LDAP 目录中的检索技术研究*)

The Research for Information Retrieval in an LDAP Directory

徐建波[1] 李仁发[2] 王 妤[1]

(湘潭工学院计算机科学系 湘潭411201)[1] (湖南大学计算机科学系 长沙410001)[2]

Abstract This paper gives brief introduction to the Lightweight Directory Access Protocol(LDAP)and the concepts behind LDAP. It is organized into following sections: how Directory Services work, how LDAP Servers Organize Directories, how LOAP Clients and Servers work. At last, we provide a simple information retrieval example of an LDAP client written with the Netscape LDAP SDK for C.

Keywords Directory Services, Lightweight Directory Access Protocol, Client/Server

1 引言

一个大型企业或跨国集团会在 Internet/Intranet 上构建各式各样的应用。传统的应用中，各应用的认证系统带有各自独立的数据库，这不仅增加了硬件的投资费用，而且给维护和管理这些企业信息带来了极大的不便。一个目录服务的基本功能是采取合适的组织策略存储企业信息，以便在 Internet/Intranet 上能直接或间接地采用最快捷、最方便的方式检索这些信息。目录服务可以让网络登录、远端数据库操作、电子邮件发送、代理服务用户认证等工作都只需参考一个单一的身份识别系统。目录服务可以说是由系统安全、应用程序及其他网络服务构成的"分布式计算环境"的中心点，它实际上已经成为网络的一种基础平台软件。体现目录服务概念的一个简单例子是著名的域名服务(DNS)，一个 DNS 服务器映射一个主机名到一个 IP 地址，网络上其它计算资源都可以看成是 DNS 服务器的客户，这些客户通过更易于记忆和方便使用的主机名来检索 IP 地址。当然，一个真正的目录服务存贮的信息远不止是主机名和 IP 地址这两种信息，事实上它可以存储包括物理设备信息、企业员工信息、商务合同信息等十分复杂的企业信息。下文分析和研究了目录服务的体系结构、数据项的组织，并用 C 语言给出一个本单位的目录服务应用实例。

2 基于 LDAP 的目录服务体系结构

传统应用中要求多个数据库服务于同一目录需求，这被称为 n+1问题。典型的例子是：一个用户在两个不同的邮件系统中若具有相同的邮箱，用户的身份认证信息不得不在两个系统中分别建立和管理，这不仅造成逻辑上的混乱和数据不一致，而且也增大了安全性风险。

全球目录服务的概念用于解决上述 n+1目录服务问题，其思想是提供一个能被其它应用访问的、单一集中的目录信息仓库，为了能被各种应用访问，首先要解决的问题是它们之间的通信协议。X. 500 ISO 是为解决此问题率先提出的通信协议标准，它提供了一个开放的称为目录访问协议(DAP)的通信标准，规定了 OSI 网络七层模型上的目录项模型、通信协议、安全认证等内容。不幸的是，由于 X. 500的通信标准不是基于 TCP/IP 协议，其开发和管理十分困难。

美国 Michigan 大学开发的 LDAP 或称为轻度目录访问协议是在 X. 500标准上发展起来的，对其进行了改进和简化，由于 LDAP 提供了一个完全基于 TCP/IP 的开放的目录访问协议，因此一经推出，得以迅速推广应用，其典型产品是 Netscape Directory Server。LDAP 目录服务通过 C/S 方式实现，通常由一个 LDAP 服务器和若干 LDAP 客户组成，但在地域分布很广的领域，也可以由多个 LDAP 服务器构成一个分布式计算环境的服务器群。一个目录服务器本质上是一个通信引擎，主要由两部分构成，如图1所示。其中前端部分负责通常的网络通信，后端部分负责数据库管理，数据库中存放企业信息。目录服务器是一个多线程应用，它具有每小时处理数千万查询请求的能力，为了提高性能，目录服务器提供高速 Cache 功能，将访问最频繁的目录项数据保存在内存中。一个 LDAP 服务建立后，通过一个端口提供服务，而其它平台的客户端应用都可以通过这个端口进行数据访问。

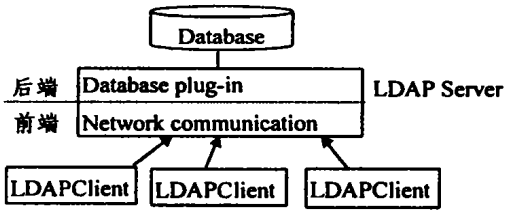

图1 LDAP 目录服务 C/S 结构

3 存贮机制

一个目录由包含描述信息的目录项组成，描述信息被存贮在目录项的属性域中。例如，目录项 jbxu 可能有下列的属性：姓名(jbxu)、邮箱号(jbxu@xtpu.org.cn)、电话号码(8290227)等，一个属性可以有超过一个以上的取值，也可以取二进制数据，如一个 JPEG 图像或一段音频格式的录音。LDAP 着眼于全球目录服务，它将企业信息组织成层次结构，从根开始向下分枝直至单独的目录项。在此树型层次结构中，较高的层表示较大的组织，树的叶子可能是表示个人的目录项或其它资源目录项，如图2所示。

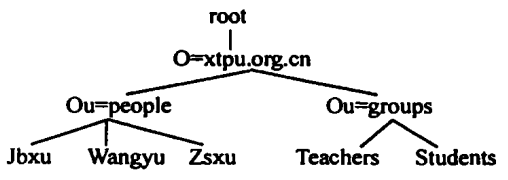

图2 目录树

(下转第110页)

*)本文得到国家自然科学基金项目资助(项目编号:69974031)。

器要做的是修改包的目的 MAC 地址，而不对 IP 包做任何修改。

在具体实现中，需要考虑怎样实现多个服务器共享一个 IP 地址而又不产生 IP 冲突。一种方法是通过配置网络接口实现：

·将负载平衡器的第一 IP(primary IP)配置为 Web 站点的 IP；

·将真实服务器的第二 IP(secondary IP)配置为 Web 站点的 IP，并配置使服务器发包时都使用第二 IP；

·用静态 ARP 缓存入口配置最近的网关，使得所有访问 Web 站点的包都首先被发送到负载平衡器。

另一种方法是禁用 ARP 协议：

·禁用真实服务器的 ARP 协议，并在真实服务器 ARP 缓存中静态固化最近网关的 MAC 地址；

·对负载平衡器的 ARP 协议不做任何的修改。

5.3 系统健壮性问题

在负载平衡系统的实现中还应该考虑 Web 服务器和负载平衡器的容错问题。对于 Web 服务器的容错，可以通过在负载平衡器上运行一个监控程序，周期性向所有服务器发询问请求探测服务器是否活跃，当发现某服务器不再活跃时，立即将它从群集中删去，调整负载平衡器的负载策略，并通知系统管理员处理。

对负载平衡器的容错通常采用热备份的方式，在负载平衡器同备份机之间采用缓存一致性策略，如果备份机未能收到负载平衡器的定时讯息，则可认为负载平衡器出现故障，便自动取代负载平衡器完成负载平衡任务。

5.4 负载平衡下的 ASP 会话

在交互式 Web 页面服务中，ASP 越来越成为企业级网络应用程序的选择。ASP 要求服务器维护 session 状态，这同群集技术的要求相违背。因为对群集下真正的负载平衡来说，每当浏览一个新页面时，服务器都潜在地丢失用户的 session 信息。为了解决这个问题，可以采用完全不使用 session，使用临时 cookies，购买第三方组件来处理 session 管理或仅对 Web 范围内的第一次点击进行负载平衡等方法来解决。

结束语 近年来，Web 服务器群集技术在实现高性能 Web 服务器方面得到了广泛的关注。作为其实现中的关键部分，负载平衡得到了广泛的研究。本文在介绍各种负载平衡技术的基础上，着重讨论了在网络端进行地址映射的负载平衡技术和该类负载平衡器的实现方法。我们看到，对于不同的应

(下转第92页)

(上接第141页)

根结点到每个目录项的路径称为 DN(Distinguished Name)，如：uid=jbxu,ou=People,o=xtpu.org.cn。在 LDAP 的 C/S 模式中，一个 LDAP 客户可以方便地完成诸如在目录树中查找、添加、修改和删除目录项等操作，比如要完成修改目录树中的某个目录项，LDAP 客户先与 LDAP 服务器通过端口(如389)建立连接，然后向 LDAP 服务器提交要修改的目录项属性值的 DN，LDAP 服务器先使用 DN 在目录树中查找到该目录项，然后完成修改其属性值的操作。

4 目录检索服务应用

Netscape Directory Server 是一个基于 LDAP 的目录服务器，可以运行在 Solaris、HP-UX、Linux、NT 操作系统上。LDAP V3的 API 功能在 RFC2251中有详细的定义，Netscape 公司的 LDAP SDK for C 和 LDAP SDK for Java 很好地实现了 RFC2251的功能。使用 LDAP SDK 提供的 API 函数，我们能开发自己的 Client 程序，完成诸如检索目录数据库的目录项，向目录数据库中添加、修改、删除和重命名目录项等操作，当然也能根据检索到目录项的属性值完成诸如身份认证这样的应用。由于 Netscape LDAP SDK for C 和 LDAP SDK for Java 完全建构在 TCP/IP 上，这样可以在支持 TCP/IP 的 UNIX、Linux、Window 等各种平台下开发客户端应用程序。下面应用 LDAP SDK for C 的几个重要的 API 函数，给出一个简单的目录项检索实例：

```
#include "ldap.h"
#define HOSTNAME "www.xtpu.edu.cn"
#define PORT 389
#define FIND_DN "uid=jbxu,ou=People,o=xtpu.org.cn"
int main( int argc,char * *argv )
{ LDAP *ld;
  LDAPMessage *result, *e;
  BerElement *ber;
  char *a, * *vals;
  ld=ldap_init( HOSTNAME,PORT)
    /* 获取一个到 LDAP Server 连接的句柄 */;
  ldap_simple_bind_s( ld,NULL,NULL );
    /* 绑定一个匿名到 LDAP server */
    /* 查找 LDAP Server 中的数据 */
  if ((ldap_search_ext_s( ld,FIND_DN,LDAP_SCOPE_BASE,
    "(objectclass = *)", NULL, 0, NULL, NULL, LDAP_NO_
      LIMIT,
    LDAP_NO_LIMIT,&result ))!= LDAP_SUCCESS ){
    perror("查找失败"); return( 1 ); }
  if (ldap_first_entry( ld,result )!= NULL ){
    printf( "找到%s:\n",FIND_DN );
    /* 在检索出的目录项中循环输出属性名和值 */
    for (a = ldap_first_attribute( ld,e,&ber ); a != NULL;
      a= ldap_next_attribute( ld,e,ber )){
     if ((vals = ldap_get_values( ld,e,a))!= NULL ){
       for (int i = 0; vals[i] != NULL; i ++ ){
         printf( "%s: %s\n",a,vals[i] );
     }ldap_value_free( vals );
    }ldap_memfree( a );
   }if ( ber != NULL ){ ber_free( ber,0 ); }
  }ldap_msgfree( result ); ldap_unbind( ld ); return( 0 );
}
```

如果在 UNIX 平台上编译和连接上述客户程序，请确保设置 LD_LIBRARY_PATH 路径变量指向库文件 libldap41.so 的位置。如果在 NT 平台连接客户程序，请将动态链接库 nsldap32v41.dll 拷到 winnt\system32目录中。

参考文献

1 Authentication Methods for LDAP. URL: ftp://ftp.isi.edu/in-notes/rfc2829.txt

2 Lightweight Directory Access Protocol (v3). URL: ftp://ftp.isi.edu/in-notes/rfc2830.txt

3 OpenLDAP Root Service. URL: ftp://ftp.isi.edu/in-notes/rfc3088.txt

4 Storing Vendor Information in the LDAP root DSE. URL: ftp://ftp.isi.edu/in-notes/rfc3045.txt

5 LDAP Authentication Password Schema. URL: ftp://ftp.isi.edu/in-notes/rfc3112.txt

6 Use of Language Codes in LDAP. URL: ftp://ftp.isi.edu/in-notes/rfc2596.txt