第 1 章 DNS 解析服务器企业实战

1.1 DNS 服务器工作原理

域名系统（Domain Name System，DNS）是互联网的一项服务。它作为将域名和 IP 地址相互映射的一个分布式数据库，能够使人更方便地访问互联网。DNS 使用 TCP 和 UDP 端口 53。当前，对于每一级域名长度的限制是 63 个字符，域名总长度则不能超过 253 个字符。

早期域名的字符仅限于 ASCII 字符的一个子集。2008 年，ICANN（互联网名称与数字地址分配机构）通过一项决议，允许使用其他语言作为互联网顶级域名的字符。使用基于 Punycode（域名代码）的 IDNA 系统，可以将 Unicode（统一码）字符串映射为有效的 DNS 字符集。因此，诸如“XXX.中国”“XXX.美国”的域名可以在地址栏直接输入并访问，而不需要安装插件。但是，由于英语的广泛使用，使用其他语言字符作为域名会产生多种问题，例如难以输入、难以在国际推广等。

我们每天打开的网站是如何解析的？我们怎么能得到网站的内容反馈呢？这些都是通过 DNS 服务器实现的。

下面介绍 DNS 服务器的构建。DNS 服务可以算是 Linux 服务中比较难的一个了，尤其是配置文件书写，少一个字符都有可能造成错误。

简单地说，DNS 的功能就是完成域名到 IP 地址的解析过程。简洁的域名更方便人们记忆，不需要记那么长的 IP 地址去访问某个网站。

1.2 DNS 解析过程

DNS 解析过程可以分为以下几步。

（1）客户机访问某个网站，请求域名解析时，DNS 首先查找本地 HOSTS 文件，如果有对应域名、IP 地址记录，则直接返回给客户机；如果没有则将该请求发送给本地的域名服务器。

（2）若本地 DNS 服务器能够解析客户端发来的请求，则直接将答案返回给客户机。

（3）本地 DNS 服务器不能解析客户端发来的请求时，有两种解析方法。

① 采用递归解析。

本地 DNS 服务器向根域名服务器发出请求，根域名服务器对本地域名服务的请求进行解析，得到记录再发给本地 DNS 服务器，本地 DNS 服务器将记录缓存，并将记录返回给客户机。

② 采用迭代解析。

本地 DNS 服务器向根域名服务器发出请求，根域名服务器返回本地域名服务器一个能够解析请求的根的下一级域名服务器的地址，本地域名服务器再向根域名服务器返回的 IP 地址发出请求，最终得到域名解析记录。

1.3 DNS 服务器种类

DNS 服务器主要有以下几种。

（1）Master（主 DNS 服务器）：拥有区域数据的文件，并对整个区域数据进行管理。

（2）Slave（从服务器或辅助服务器）：拥有主 DNS 服务器的区域文件的副本，辅助 DNS 服务器对客户端进行解析，当主 DNS 服务器崩溃后，可以完全接替主服务器的工作。

（3）Forward：将任何查询请求都转发给其他服务器，起代理的作用。

（4）Cache：缓存服务器。

（5）Hint：根 DNS Internet 服务器集。

1.4 DNS 服务器安装配置

（1）安装 Bind DNS 软件包：

```
yum  install  bind*  -y
```

（2）配置文件/etc/named.conf 内容：

```
options {
       listen-on port 53 { any; };
       listen-on-v6 port 53 { any; };
       directory      "/var/named";
       dump-file      "/var/named/data/cache_dump.db";
       statistics-file    "/var/named/data/named_stats.txt";
       memstatistics-file   "/var/named/data/named_mem_stats.txt";
       allow-query    { any; };
       recursion yes;
       dnssec-enable yes;
       dnssec-validation yes;
       dnssec-lookaside auto;
       /* Path to ISC DLV key */
       bindkeys-file "/etc/named.iscdlv.key";
       managed-keys-directory "/var/named/dynamic";
};
logging {
       channel default_debug {
              file "data/named.run";
              severity dynamic;
       };
};

zone "." IN {
       type hint;
       file "named.ca";
};
include "/etc/named.rfc1912.zones";
include "/etc/named.root.key";
```

1.5　DNS 主配置文件详解

DNS 主配置文件 named.conf 详解如下：

```
options {
directory "/var/named";                                  #指定配置文件所在目录,必须配置此项
dump-file  "/var/named/data/cache_dump.db;;#保存 DNS 服务器搜索到的对应 IP 地址
                                                         #的高速缓存
```

```
statistics_file  "/var/named/data/named_stats.txt; #DNS 的一些统计数据列出时
                                                   #就写入这个设置指定的文件,
                                                   #即搜集统计数据
pid-file "/var/run/named/named.pid;     #用于记录 named 程序的 PID 文件,可在
                                        #NAMED 启动、关闭时提供正确的 PID
allow_query (any;);     #是否允许查询,或允许哪些客户端查询。可以把 any 换上网段地址,
                        #以设置允许查询的客户端
allow_transfer(none;); #是否允许主服务器里的信息传到从服务器,只有在同时拥有
                        #主服务器和从服务器时才设置此项。none 为不允许
forwarders{192.168.3.11;192.168.3.44;};
                        #设置向上查找的"合法"的 DNS。地址之间要用";"分隔。(笔者
                        #的理解是此处定义的如同 Windows 里定义的转发一样,当本地 DNS
                        #服务器解析不了时,转发到用户指定的一个 DNS 服务器上解析)
#当不配置此项时,本机无法解析的都会在 name.ca 中配置的根服务器上查询;但如果配置了此项,
#本机查找不到的,就丢给此项中配置的 DNS 服务器处理
forward only            #让 DNS 服务器只作为转发服务器,自身不进行查询
motify                  #当主服务器变更时,向从服务器发送信息。有两个选项,yes 和 no
};
```

配置/etc/named.rfc1912.zones 文件（用于定义根区域和自定义区域），添加如下代码：

```
#add named by www.jfedu.net
zone "jfedu.net" IN {
       type master;
       file "jfedu.net.zone";
       allow-update { none; };
};
zone "1.168.192.in-addr.arpa" IN {
       type master;
       file "jfedu.net.arpa";
       allow-update { none; };
};
```

1.6 DNS 自定义区域详解

DNS 自定义解析区域文件，详解如下：

```
#定义正向解析文件,此处以 jfedu.net 域为例
  zone "jfedu.net" IN {
  type master;                          #定义服务器类型
  file "jfedu.net";                     #指定正向解析文件名
```

```
};
#定义反向解析文件
zone "1.168.192.in-addr.arpa" {
  type master;                          #服务器类型
  file "named.192.168.9" ;              #反向解析文件名
};
```

/var/named/目录创建如下两个文件，其中 jfedu.net.zone 正向解析文件内容如下：

```
$TTL 86400
@   IN SOA  ns.jfedu.net.  root (
                 42  ; serial
                 3H  ; refresh
                 15M ; retry
                 1W  ; expire
                 1D )    ; minimum
@               IN NS           ns.jfedu.net.
ns              IN A            192.168.0.111
www             IN A            192.168.0.111
@               IN MX 10        mail.jfedu.net.
mail            IN A            192.168.0.111
```

在/var/named/目录创建如下两个文件，其中 jfedu.net.zone 反向解析文件内容如下：

```
$TTL 86400
@   IN SOA  ns.jfedu.net. root (
                 42  ; serial
                 3H  ; refresh
                 15M ; retry
                 1W  ; expires
                 1D )    ; minimum

@   IN NS      ns.jfedu.net.
111 IN PTR     mail.jfedu.net.
111 IN PTR     ns.jfedu.net.
111 IN PTR     www.jfedu.net.
```

1.7　DNS 正反向文件详解

DNS 正反向文件详解如下：

```
$TTL  86400          #外 DNS 服务器请求本 DNS 服务器的查询结果,在外 DNS 服务器上的缓
                     #存时间,以 s 为单位
```

```
@  IN SOA ns.jfedu.net. root. (          #格式为
#【主机名或域名】ttl] [calss] [type] [orgin] [mail]
#主机名或域名一般用@代替。每个区域都有自己的 SOA 记录,此处为指定的域名用@表示当前的源,
#也可以手动指定域名
#SOA 记录(起始授权机构)NS(Name  Server)记录(域名服务器)
#ttl: 通常省略
#class: 类别,说明网络类型
#type: 类型,SOA 记录的类型就是 SOA,指明哪个 DNS 服务器对这个区域有授权
#origin: 区域文件资源,这个区域文件资源就是这个域主 DNS 服务器的主机名,注意这里要求是完
#整的主机名,后面一定要加上"."。上例中,"www.jfedu.net."如果没有加后面的点,结果将是
#www.jfedu.net.jfedu.net
#mail: 一般指管理员的邮箱。但和一般的邮箱不同,此处用"."代替了"@",尾部也要加上"."
2009121001  #作为版本控制,当域区文件修改时,序号就增加,辅助服务器对比发现与自己的不
            #同后,就会做出更新,与主服务器同步
28800    #辅助服务器与主服务器进行更新的等待时间。间隔多久与主服务器进行更新,单位为 s
14400    #重试间隔。当辅助服务器请求与主服务器更新失败后,再间隔多久重试传递
720000   #到期时间。当辅助服务器与主服务器之间刷新失败后,辅助服务器还提供多久的授权回
         #答。因为当与主服务器失去联系一定时间(即此处定义的时间)后,辅助服务器会把本
         #地数据当作不可靠的数据,将停止提供查询
#如果主服务器恢复正常,则辅助服务器重新开始计时
86400 ) #最小 TTL,即最小有效时间,表明客户端得到的回答在多长时间内有效。如果 TTL 时间
         #长,则客户端缓存保存时间长,客户端在收到查询结果时开始计时,TTL 时间内有相同
         #的查询周期不再查询服务器,而是直接查自己的缓存;如果 TTL 时间短,则缓存更新
         #的频率快
@          IN   NS   www.jfedu.net.        #ns 记录
www   IN       A     192.168.1.13          #A 记录
ftp         IN   CName   www.jfedu.net.    #别名类型
mail   IN     MX  10   192.168.1.12        #邮件交换器
#IN                                        #表示后面的数据使用的是 Internet 标准
#SOA                                       #表示授权开始
#@                                         #代表相应的域名
test.com                                   #授权主机
root.test.com                              #管理者信箱
#NS: 表示这个主机是一个域名服务器
#A: 定义了一条 A 记录,即主机名到 IP 地址的对应记录
#MX 定义了一条邮件记录
#CNAME                                     #定义了对应主机的一个别名
#type 类型有三种,分别是 master、slave 和 hint,它们的含义分别如下
#master: 表示定义的是主域名服务器
#slave: 表示定义的是辅助域名服务器
#hint: 表示定义的是互联网中的根域名服务器
```

```
#zone 定义域区,一个 zone 关键字定义一个区域
#PTR 记录用来解析 IP 地址对应的域名
#注释二
#Serial：其格式通常为"年月日+修改次序"
#当辅助服务器要进行资料同步的时候,会比较这个字符串。如果发现在这里的字符串比其记录
#的字符串"大",就进行更新,否则忽略。注意： Serial 不能超过 10 位数字
#Refresh：告诉辅助服务器隔多久要进行资料同步(是否同步要看 Serial 的比较结果)
#Retry：如果辅助服务器更新失败,要隔多久再进行重试
#Expire：记录逾期时间：当辅助服务器一直未能成功与主服务器取得联系,到这里就放弃重试,
#同时这里的资料也将标识为过期(expired)
#Minimum：最小默认 TTL 值,如果在前面没有用"$TTL"定义,就会以此值为准
```