Linux 典藏大系

ChinaUnix

资深Linux系统管理专家，10年Linux系统管理和维护经验的总结
立足实践，用全实例讲解在Linux上构建各种最新网络服务的方法

Linux 服务器架设指南

林天峰 等编著

DVD-ROM

8.5小时多媒体语音视频讲解

另外赠送35小时Linux专题视频、Ubuntu安装文件

- 内容全面：涵盖大多数常见的Linux网络服务器的相关知识和架设方法
- 内容新颖：讲解所涉及的各种软件时都使用最新的稳定版本
- 内容深入：不仅介绍了各种服务器的架设实务，还特别分析了相关的网络协议
- 注重实践：用可操作性很强的实例讲解服务器架设，并进行了严格测试
- 讲述准确：对所讲述的内容都与原始RFC文档和官方网站进行了核实
- 视频教学：专门录制了8.5小时多媒体教学视频讲解书中的重点内容和操作

清华大学出版社

第 14 章　DNS 服务器架设与应用

DNS（Domain Name System，域名服务系统）是 Internet 上用得最频繁的服务之一，它是一个分布式数据库，组织成域层次结构的计算机和网络服务命名系统。通过它人们可以将域名解析为 IP 地址，从而使人们能够通过简单好记的域名来代替枯燥难记的 IP 地址来访问网络。本章将详细介绍 DNS 服务的基本概念、工作原理、BIND 的运行、架设和使用方法。

14.1　DNS 工作原理

DNS 是一个分布式数据库，它在本地负责控制整个分布式数据库的部分段，每一段中的数据通过客户服务器模式在整个网络上均可存取，通过采用复制技术和缓存技术使得整个数据库可靠的同时，又拥有良好的性能。下面介绍一下 DNS 的工作原理及 DNS 协议的有关情况。

14.1.1　名称解析方法

网络中为了区别各个主机，必须为每台主机分配一个唯一的地址，这个地址即称为“IP 地址”。但这些数字难以记忆，所以就采用“域名”的方式来取代这些数字了。不过最终还是必须要将域名转换为对应的 IP 地址才能访问主机，因此需要一种将主机名转换为 IP 地址的机制。在常见的计算机系统中，可以使用 3 种技术来实现主机名和 IP 地址之间的转换：Host 表、网络信息服务系统（NIS）和域名服务（DNS）。

1．Host表

Host 表是简单的文本文件，文件名一般是 hosts，其中存放了主机名和 IP 地址的映射关系，计算机通过在该文件中搜索相应的条目来匹配主机名和 IP 地址。Hosts 文件中的每一行就是一个条目，包含一个 IP 地址及与该 IP 地址相关联的主机名。如果希望在网络中加入、删除主机名或者重新分配 IP 地址，管理员所要做的就是增加、删除或修改 hosts 文件中的条目，但是要更新网络中每一台计算机上的 hosts 文件。

在 Internet 规模非常小的时候，这个集中管理的文件可以通过 FTP 发布到各个主机，每个 Internet 站点可以定期地更新其 hosts 文件的副本，并且发布主机文件的更新版本来反映网络的变化。但是，当 Internet 上的计算机迅速增加时，通过一个中心授权机构为所有

Internet 主机管理一个 hosts 文件的工作将无法进行。文件会随着时间的推移而增大，这样按当前和更新的形式维持文件以及将文件分配至所有站点将变得非常困难。

说明：虽然 Host 表目前不再广泛使用，但大部分的操作系统依旧保留。

2．NIS系统

将主机名转换为 IP 地址的另一种方案是 NIS（Network Information System，网络信息系统），它是由 Sun Microsystems 开发的一种命名系统。NIS 将主机表替换成主机数据库，客户机可以从它这里得到所需要的主机信息。然而，因为 NIS 将所有的主机数据都保存在中央主机上，再由中央主机将所有数据分配给所有的客户机，以至于将主机名转换为 IP 时的效率很低。因为在 Internet 迅猛发展的今天，没有一种办法可以用一张简单的表或一个数据库为如此众多的主机提供服务。因此，NIS 一般只用在中型以下的网络。

说明：NIS 还有一种扩展版本，称为 NIS+，提供了 NIS 主计算机和从计算机间的身份验证和数据交换加密功能

3．DNS系统

DNS 是一种新的主机名称和 IP 地址转换机制，它使用一种分层的分布式数据库来处理 Internet 上众多的主机和 IP 地址转换。也就是说，网络中没有存放全部 Internet 主机信息的中心数据库，这些信息分布在一个层次结构中的若干台域名服务器上。DNS 是基于客户/服务器模型设计的。本质上，整个域名系统以一个大的分布式数据库方式工作。具有 Internet 连接的企业网络都可以有一个域名服务器，每个域名服务器包含有指向其他域名服务器的信息，结果是这些服务器形成了一个大的协调工作的域名数据库。

14.1.2　DNS 组成

每当一个应用需要将域名翻译成为 IP 地址时，这个应用便成为域名系统的一个客户。这个客户将待翻译的域名放在一个 DNS 请求信息中，并将这个请求发给域名空间中的 DNS 服务器。服务器从请求中取出域名，将它翻译为对应的 IP 地址，然后在一个回答信息中将结果返回给应用。如果接到请求的 DNS 服务器自己不能把域名翻译为 IP 地址，将向其他 DNS 服务器查询。整个 DNS 域名系统由以下 3 个部分组成。

1．DNS域名空间

指定用于组织名称的域的层次结构，它如同一棵倒立的树，层次结构非常清晰，如图 14-1 所示。根域位于顶部，紧接着在根域的下面是几个顶级域，每个顶级域又可以进一步划分为不同的二级域，二级域再划分出子域，子域下面可以是主机也可以是再划分的子域，直到最后的主机。在 Internet 中的域是由 InterNIC 负责管理的，域名的服务则由 DNS 来实现。

2．DNS服务器

DNS 服务器是保持和维护域名空间中数据的程序。由于域名服务是分布式的，每一个 DNS 服务器含有一个域名空间自己的完整信息，其控制范围称为区（Zone）。对于本区内的请求由负责本区的 DNS 服务器解释，对于其他区的请求将由本区的 DNS 服务器与负责该区的相应服务器联系。

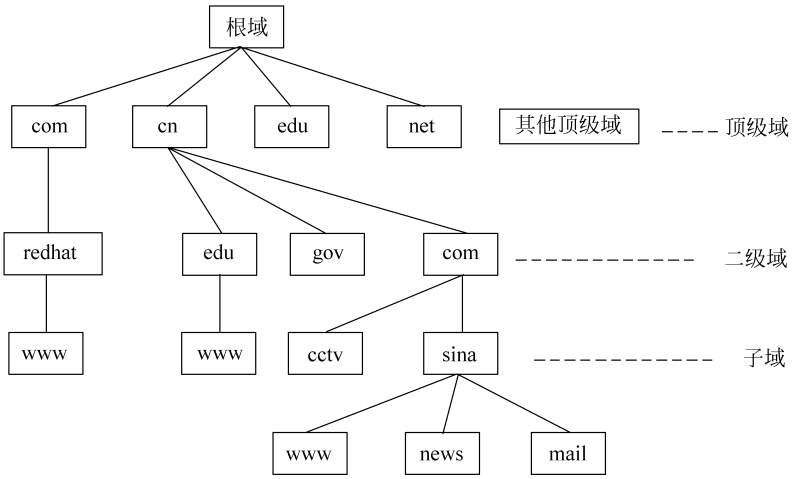

图 14-1　DNS 域名空间

3．解析器

解析器是简单的程序或子程序，它从服务器中提取信息以响应对域名空间中主机的查询，用于 DNS 客户端。

14.1.3　DNS 查询的过程

当客户端程序要通过一个主机名称来访问网络中的一台主机时，它首先要得到这个主机名称所对应的 IP 地址，因为 IP 数据报中允许放置的是目地主机的 IP 地址，而不是主机名称。可以从本机的 hosts 文件中得到主机名称所对应的 IP 地址，但如果 hosts 文件不能解析该主机名称时，只能通过向客户机所设定 DNS 服务器进行查询了。

说明：在 UNIX 系统中，可以设置 hosts 和 dns 的使用次序。

可以以不同的方式对 DNS 查询进行解析。第一种是本地解析，就是客户端可以使用缓存信息就地应答，这些缓存信息是通过以前的查询获得的；第二种是直接解析，就是直接由所设定的 DNS 服务器解析，使用的是该 DNS 服务器的资源记录缓存或者其权威回答（如果所查询的域名是该服务器管辖的）；第三种是递归查询，即设定的 DNS 服务器代表客户

端向其他 DNS 服务器查询，以便完全解析该名称，并将结果返回至客户端。第四种是迭代查询，即设定的 DNS 服务器向客户端返回一个可以解析该域名的其他 DNS 服务器，客户端再继续向其他 DNS 服务器查询。

1．本地解析

本地解析的过程如图 14-2 所示。客户机平时得到的 DNS 查询记录都保留在 DNS 缓存中，客户机操作系统上都运行着一个 DNS 客户端程序。当其他程序提出 DNS 查询请求时，这个查询请求要传送至 DNS 客户端程序。DNS 客户端程序首先使用本地缓存信息进行解析，如果可以解析所要查询的名称，则 DNS 客户端程序就直接应答该查询，而不需要向 DNS 服务器查询，该 DNS 查询处理过程也就结束了。

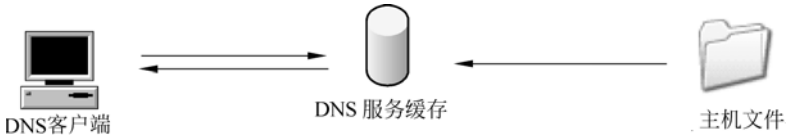

图 14-2　本地解析

2．直接解析

如果 DNS 客户端程序不能从本地 DNS 缓存回答客户机的 DNS 查询，它就向客户机所设定的局部 DNS 服务器发一个查询请求，要求局部 DNS 服务器进行解析。如图 14-3 所示，局部 DNS 服务器得到这个查询请求，首先查看一下所要求查询的域名是不是自己能回答的，如果能回答，则直接给予回答，如是不能回答，再查看自己的 DNS 缓存，如果可以从缓存中解析，则也是直接给予回应。

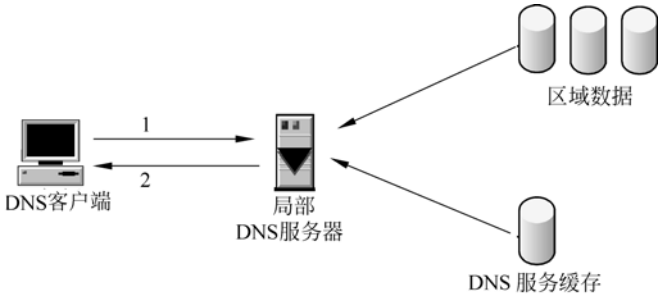

图 14-3　局部 DNS 服务器解析

3．递归解析

当局部 DNS 服务器自己不能回答客户机的 DNS 查询时，它就需要向其他 DNS 服务器进行查询。此时有两种方式，如图 14-4 所示的是递归方式。局部 DNS 服务器自己负责向其他 DNS 服务器进行查询，一般是先向该域名的根域服务器查询，再由根域名服务器一级级向下查询。最后得到的查询结果返回给局部 DNS 服务器，再由局部 DNS 服务器返回给

客户端。

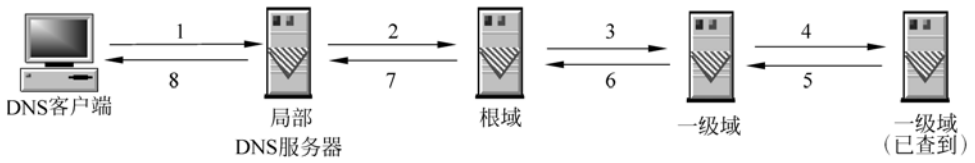

图 14-4　DNS 解析的递归方式

4．迭代解析

当局部 DNS 服务器自己不能回答客户机的 DNS 查询时，也可以通过迭代查询的方式进行解析，如图 14-5 所示。局部 DNS 服务器不是自己向其他 DNS 服务器进行查询，而是把能解析该域名的其他 DNS 服务器的 IP 地址返回给客户端 DNS 程序，客户端 DNS 程序再继续向这些 DNS 服务器进行查询，直到得到查询结果为止。

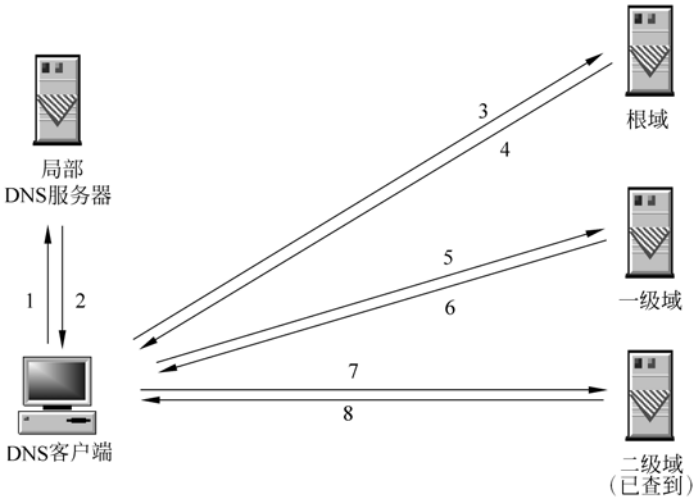

图 14-5　DNS 解析的迭代方式

以上介绍了 DNS 解析的 4 种方式，下面看一下 DNS 报文的格式。

14.1.4　DNS 报文格式

DNS 客户端与 DNS 服务器进行交互时，需要传送各种各样的数据报，这些数据报的总体格式如图 14-6 所示。DNS 报文由 12 字节长的首部和 4 个长度可变的字段组成。标识字段由客户程序设置并由服务器返回结果，客户程序通过它来确定响应与查询是否匹配。标识字段包含了以下内容。

- 定义是查询报文还是响应报文；

- 查询类型是标准查询、反向查询还是服务器状态请求；
- 是否是权威回答；
- 查询方式是递归查询还是迭代查询；
- 是否支持递归查询；
- 查询是否有差错或要查的域名不存在。

问题部分中每个问题的格式如图 14-7 所示，通常只有一个问题。查询名是要查找的名字，它是一个或多个标识符的序列。每个标识符以首字节的计数值来说明随后标识符的字节长度，每个名字以最后字节为 0 结束，长度为 0 的标识符是根标识符。计数字节的值必须是 0～63 的数，因为标识符的最大长度仅为 63。与其他常用报文格式不一样的是，该字段无需以整 32bit 边界结束，即无需填充字节。图 14-8 显示了如何存放域名 www.wzvtc.cn。其中，3、5、2 表示后续字符的个数。

图 14-6　DNS 数据报的总体格式

图 14-7　DNS 数据包问题部分格式

3	w	w	w	5	w	z	v	t	c	2	c	n	0

图 14-8　域名 www.wzvtc.cn 的表示方法

每个问题都有一个查询类型，而每个应答（也称为一条资源记录）也有一个应答类型。大约有 20 个不同的类型值，有一些目前已经过时，表 14-1 列出了常用的一些值。其中有两种可以用于查询类型：一种是 A 类型，表示期望获得查询名的 IP 地址。还有一种是 PTR 类型，表示请求获得一个 IP 地址对应的域名，也称为指针查询。查询报文格式中的查询类通常是 1，指互联网地址。

注意：某些站点也支持其他非 IP 地址查询，此时查询类将是其他数值。

表 14-1　DNS问题和响应的类型值和查询类型值

名　字	数　值	描　述	名　字	数　值	描　述
A	1	IP 地址	HINFO	13	主机信息
NS	2	名字服务器	MX	15	邮件交换记录
CName	5	规范名称	AXFR	252	对区域转换的请求
PTR	12	指针记录	*或 ANY	255	对所有记录的请求

DNS 报文一般格式中的最后 3 个字段是回答字段、授权字段和附加信息字段，它们均采用一种称为资源记录 RR（Resource Record）的相同格式。图 14-9 显示了资源记录的格式，各个字段的含义如下：

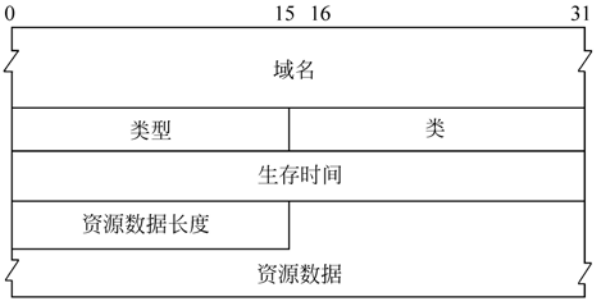

图 14-9　DNS 资源记录格式

- 域名是记录中资源数据对应的名字，它的格式和前面介绍的查询名字段格式（图 14-8）相同。
- 类型说明 RR 的类型码，它的值的取值及含义见表 14-1。
- 类通常为 1，指 Internet 数据。
- 生存时间字段是客户程序保留该资源记录的秒数，资源记录通常的生存时间值为 2 天。
- 资源数据长度说明资源数据的数量。
- 资源数据的格式依赖于类型字段的值，对于类型 1（A 记录）资源数据是 4 字节的 IP 地址。

14.1.5　实际的 DNS 报文数据

在客户机利用域名访问一台主机时，首先要向所设的 DNS 服务器发送查询报文，以获得该域名所对应的 IP 地址，DNS 服务器要根据具体情况返回给客户端应答。下面通过抓包工具 Ethereal 捕获这些数据包，并进行观察，以便更深入地理解 DNS 协议。

假设在一台 IP 地址为 10.10.91.252 的客户机上执行 ping www.baidu.cn 命令，成功执

行完成后，用 Ethereal 抓到的数据包如图 14-10 所示。在命令执行前，先用 ipconfig /flushdns 清除本地的 DNS 缓存，否则客户机有可能不发送 DNS 报文而是直接从 DNS 缓存中得到 IP 地址。

从图 14-10 可以看出，客户机 10.10.91.252 在 ping 一个域名时，要先通过 DNS 查询获得该域名所对应的 IP 地址，因此向所设的 DNS 服务器 10.10.1.29 发送了数据包 1，可以看出查询类型为 A。DNS 服务器 10.10.1.29 通过数据包 2 告诉客户机，www.baidu.cn 域名所对应的 IP 地址是 220.181.6.18。于是，ping 命令向 220.181.6.18 发送了数据包 3、5、7 和 9 四个 ICMP 请求，而 220.181.6.18 则通过 4、6、8 和 10 四个数据包进行了回复。

从数据包 2 还可以看出，DNS 服务器返回了一条 CNAME 资源记录和两条 A 资源记录，客户端根据自己的算法取出了第一条 A 资源记录中所包含的 IP 地址。另外，从图 4-10 中所示窗口的中间部分还可以看出，DNS 报文是通过 UDP 协议发送的，服务器的端口号是 53 号。

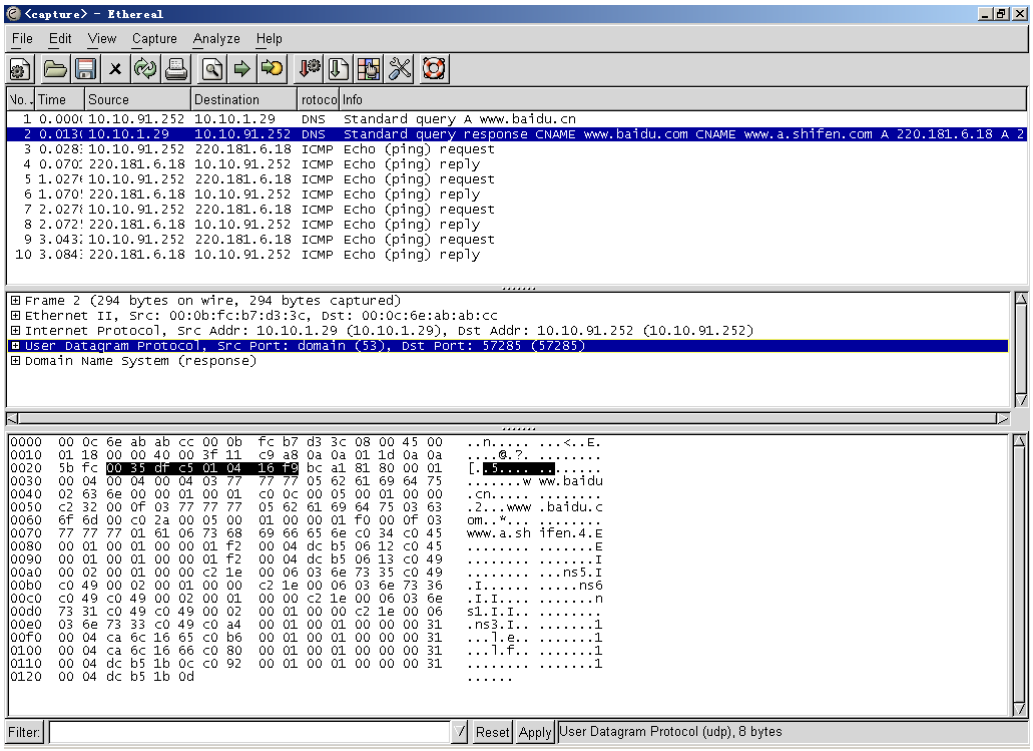

图 14-10　Ethereal 抓到的 DNS 数据包

说明：以上是 DNS 客户端与服务器的报文交互情况。实际上，DNS 服务器为了解析 www.baidu.cn 域名，可能还需要与其他 DNS 服务器进行交互，这些交互数据包不能在客户机上捕获。

14.2　BIND 的安装与运行

Bind 是最知名的域名服务器软件，它完整地实现了 DNS 协议规定的各种功能，可以在各种主流的操作系统平台上运行，并且被作为许多供应商的 UNIX 标准配置封装在产品中。下面介绍一下有关 Bind 服务器软件的安装与运行方法。

14.2.1　BIND 简介

Linux 系统下架设 DNS 服务器通常是使用 Bind 程序来实现的。Bind 是 Berkeley Internet Name Domain Service 的简写，是一款架设 DNS 服务器的开放源代码软件。Bind 原本是美国 DARPA 资助伯克里大学开设的一个研究生课题，后来经过多年的变化发展，已经成为世界上使用最为广泛的 DNS 服务器软件，目前 Internet 上绝大多数的 DNS 服务器都是用 Bind 来架设的。

Bind 经历了第 4 版、第 8 版和最新的第 9 版，第 9 版修正了以前版本中的许多错误，并提升了执行时的效能。Bind 能够运行在当前大多数的操作系统系统平台之上。目前 Bind 软件由 ISC（Internet Software Consortium，因特网软件联合会）这个非赢利性机构负责开发和维护。ISC 的官方网站域名为 http://www.isc.org/，包含了 Bind 的最新错误修复和更新。

14.2.2　BIND 的获取与安装

在 Red Hat Enterprise Linux 5 下安装 BIND 服务器可以有两种方式，一种是源代码方式安装，一种是 RPM 软件包方式安装。源代码可以从 ftp://ftp.isc.org 处下载，目前最新的版本是 9.5.0 版，文件名是 bind-9.5.0-P2.tar.gz。RHEL 5 自带的 BIND 版本是 9.3.9 版，文件名是 bind-9.3.3-7.el5.i386.rpm，在发行版的第 2 张盘上。

先看一下 RPM 方式安装。如果安装 RHEL 5 系统的时候没有选择安装 bind-9.3.3-7.el5 包，需要从第二张安装光盘把相应文件复制到当前目录以后，再用以下命令安装。

```
# rpm -ivh  bind-9.3.3-7.el5.i386.rpm
```

如果安装成功，会出现以下提示。

```
warning: bind-9.3.3-7.el5.i386.rpm: Header V3 DSA signature: NOKEY, key ID
37017186
Preparing...          ########################################### [100%]
   1:bind             ########################################### [100%]
#
```

再输入以下命令，可以看到安装后的文件分布情况。

```
# rpm -ql bind-9.3.3-7.el5
```

其中比较重要的文件分布如下：

- /etc/rc.d/init.d/named：Bind 开机自动启动时所用的启动脚本。
- /usr/sbin/bind-chroot-admin：启用或禁用 chroot 功能的命令。
- /usr/sbin/named：named 进程的程序文件。
- /usr/sbin/rndc：远程控制 named 进程运行的工具。
- /usr/sbin/rndc-confgen：产生 rndc 密钥的工具。
- /usr/share/doc/bind-9.3.3：该目录下安装了 BIND 的帮助文档和例子文件。
- /usr/share/man/man5：这个目录下安装了 BIND 的手册页。
- /usr/share/man/man8：这个目录下也安装了 BIND 的手册页。
- /var/named：Bind 配置文件的默认存放目录（不包含主配置文件）。
- /var/run/named：named 进程 PID 文件的存放目录。

named 进程是以 named 用户的身份运行的，因此，操作系统中要事先存在这个用户。

说明：当默认安装 RHEL 5 时，named 用户已经创建，如是由于某种原因该用户不存在了，需要重新创建。

如果采用源代码方式安装，则从 ftp://ftp.isc.org/isc/bind9/9.5.0-P2/处下载 Bind 的最新版 9.5.0 版的源代码文件 bind-9.5.0-P2.tar.gz，文件复制到当前目录后，使用以下命令进行安装。

```
# rpm –e bind-9.3.3-7.el5          //如果安装了 bind 9.3.3 包，则先拆除
# tar xvzf bind-9.5.0-P2.tar.gz  //解压源代码文件包，到 bind-9.5.0-P2 目录中
# cd bind-9.5.0-P2
# ./configure
# make                             //编译连接，产生可执行文件
# make install                     //把文件安装到相应的目录
```

当练习测试时，可选择上述两种安装方式中的一种，本章后面的例子是以 RPM 安装方式为基础进行讲解的。

14.2.3　BIND 的简单配置与运行

与其他服务器相比，BIND 的配置文件结构要复杂得多，而且在配置文件不正确的情况下，BIND 将无法运行。为了使 BIND 能初步得到运行，下面先提供一套最简单的配置文件，使得 BIND 能正常地运行起来，并具有初步的域名解析功能，具体内容的解释见 14.3 节。

（1）在/etc 目录下建立 BIND 的主配置文件 named.conf，内容如下所示。

```
# vi /etc/named.conf
options {
  directory "/var/named";
  pid-file "/var/named/named.pid";
  forwarders { 61.153.177.196;  61.153.177.197; };
  allow-query { any; };
};
zone "." IN {
```

```
type hint;
file "named.root";
};
zone "0.0.127.in-addr.arpa" IN {
 type master;
 file "named.local";
allow-update { none; };
};
zone  "wzvtc.cn" IN {
   type master;
   file "named.wzvtc.cn";
   allow-update { none; };
};
zone  "1.10.10.in-addr.arpa" IN {
       type master;
       file "named.1.10.10";
       allow-update  { none; };
};
```

（2）在/var/named 目录下要建立在主配置文件中指定的 4 个区文件，文件名分别是named.root、named.local、named.wzvtc.cn 和 named.1.10.10。其中，named.root 可以由以下命令生成。

```
# dig > /var/named/named.root
```

/var/named/named.local 文件的内容如下：

```
# vi  /var/named/named.local
$TTL 3h
0.0.127.in-addr.arpa. IN SOA terminator.wzy.edu. ltf@wzvtc.cn. (
                      1       ; Serial
                      3h      ; Refresh after 3 hours
                      1h      ; Retry after 1 hour
                      1w      ; Expire after 1 week
                      1h )    ; Negative caching TTL of 1 hour
1.0.0.127.in-addr.arpa.  IN  PTR  localhost.
```

/var/named/named.wzvtc.cn 文件的内容如下：

```
# vi  /var/named/named.wzvtc.cn
$TTL 3h
wzvtc.cn. IN SOA ns.wzvtc.cn. ltf@wzvtc.cn. (
                      1       ; Serial
                      3h      ; Refresh after 3 hours
                      1h      ; Retry after 1 hour
                      1w      ; Expire after 1 week
                      1h )    ; Negative caching TTL of 1 hour

wzvtc.cn.       IN  NS   ns.wzvtc.cn.
wzvtc.cn.       IN  MX   10  mail
ns.wzvtc.cn.    IN  A    10.10.1.29
mail            IN  A    10.10.1.6
www             IN  A    10.10.1.3
oa              IN  CNAME  www
lib             IN  A     10.10.1.8
gsx             IN  A     221.224.2.234
```

/var/named/named.1.10.10 文件的内容如下：

```
# vi /var/named/named.1.10.10
$TTL 3h
1.10.10.in-addr.arpa. IN SOA ns.wzvtc.cn. ltf@wzvtc.cn.(
                      1       ; Serial
                      3h      ; Refresh after 3 hours
                      1h      ; Retry after 1 hour
                      1w      ; Expire after 1 week
                      1h )    ; Negative caching TTL of 1 hour

1.10.10.in-addr.arpa.       IN   NS  ns.wzvtc.cn.
29.1.10.10.in-addr.arpa.    IN   PTR  ns.wzvtc.cn.
6.1.10.10.in-addr.arpa.     IN   PTR  mail.wzvtc.cn.
8.1.10.10. in-addr.arpa.    IN   PTR  lib.wzvtc.cn.
```

（3）为了使 BIND 的运行不受 chroot 功能的影响，先用以下命令禁用 chroot。

```
# bind-chroot-admin -d
```

（4）用以下命令启动 named 进程，加-g 选项的目的是为了显示启动过程的详细信息，以便出错时能及时发现原因。

```
# /usr/sbin/named  -g  &
[1] 3849
# 13-Nov-2008 18:18:54.336 starting BIND 9.3.3rc2 -g
13-Nov-2008 18:18:54.337 found 1 CPU, using 1 worker thread
13-Nov-2008 18:18:54.339 loading configuration from '/etc/named.conf'
13-Nov-2008 18:18:54.340 listening on IPv4 interface lo, 127.0.0.1#53
13-Nov-2008 18:18:54.341 listening on IPv4 interface eth0, 10.10.1.29#53
13-Nov-2008 18:18:54.343 command channel listening on 127.0.0.1#953
13-Nov-2008 18:18:54.344 command channel listening on ::1#953
13-Nov-2008 18:18:54.344 ignoring config file logging statement due to -g
option
13-Nov-2008 18:18:54.345 zone 1.10.10.in-addr.arpa/IN: loaded serial 1
13-Nov-2008 18:18:54.346 zone wzvtc.cn/IN: loaded serial 1
13-Nov-2008 18:18:54.346 running
```

（5）用以下命令查看 named 进程是否已正常启动。

```
# ps -eaf|grep named
root     3849  3469  0 18:18 pts/0    00:00:00 /usr/sbin/named -g
root     3862  3469  0 18:23 pts/0    00:00:00 grep named
#
```

（6）由于 DNS 采用的是 UDP 协议，监听的是 53 号端口，可以用下面的命令进一步验证 named 是否已正常工作。

```
# netstat -an | grep :53
tcp       0     0 10.10.1.29:53         0.0.0.0:*                   LISTEN
tcp       0     0 127.0.0.1:53          0.0.0.0:*                   LISTEN
udp       0     0 10.10.1.29:53         0.0.0.0:*
udp       0     0 127.0.0.1:53          0.0.0.0:*
```

可见，UDP53 号端口已经打开，同时也可以看到，TCP 的 53 号端口也是处于监听状态的，这个端口主要是用于 DNS 服务器之间传送域数据。

（7）上述步骤完成后，再检查一下防火墙是不是开放了 TCP 和 UDP 的 53 号端口。如

果还没开放，可输入以下命令打开。

```
#iptables -I INPUT -p tcp --dport 53 -j ACCEPT
#iptables -I INPUT -p udp --dport 53 -j ACCEPT
```

（8）最后，可以在本机或网络中的其他计算机进行以下测试（“>”后面是用户输入的内容）。

```
C:\ >nslookup
Default Server:  ns.wzvtc.cn
Address:  10.10.1.2      # 原来默认的 DNS 服务器是 10.10.1.2

> server 10.10.1.29      # 把 DNS 服务器设为 10.10.1.29
Default Server:  [10.10.1.29]
Address:  10.10.1.29

> www.wzvtc.cn           # 查询 www.wzvtc.cn 的 IP 地址
Server:  [10.10.1.29]
Address:  10.10.1.29

Name:    www.wzvtc.cn
Address:  10.10.1.3      # DNS 服务器回复 www.wzvtc.cn 的 IP 地址是 10.10.1.3

> mail.wzvtc.cn          # 再查询 mail.wzvtc.cn 的 IP 地址
Server:  [10.10.1.29]
Address:  10.10.1.29

Name:    mail.wzvtc.cn
Address:  10.10.1.6      # DNS 服务器回复 mail.wzvtc.cn 的 IP 地址是 10.10.1.6

> www.baidu.cn           # 查询其他域的域名 www.baidu.cn
Server:  [10.10.1.29]
Address:  10.10.1.29

Non-authoritative answer:              # 表示是非权威的回答
Name:    www.a.shifen.com
Addresses:  220.181.6.18, 220.181.6.19 # DNS 服务器回复 www.baidu.cn 的 IP 地
                                         址，有两个
Aliases:  www.baidu.cn, www.baidu.com

> exit

C:\ >
```

从以上测试可以看出，DNS 已经能正常地工作，能解析 wzvtc.cn 区中的域名，还能通过其他的 DNS 服务器解析互联网上的所有域名。

14.2.4　chroot 功能

chroot 是 Change Root 的缩写，它可以将文件系统中某个特定的子目录作为进程的虚拟根目录，即改变进程所引用的“/”根目录位置。chroot 对进程可以使用的系统资源、用户权限和所在目录进行严格控制，程序只在这个虚拟的根目录及其子目录具有权限，一旦

离开该目录就没有任何权限了，所以也将 chroot 称为“jail 监禁”。

说明：在 Vsftpd 服务器架设中，也有几个关于 chroot 的配置选项，可以把操作系统中的某一个目录（通常是该用户的主目录）作为用户的根目录，用户登录到 FTP 服务器时，看到的根目录并不是服务器上真正的根目录，而是其他目录。用户不能访问除这个目录以外的任何文件，即就是把用户监禁在某一目录中，用户的任何操作仅对这个目录有效，不会影响到系统和其他用户的文件。

早期 Linux 服务都是以 root 权限启动和运行的，随着技术的发展，各种服务变得越来越复杂，导致 BUG 和漏洞也越来越多。黑客利用服务的漏洞入侵系统，就能获得 root 级别的权限，从而可以控制整个系统。为了减缓这种攻击所带来的负面影响，现在的服务器软件通常设计成以 root 权限启动，然后服务器进程自行放弃 root 权限，再以某个低权限的系统账号来运行进程。这种方式的好处在于该服务被攻击者利用漏洞入侵时，由于进程权限比较低，攻击者得到的访问权限是基于这个较低权限的，因此对系统造成的危害比以前减轻了许多。

基于同样的道理，chroot 的使用并不能说是让程序本身更安全了，它跟没有 chroot 的程序比较，依然有着同样多的 bug 和漏洞，依然会被攻击者利用这些 bug 和漏洞进行攻击并得逞。但由于程序本身的权限被严格限制了，因此攻击者无法造成更大的破坏，也无法夺取操作系统的最高权限。DNS 服务器主要是用于域名解析，需要面对来自网络各个位置的大量访问，并且一般不限制来访者的 IP，因此，存在的安全隐患和被攻击的可能性相当大，使用 chroot 功能也就特别地有意义了。

在 Red Hat Enterprise Linux 5 下，chroot 的安装包文件名为 bind-chroot-9.3.3-7.el5.i386.rpm，在第二张安装盘中。把安装文件复制到当前目录后，输入以下命令进行安装。

```
# rpm  -ivh  bind-chroot-9.3.3-7.el5.i386.rpm
```

当成功安装 chroot 后，named 的虚拟根目录变为/var/named/chroot，即以后运行 named 进程时，会把这个目录当作根目录。同时，这个虚拟根目录下还自动创建了 dev、etc 和 var 3 个目录，分别对应实际根目录下的同名目录。另外，安装 chroot 时，还会自动把实际根目录下的这 3 个目录中的配置文件都复制到虚拟根目录下对应的 3 个目录中，例如，/etc/named.conf 会复制到/var/named/chroot/etc。因此，以后编辑 named 的配置文件时，要注意其存放的目录位置。

当 BIND 包安装完后，会在/usr/sbin 目录下出现 bind-chroot-admin 文件，这是一个与 chroot 有关的命令文件，利用它，可以禁用或启用 chroot 功能，也可以使虚拟根目录下的 named 配置文件与实际根目录下的 named 配置文件进行同步。其命令格式如下所示。

```
# bind-chroot-admin
Usage:
  -e | --enable:   enable the bind-chroot environment
  -d | --disable:  disable the bind-chroot environment
  -s | --sync:    sync files between the bind chroot and / environments,
                    so they are correct for the current state of the
                    bind-chroot
```

```
                    (enabled / disabled)
  $BIND_CHROOT_PREFIX, default /var/named/chroot, is the location of the
  chroot.
  $BIND_DIR, default /var/named, is the default un-chrooted bind directory.
```

在 bind-chroot-admin 命令后加-e 选项可以启用 chroot 功能，加-d 选项禁用 chroot 功能，加-s 选项同步配置文件。在实际工作中，最好要启用 chroot 功能，可以使服务器的安全性能得到提高，但在本章中，为了讲解和实验的方便，以及减少出错的可能性，禁用了 chroot 功能。

14.2.5　使用 rndc

rndc 是 BIND 安装包提供的一种控制域名服务运行的工具，它可以运行在其他计算机上，通过网络与 DNS 服务器进行连接，然后根据管理员的指令对 named 进程进行远程控制，此时，管理员不需要 DNS 服务器的根用户权限。

使用 rndc 可以在不停止 DNS 服务器工作的情况进行数据的更新，使修改后的配置文件生效。在实际情况下，DNS 服务器是非常繁忙的，任何短时间的停顿都会给用户的使用带来影响。因此，使用 rndc 工具可以使 DNS 服务器更好地为用户提供服务。

rndc 与 DNS 服务器实行连接时，需要通过数字证书进行认证，而不是传统的用户名/密码方式。在当前版本下，rndc 和 named 都只支持 HMAC-MD5 认证算法，在通信两端使用共享密钥。rndc 在连接通道中发送命令时，必须使用经过服务器认可的密钥加密。为了生成双方都认可的密钥，可以使用 rndc-confgen 命令产生密钥和相应的配置，再把这些配置分别放入 named.conf 和 rndc 的配置文件 rndc.conf 中，具体操作步骤如下所示。

（1）执行 rndc-confgen 命令，得到密钥和相应的配置。

```
# rndc-confgen
# Start of rndc.conf
key "rndckey" {
  algorithm hmac-md5;
  secret "TKuaJSEo58zohJBfrdF7dQ==";
};

options {
  default-key "rndckey";
  default-server 127.0.0.1;
  default-port 953;
};
# End of rndc.conf

# Use with the following in named.conf, adjusting the allow list as needed:
# key "rndckey" {
#      algorithm hmac-md5;
#      secret "TKuaJSEo58zohJBfrdF7dQ==";
# };
#
# controls {
#      inet 127.0.0.1 port 953
#            allow { 127.0.0.1; } keys { "rndckey"; };
# };
```

```
# End of named.conf
```

（2）在/etc 目录下创建 rndc.conf 文件，根据提示输入上述输出中不带注释的内容。

```
# vi  /etc/rndc.conf
key "rndckey" {
  algorithm hmac-md5;
  secret "TKuaJSEo58zohJBfrdF7dQ==";
};

options {
  default-key "rndckey";
  default-server 127.0.0.1;
  default-port 953;
};
```

（3）根据提示，把下列内容放入原有的/etc/named.conf 文件后面。

```
key "rndckey" {
     algorithm hmac-md5;
     secret "TKuaJSEo58zohJBfrdF7dQ==";
};

controls {
     inet 127.0.0.1 port 953
          allow { 127.0.0.1; } keys { "rndckey"; };
};
```

（4）重启 named 进程后，就可以使用 rndc 工具对 named 进行控制了。例如，下面的命令可以使 named 重新装载配置文件和区文件。

```
# rndc reload
server reload successful
#
```

此外，所有 rndc 支持的命令及帮助信息可以通过不带参数的 rndc 命令显示。

```
[root@localhost named]# rndc
Usage: rndc [-c config] [-s server] [-p port]
       [-k key-file ] [-y key] [-V] command

command is one of the following:
  reload      Reload configuration file and zones.
  reload zone [class [view]]
  ...
  status      Display status of the server.
  recursing   Dump the queries that are currently recursing
  (named.recursing)
  *restart    Restart the server.

* == not yet implemented
Version: 9.3.3rc2
```

可以看到，rndc 提供了非常丰富的命令，可以让管理员在不重启 named 进程的情况下，完成大部分的 DNS 服务器管理工作。

说明：rndc 命令后面可以跟“-s”和“-p”选项连接到远程 DNS 服务器，以便对远程

DNS 服务器进行管理，但此时双方的密钥要一致才能正常连接。

14.3　BIND 的配置

14.2 节提供了一个简单的配置例子，使得 BIND 运行后具有初步的 DNS 服务器功能。本节先详细解释各种配置选项的含义，再通过几个例子使读者能配置相对复杂的 DNS 服务器。与其他大多数服务器不同的是，BIND 配置需要较多的配置文件，而不是所有的配置都集中在一个配置文件里，因此相对要复杂些。

14.3.1　BIND 的主配置文件

BIND 主配置文件由 named 进程运行时首先读取，文件名为 named.conf，默认在/etc 目录下。该文件只包括 Bind 的基本配置，并不包含任何 DNS 的区域数据。安装 DNS 服务后，安装程序不会自动生成/etc/named.conf 文件，用户需要自行创建或将/usr/share/doc/bind-9.3.3/sample/etc/named.conf 范本文件复制为/etc/named.conf。

named.conf 配置文件由语句与注释组成，每一条主配置语句均有自己的选项参数。这些选项参数以子语句的形式组成，并包含在花括号内，作为主语句的组成部分。每一条语句，包括主语句和子语句，都必须以分号结尾。注释符号可以使用类似于 C 语言中的块注释“/*”和“*/”符号对，以及行注释符“//”或“#”。BIND 9 支持的主配置语句及功能如表 14-2 所示。

表 14-2　BIND 9 主配置语句名称

主配置语句名称	功　能
acl	定义一个访问控制列表，用于以后对列表中的 IP 进行访问控制
controls	定义有关本地域名服务器操作的控制通道，这些通道被 rndc 用来发送控制命令
include	把另一个文件中的内容包含进来做为主配置文件的内容
key	定义一个密匙信息，用于通过 TSIG 进行授权和认证的配置中
logging	设置日志服务器，以及日志信息的发送位置
options	设置 DNS 服务器的全局配置选项

续表

主配置语句名称	功　能
server	定义了与远程服务器交互的规则
trusted-keys	定义信任的 DNSSED 密匙
view	定义一个视图
zone	定义一个区域

1．acl语句

acl 主配置语句用于定义一个命名的访问列表，里面包含了一些用 IP 表示的主机，这个访问列表可以在其他语句使用，表示其所定义的主机。其格式如下：

```
acl acl-name {
    address_match_list
};
```

address_match_list 表示 IP 地址或 IP 地址集。其中，none、any、localhost 和 localnets 这 4 个内定的关键字有特别含义，分别表示没有主机、任何主机、本地网络接口 IP 和本地子网 IP。一个具体的例子如下所示。

```
acl "someips" {             //定义一个名为 someips 的 ACL
  10.0.0.1; 192.168.23.1; 192.168.23.15;      //包含 3 个单个 IP
};
acl "complex" {             //定义一个名为 complex 的 ACL
  "someips";                //可以包含其他 ACL
  10.0.15.0/24;             //包含 10.0.15.0 子网中的所有 IP
  !10.0.16.1/24;            //非 10.0.16.1 子网的 IP
  {10.0.17.1;10.0.18.2;};   //包含了一个 IP 组
  localhost;                //本地网络接口 IP（含实际接口 IP 和 127.0.0.1）
 };
zone "example.com" {
  type slave;
  file "slave.example.com";
  allow-notify {"complex";};    //在此处使用了前面定义的 complex 访问列表
};
```

2．controls语句

controls 主语句定义有关本地域名服务器操作的控制通道，这些通道被 rndc 用来发送控制命令。在上节的例子 named.conf 配置文件中有以下语句，现解释如下：

```
controls {
    inet 127.0.0.1 port 953    //在 127.0.0.1 接口的 953 号端口进行监听
       allow { 127.0.0.1; }    //只接受 127.0.0.1 的连接，即只有在本机使用 rndc
                               //才能对 named 进行控制
       keys { "rndckey"; };    //使用名为 rndckey 的密钥才能访问
};
```

3．include语句

include 主语句表示把另一个文件的内容包含进来，作为 named.conf 文件的配置内容，其效果与把那个文件的内容直接输入 named.conf 时一样。之所以这样做，一是为了简化一些分布式的 named.conf 文件的管理，此时，每个管理员只负责自己所管辖的配置内容。二是为了安全，因为可以把一些密钥放在其他文件，不让无关的人查看。

4．key语句

key 主语句定义一个密匙，用于 TSIG 授权和认证。它主要在与其他 DNS 服务器或 rndc 工具通信时使用，可以通过运行 rndc-confgen 命令产生。在 14.2.5 小节的例子 named.conf 配置文件中有以下语句，现注释如下：

```
key "rndckey" {              //定义一个密钥，名为 rndckey
      algorithm hmac-md5;   //采用 hmac-md5 算法，这也是目前唯一支持的加密算法
      secret "TKuaJSEo58zohJBfrdF7dQ==";    //密钥的具体数据
 };
```

5．logging语句

logging 是有关日志配置的主语句，可以有众多的子语句，指明了日志记录的位置、日志的内容、日志文件的大小和日志的级别等内容。下面是一个典型的日志语句内容。

```
logging{
  channel simple_log {  //定义一个名为 simple_log 的日志通道。可以定义多个通道，每
                        //个通道代表一种日志
  file "/var/log/named/bind.log" versions 3
                    //该日志记录在/var/log/named/bind.log 文件中，版本号为 3

  size 5m;       //文件的大小是 5MB，超过 5MB 时，会以 bind.log.1 的名字备份起来
    severity warning;    //高于或等于 warning 级别的日志才被记录
    print-time yes;      //日志记录包含时间域
    print-severity yes; //日志记录包含日志级别域
    print-category yes; //日志记录包含日志分类域
  };
  category default{      //所有的分类都记录到 simple_log 日志通道中
    simple_log;
  };
};
```

6．options语句

options 语句设定可以被整个 BIND 使用的全局选项。这个语句在每个配置文件中只有一处，如果出现多个 options 语句，则第一个 options 的配置有效，并且会产生一个警告信息。如果没有 options 语句，每个子语句使用默认值。options 选项的子语句很多，下面先解释一下在 14.2.3 小节的例子主配置文件中出现的子语句。

- directory：指定服务器的工作目录。配置文件其他语句中所使用的相对路径，指的都是在这个子语句指定的目录下。大多数的输出文件默认时也生成在这个目录下。如果没有设定，工作目录默认设置为服务器启动时的目录。指定目录时，应该以绝对路径表示。
- pid-file：设定进程 PID 文件的路径名，如果没有指定，默认为/var/run/named.pid。因此，此时要注意运行进程的用户 named 对该目录要有写入的权限，否则，named 将不能正常启动。pid-file 是给那些需要向运行着的服务器发送信号的程序使

用的。

- forwarders：设定转发使用的 IP 地址。该子语句只有在 forward 设置成允许转发后才生效，默认的列表是空的，表示不转发。转发也可以设置在每个域中，这样全局选项中的转发设置就不会起作用了。用户可以将不同的域转发到不同的其他 DNS 服务器上，或者对不同的域实现 forward only 或 first 的不同方式，也可以选择根本就不转发。
- allow-query：主语句用于设定 DNS 服务器为哪些客户机提供 DNS 查询服务，可以在后面的花括号内放置命名的 ACL 或 address_match_list，any 表示任何主机都可以访问。allow-query 也能在 zone 语句中设定，这样全局 options 中的 allow-query 选项在 zone 中就不起作用了。默认时是允许所有主机进行查询。

7．server语句

server 主语句定义了与远程服务器交互的规则，例如，决定本地 DNS 服务器是作为主域名服务器还是辅域名服务器，以及与其他 DNS 服务器通信时采用的密钥等。语句可以出现在配置文件的顶层，也可以出现在视图语句的内部。如果一个视图语句包括了自己的 server 语句，则只有那些视图语句内的 server 语句才起作用，顶层的 server 语句将被忽略。如果一个视图语句内不包括 server 语句，则顶层 server 语句将被当做默认值。

8．trusted-keys语句

trusted-keys 语句定义 DNSSEC 安全根的 trusted-keys。DNSSEC 指由 RFC2535 定义的 DNS sercurity。当一个非授权域的公钥是已知的，但不能安全地从 DNS 服务器获取时，需要加入一个 trusted-keys。这种情况一般出现在 singed 域是一个非 signed 域的子域的时候，此时加了 trusted key 后被认为是安全的。trusted-keys 语句能包含多重输入口，由键的域名、标志、协议算法和 64 位键数据组成。

9．view语句

view 语句定义了视图功能。视图是 BIND 9 提供的强大的新功能，允许 DNS 服务器根据客户端的不同有区别地回答 DNS 查询，每个视图定义了一个被特定客户端子集见到的 DNS 名称空间。这个功能在一台主机上运行多个形式上独立的 DNS 服务器时特别有用。

10．zone语句

zone 语句定义了 DNS 服务器所管理的区，也就是哪一些域的域名是授权给该 DNS 服务器回答的。一共有 5 种类型的区，由其 type 子语句指定，具体名称和功能如下所示。

- Master（主域）：主域用来保存某个区域（如 www.wzvtc.cn）的数据信息。
- Slave（辅域）：也叫次级域，数据来自主域，起备份作用。
- Stub：Stub 区与辅域相似，但它只复制主域的 NS 记录，而不是整个区数据。它不是标准 DNS 的功能，只是 BIND 提供的功能。
- Forward（转发）：转发域中一般配置了 forward 和 forwarders 子句，用于把对该

域的查询请求转由其他 DNS 服务器处理。

- Hint：Hint 域定义了一套最新的根 DNS 服务器地址，如果没有定义，DNS 服务器会使用内建的根 DNS 服务器地址。

在 14.2.3 小节的例子 named.conf 配置文件中有以下语句，现解释如下：

```
zone "." IN {           //定义一个名为“.”的区，查询类为 IN
type hint;              //类型为 hint
file "named.root";      //区文件是 named.root
};
zone "0.0.127.in-addr.arpa" IN {   //定义一个名为 0.0.127.in-addr.arpa 的区，
                                   //查询类为 IN
 type master;               //类型为 master
 file "named.local";        //区文件是 named.local
 allow-update { none; };    //不允许任何客户端对数据进行更新
};
zone  "wzvtc.cn" IN {       //定义一个名为 wzvtc.cn 的区，查询类为 IN
   type master;             //类型为 master
   file "named.wzvtc.cn";   //区文件是 named.wzvtc.cn
   allow-update { none; }; //不允许任何客户端对数据进行更新
};
zone  "1.10.10.in-addr.arpa" IN {
                   //定义一个名为 1.10.10.in-addr.arpa 的区，查询类为 IN
      type master;                 //类型为 master
      file "named.1.10.10";        //区文件是 named.1.10.10
      allow-update  { none; };     //不允许任何客户端对数据进行更新
};
```

说明：在每一个 zone 语句中，都用 file 子语句定义一个区文件，这个文件里存放了域名与 IP 地址的对应关系。14.3.3 节将对区文件进行详细解释。

14.3.2　根服务器文件 named.root

在主配置文件/etc/named.conf 中，定义了一个根域，区文件是/var/named 目录下的 named.root 文件。它是一个非常重要的文件，包含了 Internet 根服务器的名字和 IP 地址。当 Bind 接到客户端的查询请求时，如果本地不能解释，也不能在 Cache 中找到相应的数据，就会通过根服务器进行逐级查询。

例如，当服务器收到 DNS 客户机的一个查询请求，要求查询一个不在本域的 www.example.com 域名时，如果 Cache 里没有相应的数据，DNS 服务器就会向 named.root 文件中列出的 Internet 根服务器请求，然后根服务器将查询交给负责域.com 的授权名称服务器，域.com 授权名称服务器再将请求交给负责域 example.com 的授权名称服务器进行查询，最后再把结果返回给客户机。

由于 Internet 根服务器的地址经常会发生变化，因此 named.root 也应该要随之更新。最新的根服务器列表可以从 ftp://ftp.rs.internic.net/domain/下载，文件名也是 named.root，它包含了国际互联网络信息中心（InterNIC）提供的最新数据。另外，也可以用 Bind 提供的

命令 dig 列出最新的根服务器，命令如下：

```
# dig

; <<>> DiG 9.3.3rc2 <<>>
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 46053
;; flags: qr rd ra; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 15

;; QUESTION SECTION:
;.                          IN     NS

;; ANSWER SECTION:
.                    459744  IN     NS     F.ROOT-SERVERS.NET.
.                    459744  IN     NS     M.ROOT-SERVERS.NET.
.                    459744  IN     NS     I.ROOT-SERVERS.NET.
.                    459744  IN     NS     E.ROOT-SERVERS.NET.
…
.                    459744  IN     NS     D.ROOT-SERVERS.NET.

;; ADDITIONAL SECTION:
M.ROOT-SERVERS.NET.     546144  IN     A      202.12.27.33
J.ROOT-SERVERS.NET.     546144  IN     A      192.58.128.30
C.ROOT-SERVERS.NET.     546144  IN     A      192.33.4.12
A.ROOT-SERVERS.NET.     546144  IN     A      198.41.0.4
…
I.ROOT-SERVERS.NET.     546144  IN     A      192.36.148.17

;; Query time: 2 msec
;; SERVER: 10.10.1.2#53(10.10.1.2)
;; WHEN: Tue Nov 18 16:48:23 2008
;; MSG SIZE  rcvd: 492
```

以上列出的就是 Internet 根服务器的 IP 地址，如果使用以下命令，可以把这些内容存到 named.root 文件中，这个文件就可以做为主配置文件中指定的根域的区文件。

```
dig > /etc/named/named.root
```

14.3.3　区域数据文件

一个区域内的所有数据，包括主机名和对应 IP 地址、刷新间隔和过期时间等，都必须要存放在 DNS 服务器内，而用来存放这些数据的文件就称为区域文件。DNS 服务器的区域数据文件一般存放在/var/named 目录下。一台 DNS 服务器内可以存放多个区域文件，同一个区域文件也可以存放在多台 DNS 服务器中。下面是 14.2.3 小节配置例子中提供的 wzvtc.cn 域的区域数据文件 named.wzvtc.cn 的内容。

```
# vi  /var/named/named.wzvtc.cn
$TTL 3h
wzvtc.cn. IN SOA ns.wzvtc.cn. ltf@wzvtc.cn. (
                      1         ; 定义序列号的值，同步辅助名称服务器数据时使用
                      3h        ; 更新时间间隔值。定义该服务器的辅助名称服务器隔
                                  多久时间更新一次
```

```
                        1h        ; 辅助名称服务器更新失败时，重试的间隔时间
                        1w        ; 辅助名称服务器一直不能更新时，其数据过期的时间
                        1h )      ; 最小默认 TTL 的值，如果第一行没有$TTL，则使用
                                    该值
wzvtc.cn.       IN  NS   ns.wzvtc.cn.
wzvtc.cn.       IN  MX   10  mail
ns.wzvtc.cn.     IN   A    10.10.1.29
mail             IN   A    10.10.1.6
www              IN   A    10.10.1.3
oa               IN   CNAME  www
lib              IN   A     10.10.1.8
gsx              IN   A     221.224.2.234
```

在区域数据文件中，使用“;”作为行注释符，除第一条语句以外，区域数据文件中的每一条语句称为一条记录。以上配置中各条语句的含义如下所示。

1．设置其他DNS服务器缓存本机数据的默认时间

$TTL 指令要求放在文件的第 1 行，定义了其他 DNS 服务器缓存本机数据的默认时间，默认单位是秒，也可以用 h（小时）、d（天）和 w（星期）为单位。DNS 服务器在应答中提供 TTL 值，目的是允许其他的服务器在 TTL 间隔内缓存数据。如果本地的 DNS 服务器数据改变不大，可以考虑几天的默认 TTL，最长可以设为一周。但是不推荐设置 TTL 为 0，此时将导致大量的 DNS 数据传输。

2．设置起始授权机构

SOA 是 Start of Authority（起始授权机构）的缩写，它指出这个域名服务器是作为该区数据的权威的来源。在指令“wzvtc.cn. IN SOA ns.wzvtc.cn. ltf@wzvtc.cn.”中，指定了负责解析 wzvtc.cn.域的授权主机名是“ns.wzvtc.cn.”，授权主机名称将在区域文件中解析为 IP 地址。IN 表示属于 Internet 类，是固定不变的，“ltf@wzvtc.cn.”表示负责该区域的管理员的 E-mail 地址。每一个区文件都需要一个 SOA 记录，而且只能有一个。SOA 资源记录还要指定一些附加参数，放在 SOA 资源记录后面的括号内，其名称和功能见例子中的注释。

3．设置名称服务器NS资源记录

“wzvtc.cn. IN NS ns.wzvtc.cn.”是一条 NS（Name Server）资源记录，定义了域“wzvtc.cn.”由 DNS 服务器“ns.wzvtc.cn.”负责解析，NS 资源记录定义的服务器称为区域权威名称服务器。权威名称服务器负责维护和管理所管辖区域中的数据，被其他服务器或客户端当作权威的来源，并且能肯定应答区域内所含名称的查询。这里的配置要求和 SOA 记录配置一致。

4．设置邮件服务器MX资源记录

“wzvtc.cn. IN MX 10 mail”是一条 MX（Mail eXchanger）资源记录，表示发往 wzvtc.cn 域的电子邮件由 mail.wzvtc.cn 邮件服务器负责处理。例如，当一个邮件要发送地址到 test@wzvtc.cn 时，发送方的邮件服务器通过 DNS 服务器查询 wzvtc.cn 这个域名的 MX 资

源记录，查到后，会把邮件发送到指定的邮件服务器，如 mail.wzvtc.cn。至于该域名对应的 IP 地址，需要通过随后的 A 资源记录设定。

说明：可以设置多个 MX 资源记录，指明多个邮件服务器，优先级别由 MX 后的数字决定，数字越小，邮件服务器的优先权越高。优先级高的邮件服务器是邮件传送的主要对象，当邮件传送给优先级高的邮件服务器失败时，可以把它传送给优先级低的邮件服务器。

5．设置主机地址A资源记录

主机地址 A（Address）资源记录是最常用的记录，它定义了 DNS 域名对应 IP 地址的信息。在上面的例子中，使用了两种方式来定义 A 资源记录，一种是使用相对名称，即在名称的末尾没有加“.”，另外一种是使用完全规范域名 FQDN（Fully Qualified Domain Name），即名称的最后以“.”结束。这两种方式只是书写形式不同而已，在使用上没有任何区别。例如，对于相对名称 mail、oa 等，Bind 会自动在相对名称的后面加上后缀“.wzvtc.cn.”，所以相当于完全规范域名的 mail.wzvtc.cn.和 oa.wzvtc.cn.。

6．设置别名CNAME资源记录

别名 CNAME（Canonical Name）资源记录也被称为规范名字资源记录。CNAME 资源记录允许将多个名称映射到同一台计算机上，使得某些任务更容易执行。例如，对于同时提供 Web、OA 服务的计算机（IP 地址为 10.10.1.3），为了便于用户访问服务，可以先为其建立一条主机地址 A 资源记录“www　IN　A　10.10.1.3”，将 www.wzvtc.cn 映射到 10.10.1.3 地址，然后再为该计算机设置 oa 别名，即建立 CNAME 资源记录“oa　IN　CNAME　www”。这样，当访问 www.wzvtc.cn 和 oa.wzvtc.cn 时，实际都是访问 IP 地址为 10.10.1.3 的计算机。

14.3.4　反向解析区域数据文件

反向解析区域数据文件的结构和格式与区域数据文件类似，只不过它的主要内容是建立 IP 地址映射到 DNS 域名的指针 PTR 资源记录。下面是 14.2.3 配置例子中提供的 named.1.10.10 域的反向解析区域数据文件 named.1.10.10 的内容。

```
# vi /var/named/named.1.10.10
$TTL 3h
1.10.10.in-addr.arpa. IN SOA ns.wzvtc.cn. ltf@wzvtc.cn.(
                          1          ; Serial
                          3h         ; Refresh after 3 hours
                          1h         ; Retry after 1 hour
                          1w         ; Expire after 1 week
                          1h )       ; Negative caching TTL of 1 hour
1.10.10.in-addr.arpa.     IN  NS     ns.wzvtc.cn.
29.1.10.10.in-addr.arpa.  IN  PTR    ns.wzvtc.cn.
6.1.10.10.in-addr.arpa.   IN  PTR    mail.wzvtc.cn.
8.1.10.10. in-addr.arpa.  IN  PTR    lib.wzvtc.cn.
```

反向域名解析是通过 in-addr.arpa 域和 PTR 记录实现的。in-addr.arpa 域入口可以设成最不重要到最重要顺序，从左至右阅读，这与 IP 地址的通常顺序相反。于是，一台 IP 地址为 10.1.2.3 的机器将会有对应的 in-addr.arpa 名称：3.2.1.10.in-addr.arpa。这个名称应该具有一个 PTR 资源记录，它的数据字段是主机名称。下面看一下以上配置的具体解释。

1．设置SOA和NS资源记录

反向解析区域文件必须包括 SOA 和 NS 资源记录，使用固定格式的反向解析区域 in-addr.arpa 作为域名，结构和格式与区域数据文件类似，这里不再重复。

2．设置指针PTR资源记录

指针 PTR 资源记录只能在反向解析区域文件中出现。PTR 资源记录和 A 资源记录正好相反，它是将 IP 地址解析成 DNS 域名的资源记录。与区域文件的其他资源记录类似，它也可以使用相对名称和完全规范域名 FQDN。例如，“6.1.10.10.in-addr.arpa. IN PTR mail.wzvtc.cn.”表示 IP 地址 10.10.1.6 对应的域名为 mail.wzvtc.cn。

14.3.5　配置 DNS 负载均衡功能

随着网络的规模越来越大，用户数急剧增加，网络服务器的负担也变得越来越重，一台服务器要同时应付成千上万用户的并发访问，必然会导致服务器过度繁忙，响应时间过长的结果。DNS 负载均衡的优点是简单易行，而且实现代价小。它在 DNS 服务器中为同一个域名配置多个 IP 地址（即为一个主机名设置多条 A 资源记录），在应答 DNS 查询时，DNS 服务器对每个查询将以 DNS 文件中主机记录的 IP 地址按顺序返回不同的解析结果，将客户端的访问引导到不同的计算机上去，从而达到负载均衡的目的。下面是一个实现邮件服务器负载平衡的配置片段（在区域数据文件中）。

```
     IN  MX  10  mail.example.com.
     IN  MX  10  mail1.example.com.
     IN  MX  10  mail2.example.com.
...
mail  IN  A      192.168.0.4
mail1 IN  A      192.168.0.5
mail2 IN  A      192.168.0.6
```

在以上配置中，mail、mail1 和 mail2 均是 example.com.域中的邮件服务器，而且优先级都是 10。当客户端（通常是 SMTP 软件）查询邮件服务器 IP 地址时，Bind 将根据 rrset-order 语句定义的次序把配置中设定的 3 条 A 记录都发送给客户端，客户端可以使用自己规定的算法从 3 条记录中挑选一条。rrset-order 语句是主配置文件中 options 主语句的一条子语句，可以定义固定、随机和轮询的次序。下面的配置是另一种实现邮件服务器负载平衡的方法。

```
       IN  MX  10  mail.example.com.
...
mail        IN  A      192.168.0.4
            IN  A      192.168.0.5
```

```
            IN  A      192.168.0.6
```

在以上配置中，mail.example.com 对应了 3 个 IP 地址，此时，具体选择哪一条 A 记录，也是由 rrset-order 语句决定。另外，在反向解析文件中，这 3 个 IP 都要对应 mail 主机，以免有些邮件服务器为了反垃圾邮件进行反向查询时出现问题。

除了邮件服务器以下，其他的服务也可以采用类似的配置实现负载均衡。例如，要使用 3 台内容相同的 FTP 服务器共同承担客户机的访问，它们的 IP 地址分别是 10.10.1.7、10.10.1.10 和 10.10.1.13。可以根据 14.2.3 小节所提供的一套配置文件，在 named.wzvtc.cn 区域数据文件中输入以下内容来达到目的。

```
ftp    IN   A   10.10.1.7
ftp    IN   A   10.10.1.10
ftp    IN   A   10.10.1.13
```

此时，为了解析客户端对 ftp.wzvtc.cn 的域名查询，DNS 服务器会轮询这 3 条 A 资源记录，以 rrset-order 子语句设定的顺序响应用户的解析请求，实现了将客户机的访问分担到每个 FTP 服务器上的负载均衡功能。测试结果如图 14-11 所示，可以看到，3 次查询 ftp.wzvtc.cn 域名，得到的 IP 地址次序是不一样的。

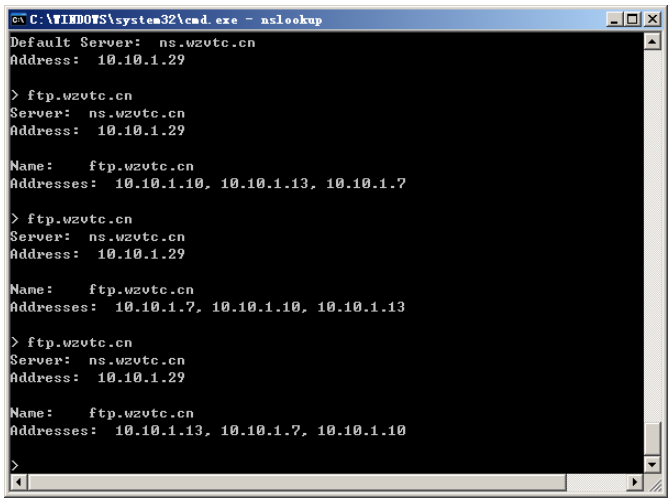

图 14-11　负载均衡测试结果

注意：以上的 Bind 配置只是为其他服务器的负载平衡提供了条件。当具体使用时，还需要多台服务器之间采取同步等措施才能真正实现。

14.3.6　直接域名、泛域名与子域

许多用户有直接使用域名访问 Web 网站的习惯，即在浏览器中不输入 www 等主机名，而是直接使用如 http://baidu.com/或 http://tom.com/等域名来访问。然而，并不是所有的 Web 网站都支持这种访问方式，只有 DNS 服务器能解析直接域名的网站才可以使用。可以在

named.wzvtc.cn 区域文件中加入以下内容实现直接域名解析。

```
wzvtc.cn.  IN    A   10.10.1.3
```

此时，域名 wzcvtc.cn 可以解析为 10.10.1.3，与 www.wzvtc.cn 域名的解析结果一样，测试情况如图 14-12 所示。

另外，如果在 named.wzvtc.cn 中加入以下语句，还可以实现一种泛域名的效果。

```
*.wzvtc.cn.   IN     A      10.10.1.3
```

泛域名是指一个域名下的所有主机和子域名都被解析到同一个 IP 地址上。在以上配置中，所有以“.wzvtc.cn”为后缀的域名的 IP 地址都将解析为 10.10.1.3。另外，默认情况下泛域名解析的优先级最高，如果区域文件中存在其他主机的 A 资源记录，它们都将失效。图 14-13 所示的是泛域名的测试结果。

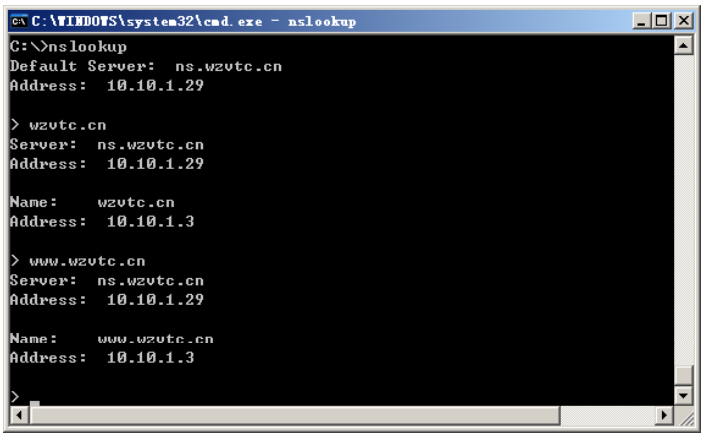

图 14-12　直接域名解析测试结果

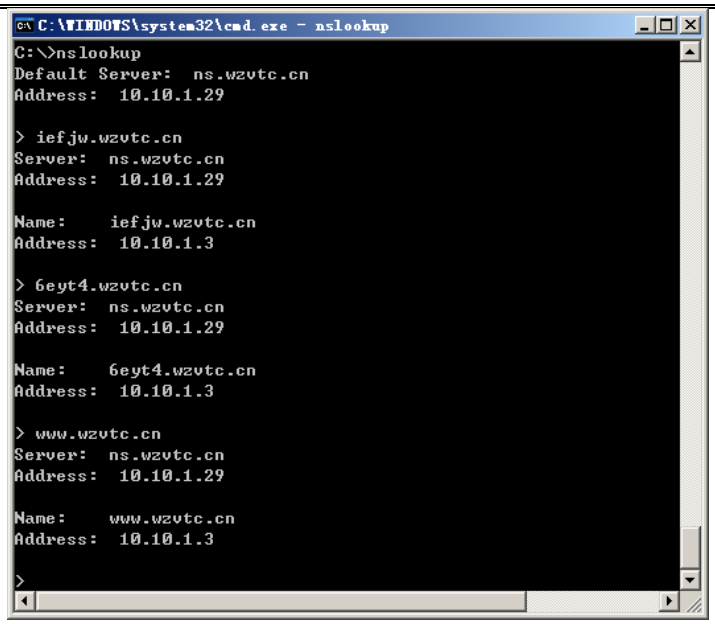

图 14-13　泛域名解析测试结果

从图 14-13 中可以看到，不管采用什么样的主机名，只要后缀是“.wzvtc.cn”，IP 地址都将解析为 10.10.1.3。

子域（Subdomain），是域名层次结构中的一个术语，是对某一个域进行细分时的下一级域。例如，wzvtc.cn 是一个顶级域名，可以把 dean.wzvtc.cn 配置成是它的一个子域。配置子域可以有两种方式，一种是把子域配置放在另一台 DNS 服务器上，还有一种是子域配置与父域配置放在一起，此时也称为虚拟子域。下面介绍一下虚拟子域的配置方法。

假设在 14.2.3 所提供的一套配置文件的基础上，要求配置一个虚拟子域，名为 dean.wzvtc.cn。此时，需要在区域文件 named.wzvtc.cn 添加以下内容。

```
$ORIGIN dean.wzvtc.cn.
mail        IN     A     10.10.3.28
ftp         IN     A     10.10.3.29
```

mail 和 ftp 是定义在子域 dean.wzvtc.cn 中的主机名，即域名 mail.dean.wzvtc.cn 和 ftp.dean.wzvtc.cn 对应的 IP 地址分别是 10.10.3.28 和 10.10.3.29。测试结果如图 14-14 所示。

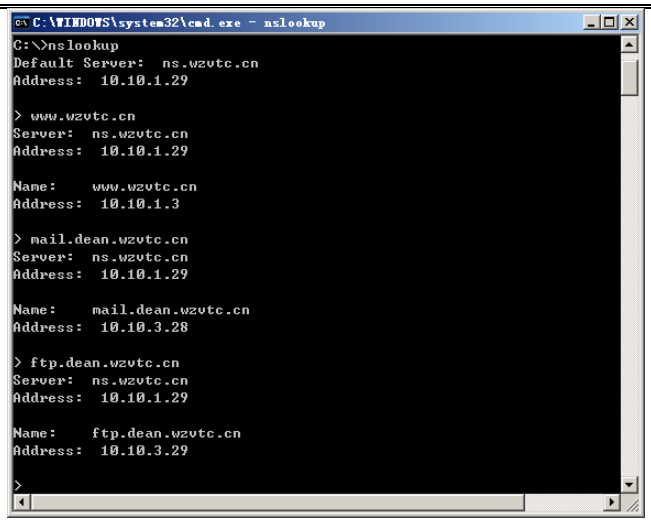

图 14-14　虚拟子域测试结果

当然，在子域 dean.wzvtc.cn 中也可以配置邮件网关等功能，其配置与父域中的配置类似。

14.3.7　辅域服务器和只缓存服务器

在 Bind 服务器中，还可以根据需要配置辅域服务器和只缓存服务器，以便能更快地能为客户端提供 DNS 服务，并提高可靠性。

1．辅域服务器

辅域服务器也可以向客户机提供域名解析功能，但它与主域服务器不同的是，它的数据不是直接输入的，而是从其他 DNS 服务器（主域服务器或其他的辅域服务器）中复制过来的，只是一份副本，所以辅域服务器中的数据无法被修改。

当启动辅域服务器时，它会和指定的所有主域服务器建立联系，并从中复制数据。在辅域服务器工作时，还会定期地更改原有的数据，以尽可能保证副本与正本数据的一致性。在大型网络中，经常设置多台辅域服务器，主要目的如下所示。

- 提供容错能力：当主域服务器发生故障时，由辅域服务器提供服务。
- 分担主域服务器的负担：在 DNS 客户端较多的情况下，通过架设辅域服务器完成对客户端的查询服务，可以有效地减轻主域服务器的负担。
- 加快查询的速度：如果本地网络必须使用某一台 DNS 服务，但与这台 DNS 服务器连接的速度较慢，可以在本地网络配置一台远程 DNS 服务器的辅服务器，使本地 DNS 客户端直接向此辅域服务器进行查询，而不需要向速度较慢的主域服务器查询，以加快速度，并减少用于 DNS 查询的外网通信量。

辅域服务器的主配置文件也是/etc/named.conf，也需要设置服务器的 options 主语句和根区域，方法与配置主域服务器的方法相同。但在配置区域时，只需要提供区域名和主域

服务器的 IP 地址，而不需要建立相应的区域文件。因为一个辅域服务器不需要在本地建立各种资源记录，而是通过一个区域复制过程来得到主域服务器上的资源记录。下面是辅域服务器主配置文件/etc/named.conf 的部分内容。

```
...    ;其余配置未变化
zone "wzvtc.cn" {
     type slave;              ;区类型为 slave
     file "slaves/wzvtc.cn.zone";
     masters {10.10.1.2;};       ;要联系的主服务器为 10.10.1.2
};
zone "1.10.10.in-addr.arpa" {
     type slave;    ;区类型为 slave
     file "slaves/1.10.10.arpa";
     masters {10.10.1.2;};       ;要联系的主服务器为 10.10.1.2
};
...；其余配置未变化
```

与前面主域服务器的配置相比，以上配置中的 wzvtc.cn 和 1.10.10.in-addr.arpa 两个区的 type 设成 slave，区域数据文件的位置也发生了变化。还有，在两个区域中均增加了“masters {10.10.1.2;}”语句，表示主域服务器的 IP 地址是 10.10.1.2。此时，要假设在 IP 地址为 10.10.1.2 的计算机上运行着 Bind，其使用的配置文件是 14.2.3 所提供的那一套，但有关 wzvtc.cn 和 1.10.10.in-addr.arpa 区域的配置改为以下内容。

```
...；其余配置未变化
zone  "wzvtc.cn" IN {
   type master;
   file "named.wzvtc.cn";
   allow-update { none; };
   allow-transfer {10.10.1.29;};   ; 允许向 10.10.1.29 传送区域数据
};

zone  "1.10.10.in-addr.arpa" IN {
       type master;
       file "named.1.10.10";
       allow-update  { none; };
   allow-transfer {10.10.1.29;};   ; 允许向 10.10.1.29 传送区域数据
};
...；其余配置未变化
```

与原来相比，两个区的配置中均多了“allow-transfer {10.10.1.29;};”子语句，表示允许向 10.10.1.29（辅域服务器 IP 地址）的计算机传送区域数据。

2．只缓存服务器

只缓存服务器是一种很特殊的 DNS 服务器，它本身并不管理任何区域，但是 DNS 客户端仍然可以向它请求查询。只缓存服务器类似于代理服务器，它没有自己的域名数据库，而是将所有查询转发到其他 DNS 服务器处理。当只缓存服务器从其他 DNS 服务器收到查询结果后，除了返回给客户机外，还会将结果保存在缓存中。当下一个 DNS 客户端再查询相同的域名数据时，就可以从高速缓存里得到结果，从而加快对 DNS 客户端的响应速度。如果在局域网中建立一台这样的 DNS 服务器，就可以提高客户机 DNS 的查询效率并减少

内部网络与外部网络的流量。

架设只缓存服务器非常简单，只需要建立主配置文件 named.conf 即可。一个典型的只缓存服务器配置如下：

```
options {
  directory "/var/named";
  version "not currently available";        ;隐藏名称与版本号
 forwarders { 202.96.0.133;61.144.56.101;};转发到其他 DNS 服务器进行查询
  forward only;                             ;只转发，自己不提供解析服务
  allow-transfer{"none";};
  allow-query {any;};
};
logging{                                    ;定义了一个日志
  channel example_log{
   file "/var/log/named/example.log" versions 3;
   severity info;
   print-severity yes;
   print-time yes;
   print-category yes;
 };
 category default{
  example_log;
 };
};
```

其中关键的语句是“forward only;”，表示只对客户端提交的查询进行转发，由其他 DNS 服务器提供查询结果，自己只对结果进行缓存，以便下次碰到同样的查询时能更快地响应。至于转发到哪一台 DNS 服务器，由“forwarders { 202.96.0.133;61.144.56.101;}”语句决定。

14.4　小　　结

DNS 是 Internet 上必不可少的一种网络服务，它提供把域名解析为 IP 地址的服务，是每一台上网的计算机都必须要使用的服务之一。本章首先介绍了 DNS 的工作原理、DNS 协议，然后介绍了用 Bind 软件架设 DNS 服务器的方法，包括 Bind 的安装、运行和配置，以及 chroot、负载均衡、泛域名、辅域服务器、只缓存服务器等特殊功能的配置方法。