基于 LDAP 的目录服务综述*

任 军

(华中科技大学 计算机科学与技术学院, 湖北 武汉 430074)

摘 要: 从简单介绍 X.500 协议出发,介绍 LDAP 的起源,简要比较 LDAP 与 X.500 协议的区别,概述 LDAP 框架模型;从应用目录服务角度,简述了 LDAP 目录服务功能模块和工作流程;介绍分析 LDAP 目录服务发展现状;最后预测其发展方向。

关键词: 目录服务; X.500 协议; LDAP

中图法分类号: TP393 **文献标识码**: A **文章编号**: 1001-3695(2005)05-0008-03

Survey of LDAP-based Directory Services

REN Jun

(College of Computer Science & Technology, Huazhong University of Science & Technology, Wuhan Hubei 430074, China)

Abstract: Introduces the origin of LDAP(Lightweight Directory Access Protocol) after scaning the X.500 protocol, compare the LDAP to X.500, and summarize the frame modol of LDAP. LDAP directory function modol and work flow was discussed briefly. The development of LDAP directory services was analysed, and its development trend was forecasted in the end.

Key words: Directory Services; X.500 Protocol; LDAP

1 引言

随着 Internet 和网络系统的广泛应用和发展,人们可以在世界范围内共享各种资源和信息。网络中的资源纷繁复杂,种类众多,因此,有效管理各种资源信息以利于检索查询至关重要。目录服务技术就是适应网络信息飞速发展而产生的,是一种管理资源信息以方便查询的技术。

目录应用在我们生活中随处可见,像电话薄、字典词典的索引,甚至餐厅的菜谱都可以看成是一种目录应用。但是我们在网络信息领域所说目录服务不是指这种印刷品目录,而是指具有通用目的、基于标准的目录服务[1]。简单定义如下:目录服务就是用一个特殊的数据库存储资源信息,将各种资源信息在目录树结构中分层存储,提供一个单一的逻辑视图,允许用户和程序透明地访问网络上的各种资源。目录服务由于应用环境不同、具体实现不同而呈现各种特点,但是其中如下五个特征是共性的[2]:①目录服务专门为读信息而做了特殊优化;②目录服务实施分布存储模型;③目录服务应能扩展它所存储的信息的种类;④具有高级检索功能;⑤目录信息在目录服务器之间可以松散地复制。

域名服务 DNS 是一个成功的目录服务的典型,但是它仅将 IP 地址与域名绑定,功能非常单一。X.500 协议是第一个具有完备意义的目录服务协议,但是它是基于 OSI 体系模型的,而且其复杂性也限制了其广泛应用。LDAP 脱胎于 X.500,因其简单实用而获得广泛应用。由于 LDAP(Lightweight Directory Access Protocol)已经成为目录服务的事实上标准,本文主要探讨基于 LDAP 的目录服务。

收稿日期: 2004-04-23; 修返日期: 2004-07-12

基金项目: 国家自然科学基金项目(60273075);华中科技大学研究生基金项目(x0219)

2 LDAP 协议简介

2.1 LDAP 起源及概况

轻权目录访问协议 LDAP(Lightweight Directory Access Protocol)是从对 X.500 协议简化的基础上演变而来,所以称为轻量级的目录服务。X.500 协议由 CCITT 和 ISO 两大国际组织各自对目录服务的开发成果融合而产生,它于 1988 年被认可,1990 年初由 CCIT 发布,以后曾数次更新,目前仍在发展中。X.500 协议的早期设计人员由于过于注重其通用性和可扩展性,导致 X.500 协议内容庞杂,开发和部署都极其复杂而且性能不高。特别一点,X.500 协议是基于 OSI 网络模型。随着 TCP/IP 的流行并成为事实上的标准,X.500 协议未能获得广泛应用也在情理之中。

LDAP 从以下几方面对 X.500 协议做了简化和发展[3]:①功能方面,缩减了 X.500 中冗余的和使用频率较小的功能,可以说以极低的代价可完成 X.500 协议 90% 的功能。②数据表示方面,统一采用文本字符串形式,避免数据解释时可能导致的二义性。③编码上,仅采用 X.500 协议的一个子集 - 简单的编码规则 BER,节约了空间,而且大大简化了其实现。④传输上,直接运行于传输层 TCP 之上,减少了在 OSI 通信协议中的高昂开销,不但提高了性能,而且使目录服务部署简单了。

第一个 LDAP 规范于 1993 年发布(RFC1487),到第二版本问世,LDAP 已获得广泛应用,LDAPv2 发布为 RFC1777。到 1997 年,随着 LDAPv3 的发布(RFC2251),LDAP 进入一个更加成熟的阶段。

在 LDAP 中目录是按照树型结构组织,目录由条目(En-

try) 组成, 条目相当于关系数据库中表的记录; 条目是具有区别名(Distinguished Name, DN) 的属性(Attribute) 集合, DN 相当于关系数据库表中的关键字(Primary Key) ; 属性由类型(Type) 和多个值(Values) 组成, 相当于关系数据库中的域(Field) 由域名和数据类型组成, 只是为了方便检索的需要, LDAP 中的 Type 可以有多个 Value, 而不是关系数据库中为降低数据的冗余性要求实现的各个域必须是不相关的。LDAP 中条目的组织一般按照地理位置和组织关系进行组织, 非常的直观。LDAP 把数据存放在文件中, 为提高效率可以使用基于索引的文件数据库, 而不是关系数据库。LDAP 协议集还规定了 DN 的命名方法、存取控制方法、搜索格式、复制方法、URL 格式、开发接口等。

LDAP 主要优点如下: 简单而通用; 普遍存在, LDAP 已广泛用于各种主流和非主流的计算平台; LDAP 目录易于理解; 由于 LDAP 高可靠性和良好性能, LDAP 目录服务能满足绝大部分重要的目录服务需求。如今, LDAP 已经成为目录服务的事实上的标准。

2.2　LDAP 协议框架模型[4]

LDAP 定义了四个基本模型以描述它的工作机制, 描述什么样的数据可以存于 LDAP 目录中, 以及如何操作这些数据。

(1) LDAP 信息模型: 定义了目录中存放信息的基本单位和数据的类型。目录中信息的基本单位是条目(Entry) , 每个条目为一个属性集合, 每个属性含有一个属性类型和一个或几个值。条目相当于现实世界的一个对象, 属性则从某一方面反映对象的特征。另外, 目录大纲(Directory Schemas) 规定了哪些属性是必须具有的, 哪些只是允许存在的。

(2) LDAP 命名模型: 定义了目录的组织和查询方式。LDAP 指定目录条目(Entry) 应被组织成倒转的树型层次结构 - 目录信息树(Directory Information Tree, DIT) 。树根(Root) 是虚根, 树的每个节点都存储信息, 每个节点都有一个属性作为相对名 Rdn。将某个节点回溯到根, 所有 Rdn 一起组成该节点的区分名 Dn。

如图 1 所示 DIT 的一个示例, 虚线所围节点有几个属性: cn = RenJun, objectclass: person, sn = Ren, telephoneNumber: 123。其中 cn = RenJun 是相对区别名 Rdn, Rdn 在其父节点容器内将该节点与其兄弟节点区分开。从该节点回溯到根得到该节点的区分名 dn: cn = RenJun, ou = nhpcc, dc = hust, dc = edu。

(3) LDAP 函数模型: 定义了访问和更新目录的操作。这些操作分为三类: 查询操作, 查询某个条目并返回结果, LDAP 既支持根据区分名查询, 又支持根据某一属性检索; 更新操作, 进行条目或其属性的增加、删除及重命名; 认证及控制操作, 对客户端认证, 控制某些交互行为。

(4) LDAP 安全模型: 定义了如何保护目录信息, 防止未授权用户对目录信息的访问和修改。

3　LDAP 目录服务

3.1　功能模块

LDAP 是运行于 TCP/IP 之上的应用层协议。LDAP 的目录服务功能建立在 Client/Server 模型之上, 所有的目录信息数据存储在 LDAP 服务器中。单独的一个 LDAP 服务器可能无法存储完整的目录信息树, 这时需要将目录信息树分布到多个 LDAP 服务器中。一个或多个 LDAP 服务器组成 LDAP 目录树, 每个 LDAP 服务器由目录服务模块、复制服务模块和管理模块三个模块组成[5], 如图 2 所示。

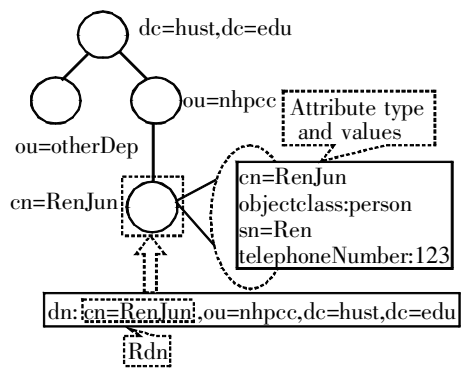


图 1　LDAP 目录信息树一例

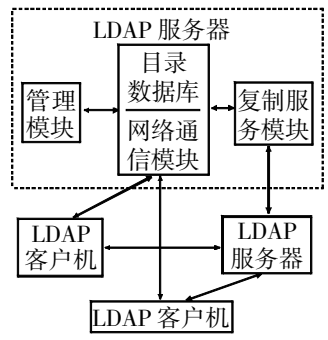


图 2　LDAP 目录服务功能逻辑模块

目录服务模块主要由两部分组成: 前端部分负责通常的客户机与服务器之间的网络通信, 完成协议解析和分析; 后端部分负责目录数据库的管理。

复制服务模块负责 LDAP 服务器之间的目录数据的复制, 保证目录服务的一致性。

管理模块负责目录信息管理, 以确保用户在期望的反应时间、完整性、安全及一致性层次上取得准确的目录信息。

3.2　工作过程

LDAP 是一种面向连接的, 基于消息的协议。其工作流程如下:

(1) 客户机根据自身需求向 LDAP 服务器发送查询或操作请求。

(2) 服务器负责对目录树中条目进行必要的操作。

(3) 服务器向客户机返回一个应答。这个应答可能包含查询结果, 或包含操作出错信息, 或者是一个引用。引用(Referral) 是一种重定向机制, 表明客户所需目录服务不在本地服务器, 则向客户机返回一个更适宜服务器的 URL。

(4) 当客户机收到引用时, 向更适宜的 LDAP 服务器发送请求。

4　LDAP 目录服务器概况

在应用中部署目录服务, 必须先安装并且运行一个或多个目录服务器。目录服务器就是一系列实现目录协议并管理存储目录数据的数据库的程序, 通常目录服务器还包含有管理目录的软件。目前绝大多数目录服务器都支持 LDAP, 还有一些是完全基于 LDAP 协议的。

(1) NDS[6] (Novell Directory Services)

NDS 是随着 Netware 4 一起发布的, 属于比较早的面向企业网络的目录服务产品。它有效地将网络系统的各种资源组织到一起, 也将各种应用软件集成到同一个资源管理平台上。NDS 曾经为处于低谷的 Novell 公司带来了希望和生机。到了 Netware 5 的时候, NDS 从中分离出来, 以便支持不同的系统平台, 这就是 eDirectory。NDS 可以支持各种规模的网络环境, 当然也支持 LDAPv3。由于 NDS 是从系统软件中剥离出来的, 所以软件的可靠性较高, 也支持多个目录服务器之间的复制。

(2) Microsoft Active Directory[7] (活动目录)

在 Microsoft Windows 2000 操作系统发布之际, 呼声最高、影响最大的当属 Active Directory(活动目录) 了。活动目录成了 Windows 2000 网络系统的核心, 它存储了当前网络环境中

所有资源的信息,包括基本的个人账户信息和各种系统服务。另外,活动目录本身与系统的安全服务紧密地集成在一起,每个用户的安全信息被保存在活动目录中,而用户对系统资源的访问也是受活动目录控制的。操作系统通过活动目录控制用户的登录,活动目录与 Kerberos 认证协议结合起来,实现了单点登录(Single Sign-on) 特性。

同时,活动目录与 Microsoft Exchange Server 有一种特殊的关系:在 Windows 2000 发布之前,Microsoft 的企业群件软件 Exchange Server 已经提供了目录服务功能,并且支持 LDAP 协议;在 Windows 2000 发布以后,Exchange Server 2000 则自然地转移到活动目录之上,利用活动目录作为它的用户信息管理设施。

(3) OpenLDAP[8]

OpenLDAP 是一个通过 Internet 进行集体开发的项目(1998 年 8 月发布 1. 0 版本)。它的目标是提供一个稳定的、商业级的、功能全面的 LDAP 套件,其中包括 LDAP 服务器和一些开发工具。由于 OpenLDAP 是源码开放的,所以它在 Linux 平台上受到广泛的欢迎,当然也可以移植到其他的系统平台上,甚至 Windows 平台上。OpenLDAP 2. x 版本支持 LDAPv3,最新的版本(2. 1 版) 可以支持 LDAPv3 协议的绝大部分特性,包括一些扩展功能。

尤其值得一提的是,著名的网格项目 Globus 中就采用了 OpenLDAP 作为其资源管理中的目录服务器。

5 应用目录服务

5. 1 目录使能的应用程序

客户端可以开发目录使能的应用程序(LDAP Directory-enabled Applications) 或直接调用 LDAP Server 的 APIs 访问目录服务。

开发目录使能的应用程序来使用目录服务,把与目录服务有关的信息封装在程序模块之中,符合软件工程的信息隐蔽原则,是值得推荐的应用目录服务方式。另外,开发目录使能的应用程序还有如下优点:①降低管理数据的开销;②使目录服务适合你的组织;③节省程序开发、部署和维护的费用。

5. 2 LDAP APIS 和 LDAP 软件开发工具包 SDK

LDAP APIS 和 LDAP 软件开发工具包 SDK 封装了目录访问和更新的底层实现,使开发人员不必关注 LDAP 编程细节,使得开发目录使能的应用程序的工作简单而迅速。目前为止,大量 LDAP APIS 和 SDK 已经被开发出来,分别针对不同的程序开发语言。C API 是最早的 LDAP 版本,后来 Netscape 公司开发了 Java API, Perl 爱好者可以使用 Net: : LDAP, Python 程序员可使用 Python-ldap 模块。其中 C API 主要函数功能如表 1 所示。

表 1 C 语言 API 主要函数功能

ldap_search( )	搜寻目录条目
ldap_compare( )	检测条目中是否有所需属性
ldap_bind( )	授权一个目录服务
ldap_unbind( )	终止一个目录服务
ldap_modify( )	修改现存的目录条目
ldap_add( )	增加一个新的目录条目
ldap_delete( )	删除一个已有的目录条目
ldap_remame( )	对 LDAP 中已有的条目重命名
ldap_result( )	返回一个先前操作的结果

微软公司开发了一个专用目录服务 SDK - 活动目录服务接口 (Active Directory Services Interface, ADSI) , 主要由 Visual Basic、C 和 C + + 等使用,可以访问各种目录服务系统。不过,ADSI 主要用于 Windows 平台。

SUN 公司为 Java 软件开发了一个专用 API/ SDK- Java 命名和目录服务接口[9] (Java Naming and Directory Interface, JNDI) 。由于 Java 语言的跨平台性,使得 Java 在网络应用中有独特的优势。JNDI 包含一组 API 和一组 SPI(Service Provider Interface) 。Java 程序通过 JNDI API 存取各种 Naming 和 Directory 服务; JNDI SPI 则使得各种 Naming 和 Directory 服务透明化,允许 Java 程序通过 JNDI 来透明地使用各种目录服务。JNDI 体系构架如图 3 所示。

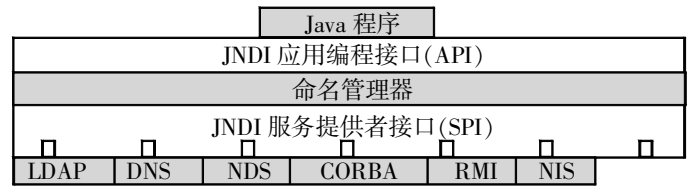


图 3 JNDI 体系架构图

6 目录服务发展的前景展望

目录服务被视为网络应用发展的下一个核心技术。随着 Internet 的飞速发展,网络上的资源正以指数方式增加,对资源的管理和查找问题日益突出。目录服务正好提供了这样一种统一而简便的网络资源管理和组织技术。无论对信息提供者、管理者或是访问者,以 LDAP 为代表的目录服务因其简单、高效、低成本,皆是一个很好的解决方案。随着网格热潮的兴起,目录服务渐渐成为研究热点,其重要性正逐渐为大家所认同。

根据网络应用和社会信息化发展的趋势,我们预测目录服务将朝着专业化、标准化两个方向发展。随着网络技术与各专业、各领域的融合,各个专业领域的各种特殊资源种类和各种对资源的特殊要求必然需要在目录中反映出来,因此目录服务需要相应扩展其功能,以满足这些领域的要求。另一方面,随着网格计算技术的蓬勃发展,要求全球共享信息资源,一个全球资源的目录服务正在形成中,为了方便地访问目录信息,目录服务标准化是必然选择。

参考文献:

[1] imothy A, Howes Ph D Mark C Smith, *et al*. Understanding and Deploying LDAP Directory Services, Second Edition[M]. USA: Addison-Wesley Pub Co., 2003. 1-260.

[2] Gerald Carter. LDAP System Administration [M]. USA: OReilly, 2003. 12-85.

[3] M Wahl, T Howes, S Kille. Lightweight Directory Access Protocol (v3) [S]. RFC 2251, 1997.

[4] M Wahl, A Coulbeck, T Howes, *et al*. Lightweight Directory Access Protocol (v3) [S]. Attribute Syntax Definitions. RFC 2252, 1997.

[5] 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): 82-84.

[6] NDS(Novell Directory Services) [EB/OL]. http://www.novell.com/.

[7] Active Directory Services Interface (ADSI) [EB/OL]. http://msdn.microsoft.com.

[8] The OpenLDAP Project Web Site[EB/OL]. http://www.openldap.org/project/.

[9] Java Naming and Directory Interface (JNDI) [EB/OL]. http://java.sun.com/products/jndi.

作者简介:

任军(1969-),男,湖北郧西人,硕士研究生,主要研究方向为分布式计算、网格资源管理及目录服务。