

文章编号:1001-9081(2008)04-1042-03

## 基于 LDAP 的信息共享平台研究与实现

吴洁明<sup>1</sup>,周宁<sup>2</sup>

(1. 北方工业大学 计算机系,北京 100041; 2. 北京国研数通技术有限公司,北京 100032)  
(wujie ming@263.net)

**摘要:**不同时期建设的计算机应用系统采用不同的技术,运行在不同的平台上,它们之间的信息共享成为人们关注的焦点。利用轻型目录(LDAP)服务器组织共享信息资源的元数据,运用对象缓冲池、多线程和消息总线等技术实现资源管理器、数据提取器和消息总线,运用系统引擎屏蔽底层的复杂结构,结合 XML 技术解决共享信息的交互和多模板显示问题,有效地实现了 Internet 环境下不同的应用系统之间的信息的共享。

**关键词:**信息共享;信息交互;消息总线;LDAP 应用;目录服务;元数据;可扩展标识语言

**中图分类号:**TP311.52 **文献标志码:**A

## Research and implementation of information sharing system based on LDAP

WU Jie-ming<sup>1</sup>, ZHOU Ning<sup>2</sup>

(1. Department of Computer, North China University of Technology, Beijing 100041, China;  
2. SRIT Software Technology Company Limited, Beijing 100032, China)

**Abstract:** The computer application systems constructed in different time use different technologies and run on the different platforms. The focal point is the information sharing. The Lightweight Directory Access Protocol (LDAP) server organization was used to share metadata of the information resource. Resource supervisor, data extractor and news main line were realized by object buffer pool, multithreading and news main line. The complex structure of rock-bottom shielded by system engine and the XML technology was unified to solve problems of multi-template demonstration and sharing information alternately. The information share in Internet environment between the different application systems has been realized effectively.

**Key words:** information sharing; information exchange; news bus; LDAP application; directory service; metadata; XML

### 0 引言

随着网络应用的深入,人们迫切要求跨部门、跨行业实现不同应用系统之间的信息资源共享。但是,由于应用系统建设时期不同,所用平台不同、技术不同、信息描述方式不同,给应用系统之间的信息共享带来了一定困难。

目前解决广域环境下不同应用信息系统之间信息资源共享主要有 3 个途径<sup>[1,2]</sup>:1)建立数据中心,将需要共享的信息采集后集中存储。2)利用 Web 服务技术,将共享信息资源升级为服务,通过服务接口访问需要的信息。3)建立数据目录表,存储数据的位置信息和其他描述信息,通过访问数据目录表发现共享信息位置及其描述信息,获取指定位置的信息。

第 1 种途径通常需要建立大规模的数据中心,软硬件投资非常大,并且数据集中存放的安全性、一致性要求都比较高<sup>[1]</sup>。第 2 种途径需要将不同的应用软件注册为服务,建立统一的服务管理中心和资源目录管理中心。首先在资源目录管理中心查找需要的信息和服务,然后启动相应的服务获得共享信息。这种方法从技术上是可行的,但需要将所有需要共享信息的应用都升级为服务,由于涉及的因素较多,实现难度比较大,目前还没有成熟的应用案例<sup>[2]</sup>。本文介绍的方法属于第 3 种,用目录服务器实现共享信息资源的元数据管理,

将 XML 作为数据转换和传输的载体,采用对象链接池和多线程等技术实现数据提取器,与 XSL 相结合,提高数据显示的灵活性。这种方法原理简单,实现成本低,可以有效地解决广域环境下不同应用系统间的信息资源共享。

### 1 基于 LDAP 的信息资源管理

LDAP (Lightweight Directory Access Protocol) 是由美国 Michigan 大学研发的一个轻量级目录访问协议。它的典型应用是将网络环境中的各种资源作为目录信息,在目录树结构中分层存储,对这些信息进行访问和管理<sup>[3]</sup>。与关系数据库相比,LDAP 的检索性能更加迅速,安全机制更加完善,并且由于 LDAP Server 是运行在 TCP/IP 的上层,因此具有更好的跨平台特性<sup>[4]</sup>。

采用 LDAP 实现信息资源共享,首先要解决共享信息资源的管理问题。这里用 LDAP 服务器存储 3 类信息:1)用户信息。共享信息的用户需要注册,由系统验证其身份。2)共享信息的位置。共享信息可以存储在广域网环境下的任何位置,具体的地址信息由专人统一管理。3)共享信息的元数据。共享信息本身存储在各个应用系统中,这里只存储它们的描述信息。图 1 是共享信息资源的目录结构,由于站点信息和元数据都属于共享信息的描述信息,因此将它们统归为

收稿日期:2007-10-17。

基金项目:北京市教委科技发展计划资助项目(KM200610009012);北京市科委资助项目(D1016002000091;D0106002040291)。

作者简介:吴洁明(1958-),女,河北清苑人,教授,主要研究方向:软件工程、软件体系结构;周宁(1959-),男,浙江宁波人,高级工程师,主要研究方向:计算机网络、计算方法。

元数据子树。

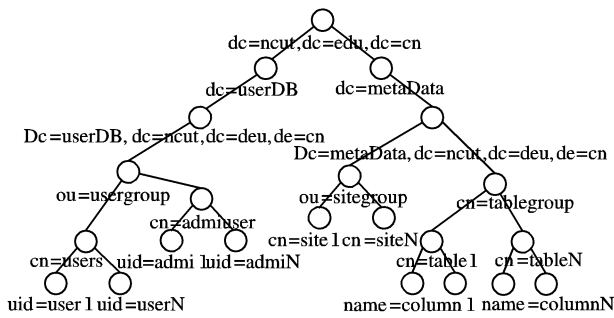


图 1 共享信息资源目录树结构

下面以某应用系统中的数据库表 state\_lookup 为例,说明共享信息的元数据描述在目录表中的描述。

```

dn : cn = state_lookup, cn = tablegroup, dc = metadata, dc = ncut,
    dc = edu, dc = cn
cn: state_lookup
objectClass: top
objectClass: table

```

objectClass 属性说明节点类型是 table,属性 dn 唯一标识了该节点在目录树中的位置。假设 state\_lookup 表有多个列,其中一列是“state\_desc”,这列的描述信息在目录中解析如下:

```

dn : name = state_desc, cn = state_lookup, cn = tableGroup,
    dc = metaData, dc = ncut, dc = edu, dc = cn
name: state_desc
objectClass: top
objectClass: column
type: string

```

其中 dn 唯一标识了该列的名称和在目录树中的位置,属性 objectClass 说明是列对象,name 属性给出了列名 state\_desc,属性 type 指出了该列在数据库中的数据类型是 string。

站点元数据为远程数据提取提供了站点的名称、IP 地址、站点的描述以及远程服务器所监听的端口等信息。下面是站点元数据的描述实例:

```

dn : cn = site1, ou = sitegroup, dc = metaData, dc = ncut, dc = edu,
    dc = cn
objectClass: top
objectClass: site
cn: site1
url: 10.17.158.68
port: 8086

```

其中属性 dn 指出了站点的全称和在目录树中该目录项所处的位置,objectClass 指出该目录项的类型。cn 属性给出了站点的名称,url 属性是 IP 地址,port 给出了监听的端口。通过上述属性的值,共享信息平台就能连接到相应地址的端口,提取远程数据。

## 2 基于 LDAP 的信息共享平台结构设计

Intenet 环境下基于 LDAP 信息共享平台的运行环境如图 2 所示。用户通过浏览器基于 HTTP 协议访问共享信息平台的服务器,由该服务器负责与 LDAP 服务器通信,获取请求的共享信息的地址和元数据描述信息。如果共享信息是本地应用系统的数据,则启动本地数据提取器,通过 JDBC 获取本地应用系统的数据。如果共享信息不在本地,则依据 LDAP 服务器给出的远程站点,向远程服务器发出 Socket 请求,由远程数据提取器获取共享信息并返回,实现远程数据共享。

整个系统为四层结构:第一层浏览器,用户通过浏览器访问共享信息平台。第二层是表示层,接收用户的请求,并转发到应用层的系统引擎,还负责显示获取的共享信息。第三层是应用层,处理来自用户的请求,这是本系统的核心,包括系统引擎、消息管理器、资源管理器和数据提取器。第四层是数据层,数据层的数据分为两个部分:一是应用系统的数据,它们是共享信息的载体,本系统只是要访问它们;二是共享信息元数据,它们存储在 LDAP 目录服务器上。图 3 是本系统的架构图。

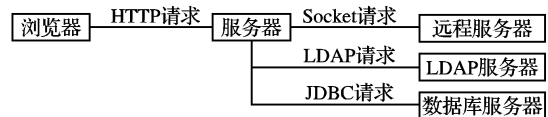


图 2 系统的物理结构

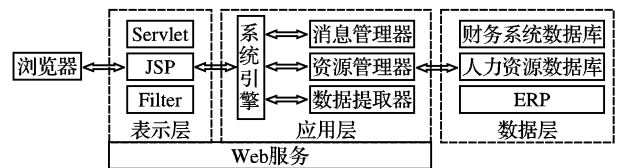


图 3 系统架构

用户通过浏览器访问表示层,请求系统服务。表示层采用 SERVLET、JSP 等技术实现的组件接收来自用户的请求,将请求封装后发送给应用层的系统引擎。系统引擎接收并解析这些请求,将请求路由到相应的组件上,如数据提取器、资源管理器、消息管理器。当组件完成用户请求后,由系统引擎将结果返回表示层。系统引擎作为应用层的代理者屏蔽了应用层的复杂结构。

## 3 资源管理器

资源管理器基于 LDAP 服务器管理用户数据、表元数据、站点元数据,接收来自系统引擎的命令,按照用户要求对目录服务器中的数据进行管理。资源管理器设计了两个接口:一个用于创建 LDAP 资源管理器,基于单体模式,保证在整个系统生命周期中资源管理器的唯一性<sup>[5]</sup>;另一个用于接收来自系统引擎的命令,完成与目录服务器相关的各种任务。

资源管理器采用 stackObjectPool(栈对象池)技术,系统环境信息存储在文件 ConnectionEnv 中。系统初始化时,读取文件 ConnectionEnv 的资源管理配置信息,初始化一定的 LDAPServer,存入对象池中,供资源管理器在需要 LDAPServer 时直接从池中获取对象。栈对象池技术有助于提高共享资源利用率,减少共享资源的频繁创建和销毁,提高系统性能。对象池的基本原理是为对象创建一个“缓冲池”,预先在“缓冲池”中初始化一些对象,当系统需要对象时,如果对象池中有空闲对象,则直接从池中获得该对象;当池中当前没有空闲对象时,则根据要求等待或创建新对象。使用完对象后,将对象归还给对象池,由对象池维护对象。为了防止系统无止境地创建对象,池中一般限制最大对象数目,资源管理器监控池中对象的数量和使用情况。

当系统引擎要求处理目录服务器的相关操作时,系统引擎向资源管理器发送命令,请求 LDAPServer 对象,资源管理器从 LDAPServer 的对象池中获取空闲的对象返回给系统引擎,由系统引擎向 LDAPServer 发送命令,执行相关操作。LDAPServer 完成操作后,将消息返回给系统引擎。系统引擎调用 LDAPServer 对象的 close 方法将自己钝化,回归到对象

池中,等待下次调用。

#### 4 数据提取器

数据提取器根据系统引擎的命令,从本地或者远程服务器中提取数据,组织成 XML Schema 定义的 XML 文档返回给系统引擎。

数据提取器有两个接口:数据提取管理器接口负责接收来自系统引擎的命令和请求,将数据提取指令送给数据提取服务对象,数据提取服务对象提供了从各个应用系统的数据库中提取数据的方法;数据提取器工厂接口负责创建数据提取器,保证数据提取器实例在整个系统生命周期中的唯一性。在设计数据提取服务对象的过程中采用了工厂方法和栈对象池技术。

数据提取分为本地提取和远程提取。提取本地数据时,系统引擎将消息发送给数据提取器。数据提取器从对象缓冲池中获得一个空闲的数据提取器对象,返回给系统引擎。由系统引擎向数据提取服务对象发送数据提取命令,数据提取服务对象收到命令后,从底层数据库中提取所需要的数据,将所提取的数据整理成标准的 XML 文档之后,向引擎发送返回消息。系统引擎接收到返回消息后,调用数据提取对象的 close 方法,完成钝化自己的工作,将数据提取对象返回到对象池中,等待下一次数据提取调用。

从远程站点提取数据时,系统引擎向数据提取器的远程客户端发送异步命令,由远程客户端向远程服务器发送提取数据请求;远程服务器端的远程提取服务对象接收到请求后,将任务送给底层的远程数据提取线程,服务器对象继续接收来自远程客户端的数据提取请求。这时远程数据提取线程担负起与远程客户端的联系,并将请求发送给底层的数据提取器,由数据提取器完成数据的提取,组装成 XML Schema 定义的 XML 文档格式,返回给远程数据提取线程,由该线程将文档返回给远程客户端,完成数据的提取。

远程数据提取和本地数据提取在底层数据提取的处理上是一致的,不同在于本地数据提取采用单线程工作方式,远程数据提取在交互方式上采用了多线程技术,提高了系统的并发性能。多个客户端线程可并发地从不同站点的服务器上提取数据,而服务器端也提供多个服务器线程,响应远程数据提取。

#### 5 消息总线

本系统的消息管理器设计成一个消息总线。在消息总线上注册各种类型的消息监听器,处理系统中发生的相应事件。消息总线中提供了两类消息监听器,一类是持久的消息监听器,存在于整个系统生命周期,由系统的配置文件决定其实例;另一类是临时消息监听器,只存在一段时间,由系统管理。消息总线为系统的扩展提供了一定的灵活性,如果系统需要记录某一类事件的发生,只需要定义相应的 Listener 接口,利用系统配置文件配置该监听器,在整个系统生命周期中就会自动响应该事件。例如,采用消息管理器和事件监听器实现日志功能,当系统完成某项操作时,在消息总线中添加该操作的事件,并向在管理器中注册的所有监听器发布该事件,此时日志监听器提取该事件,完成日志记录工作。设计一个核心类负责消息监听器的注册和卸载,以及事件的通知。设计两个消息队列分别缓存 LDAP 资源管理器事件和数据提取器事件,等待消息总线通知监听器。

#### 6 结语

本文分析了 Internet 环境下解决不同应用系统之间信息共享的三种途径,提出了一种基于 LDAP 服务器构建信息共享平台的具体实现方案。由 LDAP 服务器实现资源管理器,管理共享信息资源的元数据,由数据提取器实现本地和远程共享信息的获取。为了灵活地展示提取的数据,在实际应用中系统提供了 XSL 模板生成器,用户可以选择合适的模板显示共享信息。系统用消息总线保障各个组件之间的消息通信,采用系统引擎有效地屏蔽了系统底层的复杂性,使系统具有良好的可扩展性和灵活性。

#### 参考文献:

- [1] 李长河,甘文丽. 基于 LDAP 和多 Agent 的分布式网络信息检索[J]. 计算机工程,2006,32(8):110-112.
- [2] (美)THOMAS E R L. SOA 概念、设计与技术[M]. 王满红,陈荣华,译. 北京:机械工业出版社,2006.
- [3] 郭军,芦文龙,赵明,等. 基于 LDAPv3 的目录服务系统中推荐的设计与实现[J]. 小型微型计算机系统,2000,21(8):802-805.
- [4] 于剑,张辉,赵红梅. LDAP 目录服务在 Web 开发中的应用[J]. 计算机应用,2003,23(10):82-83,86.
- [5] COOPER J W. The design patterns java companion[M]. Boston: Addison Wesley, 1998.
- [6] 瞿裕忠,张剑峰,陈峥,等. XML 语言及相关技术综述[J]. 计算机工程,2000,26(12):4-6,30.
- [7] 徐建波,李仁发,王舒. LDAP 目录中的检索技术研究[J]. 计算机科学,2002,29(4):141,110.
- [8] 唐建平. 基于 LDAP 技术的企业基础信息平台构建[J]. 计算机应用,2003,23(11):66-68.
- [9] 陈小弟,李长河,张熙,等. 基于 LDAP 的 Internet 分布式目录服务的分析与实现[J]. 计算机工程,2002,28(8):271-273.
- [10] 赵保翠,刘岗. 基于 LDAP 的统一用户管理系统的设计和实现[J]. 微电子学与计算机,2005,22(11):59-61.
- [11] 唐建平. 基于 LDAP 技术的企业基础信息平台构建[J]. 计算机应用,2002,23(11):66-68,102.
- [12] 李少春,蒋泽军,王丽芳,等. J2EE 环境下基于 LDAP 的访问控制的设计和实现[J]. 微电子学与计算机,2005,22(3):118-120,124.
- [13] IETF RFC 2222, Simple Authentication and Security Layer (SASL)[S]. IETF, 1997.
- [14] IETF RFC 2251, Lightweight Directory Access Protocol (v3)[S]. IETF, 1997.
- [15] IETF RFC 2253, Lightweight Directory Access Protocol (v3): UTF-8 Stringy Representation of Distinguished Names[S]. IETF, 1997.
- [16] HOWES T, SMITH M C, GOOD G S, et al. Understanding and Deploying LDAP Directory Services[M]. Basingstoke Hampshire: MacMillan, 1999.
- [17] HOWES T, SMITH M. RFC 2255, The LDAP URL Format[S]. 1997.
- [18] STOKES E, BYRNE D, BLAKLEY B, et al. RFC 2820, Access Control Requirements for LDAP[S]. 2000.
- [19] Sun Microsystem. iPlanet Directory Server Deployment Guide[Z]. Sun Microsystem, 1994.
- [20] ERNESTO L, MALERE P. LDAP Linux HOWTO [EB/OL]. [2007-10-01]. <http://www.linuxjunkies.org/html/LDAP-HOWTO.html>.
- [21] HOLLAND J P. OpenLDAP 快速指南[EB/OL]. [2007-10-01]. <http://www.Unlinux.com>.