以方便的修改。

3) 整个系统的资源能够有效集中控制,统一设定系统架构的资源获取点。尽可能使得各种资源的可替换,界面图标、菜单、快捷键、链接等显示单元可以有效的通过配置文件进行动态的修改、增加、删除等。

4) 采用组件化的设计思想,系统框架能够支持动态组件的接入,包括组件的启动、关闭、重启、新增。功能之间需要较低的耦合,功能内部需要高度内聚。组件是可替换的、可升级的,方便后续功能组件的替换与更新。

5) 系统的快速的二次开发周期,高度可维护性,灵活的可扩展性,低廉的开发成本,以及易于集成其他软件^[3]都是本系统首要追求的性能要求。

1 拓扑发现系统框架设计

考虑到要把本系统设计成为一个通用的拓扑发现系统平台,可以方便的添加和支撑各种其它的网络的

收稿日期:2010-06-17;修回日期:2010-09-20

作者简介:赵中伟(1982-),男,江苏南京人,硕士,高级工程师,研究方向为网络管理、通信软件。

拓扑发现功能,因此,开放性和可重用性是本系统应该优先考虑的问题,因此系统架构的应该符合架构设计模式的开闭原则,即软件架构应该对修改关闭,对扩展开放^[4,5]。为了满足这种需求,考虑到软件开发的时间周期和难度,考虑当前的很多优秀开源项目基于这些设计思想,例如非常成熟的 Spring^[6]和 Hibernate^[7]开源框架。因此,拓扑发现框架在借鉴这些优秀开源框架的设计思想的前提下,进行一定的修改和完善,从而达到支持扩展的,具有高度通用性的拓扑发现框架。除此之外,界面的展现也是非常重要,随着 web 应用的趋势,拓扑发现系统框架需要支持 web 接口。此外 MVC 模式已经在目前的工业设计中得到了广泛的应用,因此可以借鉴 MVC 的思想,采用 Java Swing 或者 Struts^[8]来构建拓扑发现系统的客户端。图 1 所示为按照本系统要求设计的整体框架图。

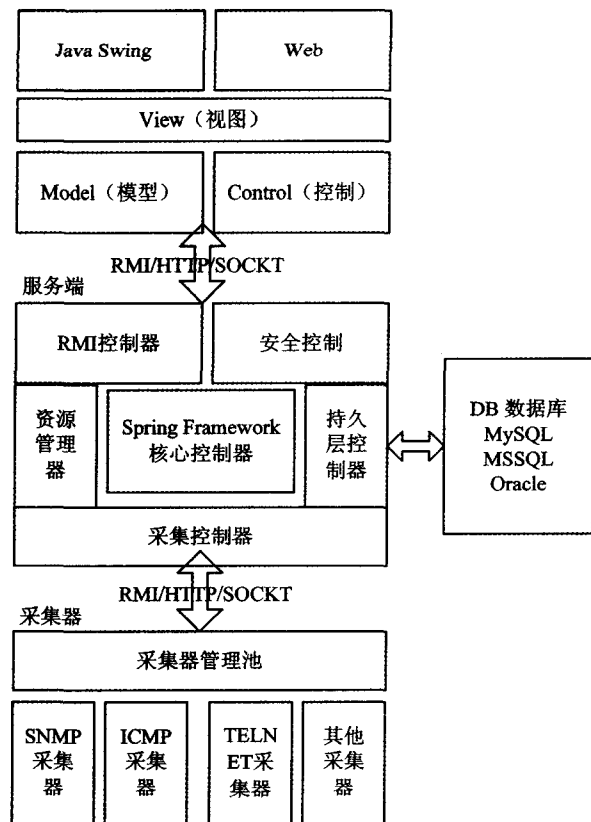


图 1 拓扑发现系统整体框架设计图

从图 1 所示的拓扑发现系统整体框架来看,框架设计分客户端框架、服务端框架、采集端框架。三层采用分布式技术,可以支持独立部署,也可以一起部署在一台主机上。独立部署时,各层框架之间采用 RMI/HTTP/SOCKET 通行协议进行消息交互。

客户端:客户端是整个拓扑发现的呈现界面,可以说一个成功的拓扑发现系统,客户端展现的如何,将直接决定拓扑发现系统的成败。考虑到拓扑发现系统客户端的实现复杂性,考虑到后续的维护以及扩展,考虑

采用基于 MVC 模式的模式开发^[9],具体实现是通过 JAVA Swing 实现胖客户端,还是使用 Struts 实现 WEB 界面有具体的应用场景决定。客户端视图 View 是呈现给用户的显示界面,用户通过操作显示界面上的显示组件,例如菜单、按钮、链接、图标等来触发相应的动作,界面的拓扑显示数据由 Model 负责保存,而 MVC 中的 C 即 Control(控制器)则负责捕获用户的各个动作,将这些动作统一管理,由动作调度线程负责逐一调度并完成对用户视图操作的相应。客户端只负责处理与界面显示相关的显示数据,具体发现的复杂数据结构不属于客户端处理范围。

服务器端:服务器端是整个系统的控制中心,主要提供包括日志服务、JNDI 服务器、客户端接入控制、数据库连接池、安全管理、任务调度、JMS 消息控制。除此之外还负责对服务器文件的统一存取,对数据库的访问控制,对采集器的控制。为了使得各种底层服务功能能够升级和替换,需要抽象出一个业务逻辑与底层服务之间的统一调用接口。该接口将本服务器提供的所有的服务进行登记,并且分配功能码。同时采用抽象调用语言,使得应用能够通过系统抽象出来的功能调用语言,调用服务器具体的底层功能。通过这种方式,避免了硬编码,降低了模块之间的耦合性。使得上层应用代码可以在不加修改的情况下,通过简单的替换底层服务,从而达到整体系统的升级和替换。

数据库:考虑到不同网络的规模,不同网络的拓扑数据的多少,因此数据库需要支持在线升级和替换。为了保持应用层与具体数据库之间的无关性,采用 Hibernate 中间件实现持久层。Hibernate 通过采用不依赖与特定数据库的 HQL 语言和封装了各种底层数据库之间的差异性,做到了对于各种数据库的支持。此外,灵活的数据库配置,统一的数据库调用方法,都为后续数据库的替换提供了方便。

采集器端:不同的网络类型,不同的网络设备,需要采集不同的数据,因此采集器设计的好坏,直接关系到拓扑发现系统是否能够扩展,是否能够支持不同的网络的关键模块。采集器通常需要通过线程池实现,线程池中每个具体的线程就是一个具体的采集器。采集器需要提供统一的调用接口,通过抽象出的统一调用接口,调用具体的采集器。采集器端其实就是一个大规模的采集器池,使用了对象池的功能,采集器池支持采集器的增加、修改、删除。并且,服务端和采集器之间交互的数据结构必须是全局性的,不能依赖与具体的采集器设计,这样可以使得采集器端的实现更加的灵活和方便。

以上各层之间可以支持独立部署,也可以支持合

统支持 RMI、HTTP、SOCKET 协议,同时也可以采用其他安全性更好的通信协议。

2 拓扑发现框架的设计

拓扑发现框架除了需要实现底层框架公用的功能,同时能够方便地支持各个具体二次开发的接入。因此需要提供方便的接入方法,使得二次开发具体应用的时候,能够方便的接入系统框架,本系统定义了业务无关性的接口,只要应用模块按照框架定义的接口,进行实现,就可以方便的接入系统框架。主要包括客户端接入接口、通信接口、服务端接入接口、服务端/采集端与数据库接口、采集器端接入接口。

2.1 客户端设计

客户端框架负责提供抽象过之后的界面显示组件,包括网元模拟显示、网元之间的连接线,只要将发现的拓扑数据,按照客户端的显示框架,加入客户端显示框架,就可以快速的在界面上显示一个网元,同时根据网元设备之间的连接关系,设置连接数据结构,就能够显示出指定网元之间的连接,客户端采用观察者设计模式^[10]。除此之外,客户端应用模块是通过主菜单的方式接入框架中,通过菜单配置的 XML 文件 menu_zh_CN.xml,通过在 XML 文件中加入菜单项,可以实现菜单的接入,同时需要指明响应类。这样的话,通过 XML 文件,利用 JAVA 反射机制,动态的捕获事件,灵活的实现事件处理器,可以更加方便的接入客户端。

```
<menu-root>
  <menu-item
    name = "NETWORK_LAYER_DISCOVERY"
    i18nkey = "NETWORK_LAYER_DISCOVERY_KEY"
    action = "Layer. NetworkDiscoveryMennuImpl"/>
</menu-root>
```

其中 name 代表菜单名,i18nkey 代表国际化字符串的 key,而 action 是具体选中该菜单之后的响应类。该类需要实现框架提供的 MenuActionIf 接口。该接口如下定义:

```
public interface MenuActionIf {
    public void actionMenu(Client client) throws Exception;
}
```

其中 client 对象为当前客户端的上下文对象。当用户需要在客户端增加新的应用功能的时候,只需要

在 menu_zh_CN.xml 文件增加一行菜单配置,增加一个实现了 MenuActionIf 的 action 类以及对应功能的显示界面即可,大大方便了客户端功能的扩展。

2.2 服务端接口设计

服务器是整个拓扑发现系统的核心,也是使得整个框架能够实现网络设备无关性的关键。服务器框架部分具有客户端远程接入(RMI 访问控制)、资源访问控制接口、配置文件访问接口、数据库访问控制接口、采集器控制接口。服务端系统框架接口设计如图 2 所示。

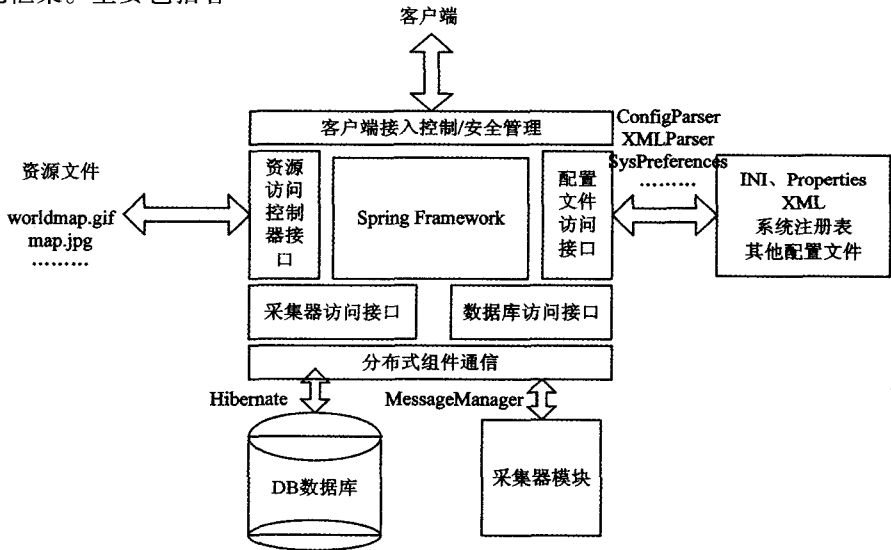


图 2 服务端系统框架接口设计

(1)资源管理器接口。

ResourceManager 资源管理器接口,提供了系统资源获取方式,这些资源包括图片、文件、数据文件、模版文件等等。其设计原理包括,根据对应的模块名,定位到指定的路径下获取相应的资源,除此之外,ResourceManager 类还封装了对与各种资源显示的国际化接口,通过该 ResourceManager 获取的各种资源可以实现自动的国际化。通过 ResourceManager 获得的资源,集中实现了国际化,以及显示的定制。通过这样的设计可以实现对资源访问的控制,便于修改资源的显示名称和渲染方法,而且即使后续资源文件需要变动,其代码的修改量也是较低级别的。

(2)配置文件访问接口。

配置文件访问接口封装了对各种配置文件的访问,支持的配置文件包括当前主流的 INI, Properties, XML 文件,除此之外还支持对 windows 的注册表的访问。

本系统框架提供了几种常用的配置文件的实现,具体实现如表 1 所示。

(3)服务端采集器控制器接口。

由于采集器是后续最有可能频繁变化的模块,因

表 1 配置文件访问器

类名	支持的功能
ConfigParser	支持对 INI 文件的访问,可以存取一组数据,或者获取单节点数据
XMLParser	XML 文件解析器,解析生成 XML 文件,提供操作 XML 节点的封装方法
SysPreferences	获取 windows 注册表下面的节点或者属性值,可以通过 SysPreferences 保存程序配置数据。除此之外,还可以将注册表中的数据导出保存

此设计的时候需要考虑应用层的代码调用应该与具体的采集器区分开。因此服务端需要抽象出采集器访问控制接口从而控制采集器的调用、运行、关闭等一系列的动作。为此,服务器端借鉴了当前流行的命令行设计思想,将对具体方法的调用,改为对具体命令的调用,而命令是全局性的抽象,与具体的方法实现是分开的,这样可以保持应用层代码的稳定。服务端定义了 MessageManager 接口,改接口封装了具体的方法,将应用层的抽象的采集器命令调用巧妙的转化为对于具体的采集器的触发和调用,框架定义了 SEND、GET、CANCEL 三个命令。这个三个命令都接收一个全局性的输入参数 MessageBean, MessageBean 是全局唯一的,与具体的采集器是无关的。其流程如图 3 所示。

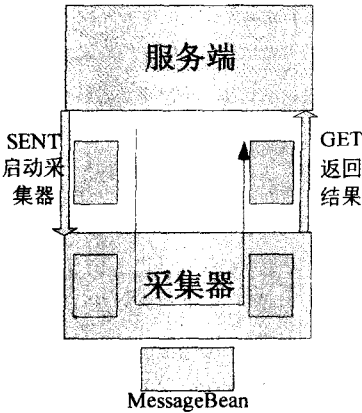


图 3 服务端控制采集器采集流程

2.3 服务端与数据库接口设计

考虑到采集器需要支持分布式部署,因此所有的采集器采集的数据统一返回给服务器,由服务器根据具体的算法逻辑,将采集的数据进行过滤并存入数据库,数据库与服务器可以是分布式部署,也可以部署在同一台机器上。将对 DB 的访问封装到 DBAccessManagerIf 接口中。

```
public interface DBAccessManagerIf {  
    public Session getConnection( String DbName,String user,  
String password,String instanceName);  
    public void returnConnection( Session session);  
    public void closeAll();  
    public void destroy();  
}
```

Hibernate 通过采用 POJO/hbm. xml 文件,将应用对象和数据库中的表对应起来,对象的属性与表的列一一对应,通过这种 OR 映射关系,使得上层用户屏蔽了关系数据表的操作,而只有操作对象的感觉^[11]。数据持久层整体框架模型如图 4 所示。

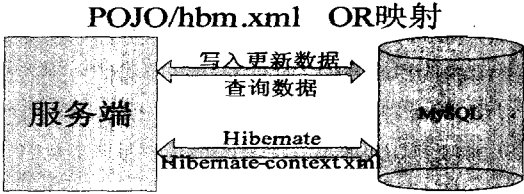


图 4 数据持久层整体框架模型

数据库连接配置:在 Hibernate-context.xml XML 文件中配置一个 dataSource,可以配置不同的数据库类型,系统默认配置 Mysql。除此之外,还可以方便的支持 SqlServer,Oracle 等其他数据库。

2.4 采集器端的接口设计

采集器是整个拓扑发现系统框架的核心。采集器的作用是通过特定的采集数据的方式获得网络中运行设备的特征数据,通过这些特征数据来做到对设备的识别。采集器利用 XML 和 JAVA 反射机制,支持随时增加新的采集器类,这种设计对于增强本系统的通用性有决定性的意义,即使网络不同,只要更换了采集器类就可以发现这些不同网络的设备,同样,如果网络新增加了一种设备,也只需要增加一个采集器类,就可以进行发现和管理。采集器采用资源池的设计模式实现。资源池是管理与共享系统资源的一种缓存机制,亦称对象池。使用资源池可以获得两大好处:一是可以降低资源创建与撤销的成本,从而提高系统整体性能;二是有效控制系统资源的使用总量,避免系统因资源耗尽而崩溃,从而提高系统的可伸缩性。

```
public interface Job {  
    public void run()throws Exception;  
    public void resume()throws Exception;  
    public void suspend()throws Exception;  
    public boolean isJobRun();  
    public void stopRun()throws Exception;  
}
```

系统默认提供 SNMP 采集器和 ICMP 采集器,这两种采集器在拓扑发现中具有重要作用^[12]。采集池提供了扩展 Job 类,要想增加自己的采集器,只需要继承 Job 类,实现 Job 类定义的方法,通过这种方式可以将自己的采集器添加到框架中。由采集管理框架负责对所有采集器进行管理和控制。数据采集端整体框架模型如图 5 所示。

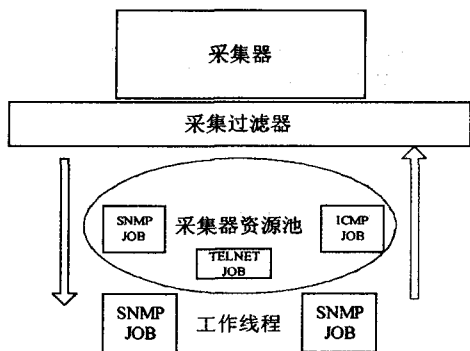


图5 数据采集端整体框架模型

3 结束语

文中介绍了一种通用的、可扩展的拓扑发现系统架构,并给出部分的实现。文中介绍的拓扑发现系统架构,可以很容易的在特定的网络下,迅速开发出适应特定网络的拓扑发现系统,能够大大缩短拓扑发现系统开发的周期和难度。此外,文中介绍的设计思想也是当前网络管理系统开发过程中的核心思想。即系统架构设计的开闭原则,系统需要尽可能的对修改关闭,对扩展开放。这样的系统才是高质量的、具有通用性的系统。

参考文献:

- [1] Cassel L N, Partridge C, Westcott J. Network Management

(上接第130页)

问题,将QoS应用到语义Web服务发现过程当中,建立了基于QoS的语义Web服务发现框架,并实现了服务的三层匹配模式。实验验证了该框架能有效提高服务发现的精确度和准确度。

未来工作主要包括:

(1)如何结合用户的反馈信息以及第三方认证机构对Web服务进行服务质量评估;

(2)服务发现的优化,例如采用聚类技术对服务进行预处理等。

参考文献:

- [1] Paolucci M, Kawamura T. Importing the Semantic Web in UDDI[C]// Proceedings of Web Services, E-business and Semantic Web Workshop (CAiSEWorkshop). Toronto, Canada: [s. n.], 2002:225-236.
- [2] Ran S P. A model for web services discovery with QoS[J]. ACM SIGCOM Exchanges, 2003, 4(1): 1-10.
- [3] Chen H C, Yu T, Lin K J. QCWS: An implementation of QoS-capable multimedia Web services[C]// In: Proceedings of the 5th International Symposium on Multimedia Soft-

Architectures and Protocols: Problems and Approaches[J]. IEEE Journal on Selected Areas in Communications, 1989, 7(7):1104-1114.

- [2] 陶杨,胡敏.网络管理原理与实践[M].北京:科学出版社,2000:35-39.
- [3] 赵中伟.一种拓扑发现系统的研究与实现[D].武汉:华中科技大学,2007.
- [4] 阎宏.JAVA与模式[M].北京:电子工业出版社,2002:50-55.
- [5] 杨涛,石坚,刘伟.基于RMI的EPON EMS北向接口接口设计与实现[J].计算机技术与发展,2007,17(5):46-48.
- [6] Johnson R, Hoeller J. Expert One-on-One J2EE Development without EJB[M]. Indianapolis: Wiley Publishing, Inc, 2004: 203-207.
- [7] Hibernate Organization, Hibernate Documentation[EB/OL]. 2004. <http://www.hibernate.org>.
- [8] 孙卫琴.精通struts:基于MVC的Java Web设计与开发[M].北京:电子工业出版社,2004:1-15.
- [9] Gamma E. Design Patterns[M]. US: Addison-Wesley Pub Co, 2000:240-245.
- [10] 王江涛.观察者设计模式在嵌入式系统中的应用[J].计算机工程,2004,30(21):66-68.
- [11] 夏昕,曹晓钢,唐勇.深入浅出Hibernate[M].北京:电子工业出版社,2006:91-96.
- [12] 李佳,石冰心.基于ICMP和SNMP的网络拓扑发现算法研究及实现[J].微型机与应用,1998(1):33-35.

ware Engineering. Taichung, Taiwan: [s. n.], 2003:38-45.

- [4] 杨胜文,史美林.一种支持服务质量约束的Web服务发现模型[J].计算机学报,2005,28(4):589-594.
- [5] 党伟超,白尚旺.一种基于OWL-S的Web服务体系结构[J].计算机技术与发展,2007,17(5):195-198.
- [6] 杜小勇,李曼,王珊.本体学习研究综述[J].软件学报,2006,17(9):1837-1847.
- [7] 史忠植,蒋运承,张海俊,等.基于描述逻辑的主体服务匹配[J].计算机学报,2004,27(5):625-635.
- [8] 徐精明,程家兴.多态蚁群算法中多参数的组合匹配探讨[J].计算机技术与发展,2009,19(12):85-88.
- [9] 胡建强,邹鹏,王怀民,等.Web服务描述语言QWSDL和服务匹配模型研究[J].计算机学报,2005,28(4):505-513.
- [10] 仲梅,宋顺林.一种语义Web服务的多层次匹配方法[J].计算机应用,2007,27(1):199-204.
- [11] 刘斌.基于本体的语义服务选择研究[D].北京:北京邮电大学,2008.
- [12] 戴雪梅,姜浩.基于带权图规划算法的语义Web服务组合[J].计算机技术与发展,2010,20(3):67-70.