11 DNS域名解析服务

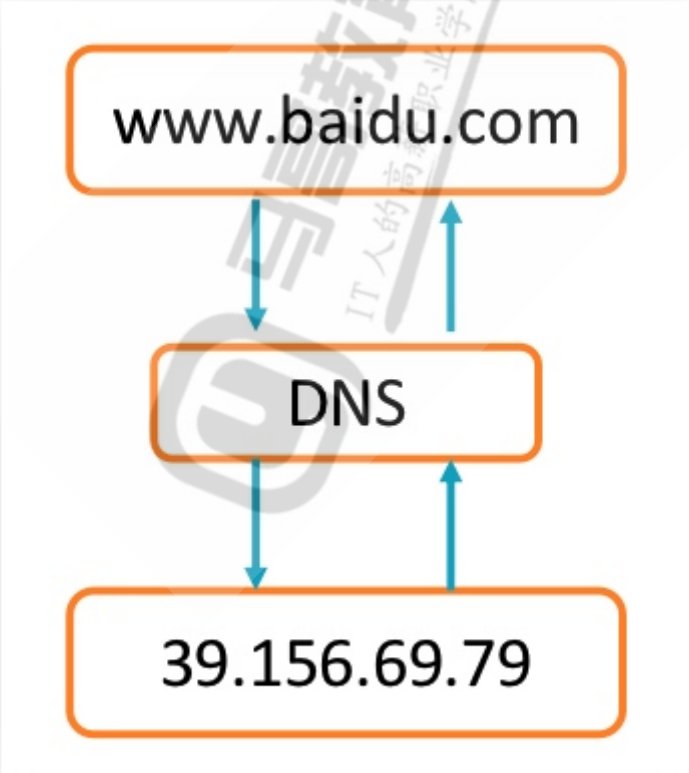
1.DNS基本概述

1.1 为什么需要域名

- 在互联网中，使用 `IP` 地址与服务器进行通信根本行不通，原因如下：
  - 1.不好记忆，例如：学校官网的`IP`地址是 `"39.104.16.126"`，难以记忆；
  - 2.`IP`地址会经常变更，所以通过`IP`地址去访问某台机器就会发生问题；
- 此时 `DNS` 协议就应运而生了；那`DNS`解决了什么问题：
  - `DNS`主要用来管理域名（比较好记忆的字符）与`IP`地址（比较难记忆）的对应关系表；

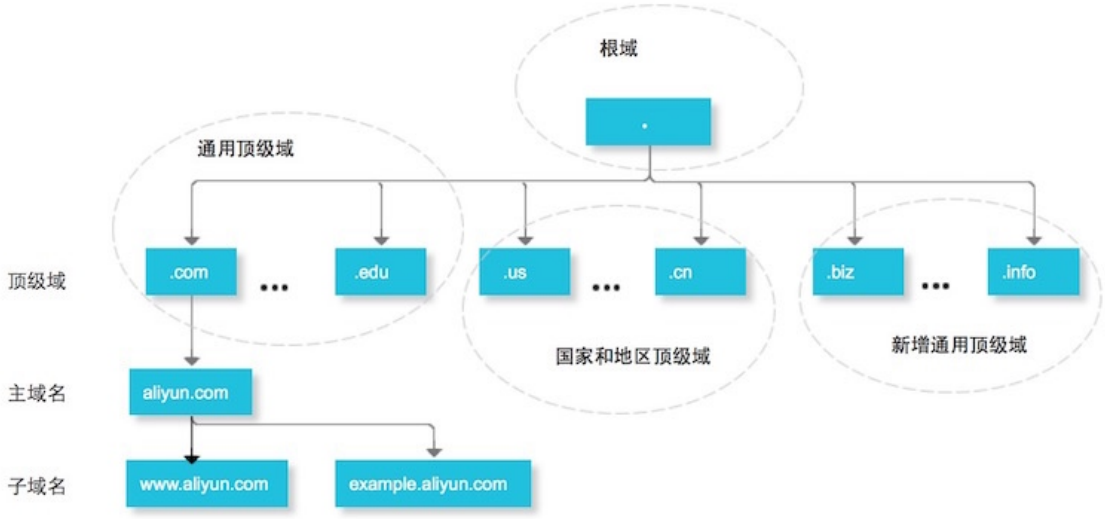
1.2 什么是DNS

`DNS（Domain Name System）` 是 ”域名系统“ 英文缩写，它所提供的服务是用来将域名转换为 `IP` 地址的工作。`DNS` 就是一位 ”翻译官“，它的基本工作原理可以用下图来表示；



1.3 DNS域名结构

- 由于因特网的用户数量较多，所以因特网在命名时采用的是层次树状结构的命名方法。
  - 1.互联网中的域名是有结构有规划的；
  - 2.由于域名进行了分级，在进行域名和IP地址解析时能更容易找到；
  - 3.其次域名具备全球唯一性；



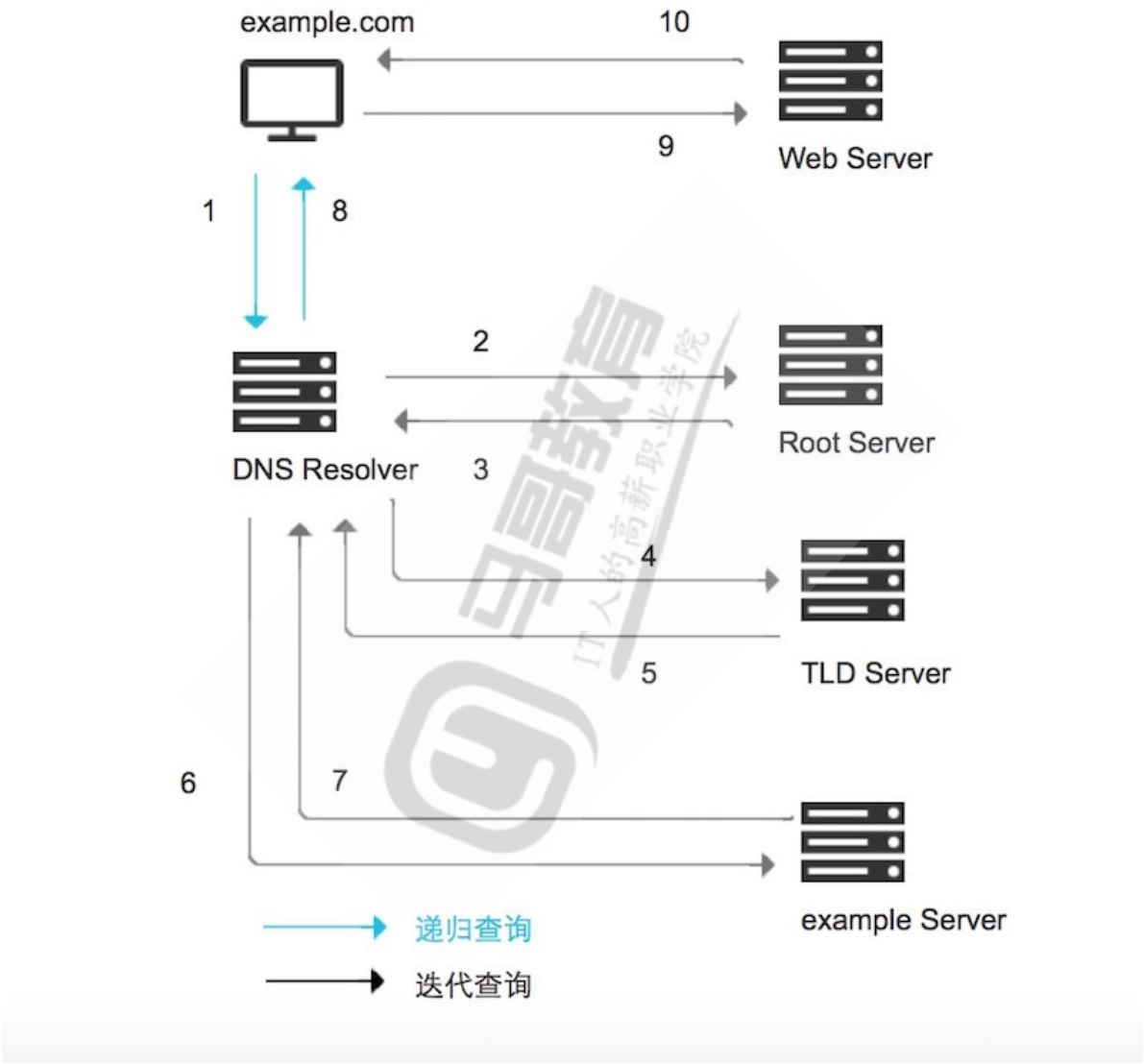
- 根域：知道所有顶级域名服务器的域名和地址
  - 全世界只有13组根服务器，其中10台设置在美国，另外的三台设置与英国，瑞典，日本；
- 顶级域：知道所有顶级域名服务器下注册的所有二级域名的IP地址
  - 顶级域有两种：通用域（com、cn）域和国家域（hk、jp）；
  - 顶级域名由 ICANN （互联网名称与数字地址分配机构）委任的注册机构负责运行；
- 二级域：负责一个区的域名服务器（oldxu.com）
  - 无需到 ICANN 进行申请，只需要到运行顶级域的注册机构（阿里、腾讯）去申请即可
  - 如果申请的域名没有被注册，也没有被注册为商标，仅需要缴纳一笔年费即可得到心仪的域名
- 三级域或主机名：
  - 根据服务器所提供的业务功能，选择配置对应的主机名称解析记录，比如（www、ops）

1.4 DNS查询原理

DNS 查询的结果通常会在本地域名服务器中进行缓存，如果本地域名服务器中有缓存的情况下，则会跳过如下 DNS 查询步骤，很快返回解析结果。
下面的示例则概述了本地域名服务器没有缓存的情况下，DNS 查询所需的步骤：

- 1、用户在浏览器中输入 example.com， 则由本地域名服务器开始进行递归查询。
- 2、本地域名服务器采用迭代查询的方法，向根域名服务器进行查询。
- 3、根域名服务器告诉本地域名服务器，下一步应该查询的顶级域名服务器 .TLD 的IP地址。

- 4、本地域名服务器向顶级域名服务器 `.com TLD` 进行查询。
- 5、.com TLD服务器告诉本地域名服务器，下一步查询 `example.com` 权威域名服务器的IP地址。
- 6、本地域名服务器向 `example.com` 权威域名服务器发送查询。
- 7、`example.com` 权威域名服务器告诉本地域名服务器所查询的主机IP地址。
- 8、本地域名服务器最后把查询的 `IP` 地址响应给浏览器。



https://s4.51cto.com/images/blog/202005/06/aa0989e8b7604faef3d796a9ecd242d6.png

1.4.1 递归查询

是指`DNS`服务器在收到用户发起的请求时，必须向用户返回一个准确的查询结果。如果`DNS`服务器本地没有存储与之对应的信息，则该服务器需要询问其他服务器，并将返回的查询结构提交给用户。

1.4.2 迭代查询

是指DNS服务器在收到用户发起的请求时，并不直接回复查询结果，而是告诉另一台DNS服务器的地址，用户再向这台DNS服务器提交请求，这样依次反复，直到返回查询结果。

1.4 DNS记录类型

1.4.1 A

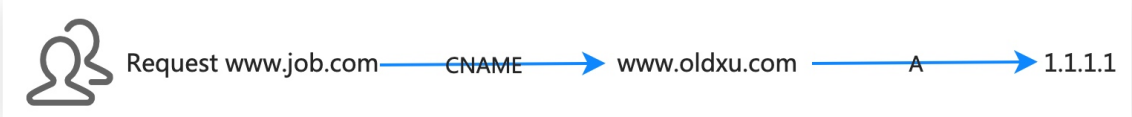
- A 记录可实现将域名指向 IP 地址，也称为正向解析；
- 正向解析：域名-->DNS服务返回 IP

1.4.2 PTR

- PTR记录可以实现IP查找域名，也称为反向解析；
- 反向解析：IP-->DNS服务返回域名；

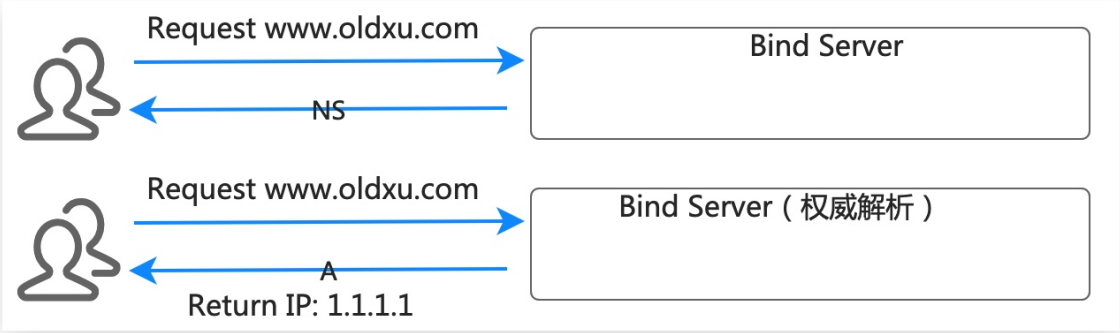
1.4.2 CNAME

- 当需要将域名指向另一个域名，再由另一个域名提供 IP 地址，就需要添加 CNAME 记录；
- 最常用 CNAME 的场景有 WAF、CDN



1.4.4 NS

- 1.客户端查询DNS服务，如当前DNS无法提供权威解析，则返回一条NS记录；
- 2.客户端在通过NS记录中提供的DNS权威服务器进行解析；



1.5 企业自建DNS

- 企业常规做法：购买域名、完成ICP备案，并使用公网DNS服务（万网..）进行免费（付费）解析
- 为什么需要：
  - 1.内网web服务，例如：jenkins、jumpserver、wik等，不适合解析至公网；
  - 2.内网中间件服务db、mq等，由于会经常迁移或扩缩容，应该使用域名对外提供，便于维护；
  - 3.服务器都有hostname，hostname应该设置为FQDN，如何维护主机名和主机的内网IP的关系；
- 综上：我们需要构建至少一套企业内部的DNS服务；

2.BIND基础应用

2.1 什么是BIND

- BIND（由美国加州大学开发并且维护的）、BIND`是一个开源、稳定、且应用广泛的DNS服务
  - 开源：指BIND服务源代码是开放的；
  - 稳定：指BIND服务运行非常稳定；
  - 广泛：政府企业、单位机构、学校、等；

2.2 BIND的组成

- BIND提供（域名解析服务、权威域名服务、DNS调试工具）
  - 域名解析服务：将域名解析为IP地址；
  - 权威域名服务：能从该服务器查询到完整域名对应的IP地址，则这台服务器就算权威解析；
  - DNS调试工具：主要提供DNS客户端调试工具，供客户端使用；

2.3 BIND服务实践

2.2.1 BIND环境准备

系统版本	外网地址	内网地址	功能及作用
Centos7	eth0：10.0.0.91	eth1：172.16.1.91	DNS-Master
Centos7	eth0：10.0.0.92	eth1：172.16.1.92	DNS-Slave
Centos7	eth0：10.0.0.93	eth1：172.16.1.93	DNS-Son

2.2.2 BIND服务安装

- `Bind`的安装非常简单，只需要通过`yum`即可完成安装；
  - `bind`提供主程序包；
  - `bind-utils`提供工具包；

```
[root@dns-master ~]# yum install bind bind-utils -y
[root@dns-master ~]# systemctl start named
[root@dns-master ~]# systemctl enable named
```

2.2.3 BIND配置文件

- 1.主配置文件格式
  - `options {}`：全局选项（监听端口、数据文件存储位置、缓存位置、权限等）
  - `logging {}`：服务日志选项
  - `zone . {}`：自定义区域配置
  - `include`：包含别的文件
- 2.主配置文件注意事项
  - 语法非常严格；
  - 文件权限属主`root`，属组`named`，文件权限`640`

- 3.主配置示例文件

```
[root@dns-master ~]# cat  /etc/named.conf
options {
  //监听地址及端口
  listen-on port 53 { localhost;Server_IP; };
  //区域配置存储目录
  directory     "/var/named";
  //dns解析过内容的缓存文件
  dump-file     "/var/named/data/cache_dump.db";
  //静态解析文件（几乎不用）
  statistics-file "/var/named/data/named_stats.txt";
  //内存的统计信息
  memstatistics-file "/var/named/data/named_mem_stats.txt";
  recursing-file  "/var/named/data/named.recursing";
  secroots-file   "/var/named/data/named.secroots";
  //允许谁本台DNS发起查询请求（localhost|ip|any）
  allow-query     { localhost; };

  //递归查询
  recursion yes;
  dnssec-enable yes;
  dnssec-validation yes;

  /* Path to ISC DLV key */
  bindkeys-file "/etc/named.root.key";

  managed-keys-directory "/var/named/dynamic";
  pid-file "/run/named/named.pid";
  session-keyfile "/run/named/session.key";
};

//控制日志输出的级别以及输出的位置
logging {
        channel default_debug {
                file "data/named.run";
                severity dynamic;
        };
};

//默认可以对任何域名提供解析服务；因为named.ca中存储的是全球根域服务器;
zone "." IN {
    type hint;
    file "named.ca";    //区域配置文件名称
};
```

```
//包含的其他文件
include "/etc/named.rfc1912.zones";
include "/etc/named.root.key";
```

2.2.4 客户端验证解析

1.配置 `DNS` 服务器指向：在 `/etc/resolv.conf` 里配置 `DNS` 的 `ip` 地址

```
[root@dns-master ~]# cat /etc/resolv.conf
# Generated by NetworkManager
nameserver 10.0.0.91
```

2.使用 `ping` 命令验证解析

```
[root@dns-master ~]# ping baidu.com
[root@dns-son ~]# ping baidu.com
PING baidu.com (220.181.38.148) 56(84) bytes of data
```

2.4 自定义区域

- 自定义域分为如下两类：
  - 主机域：
    - 1.主机域其实是一个假域；
    - 2.主机域其实是不能解析到互联网上；
    - 3.主机域它只对局域网（内网）提供服务；
  - 业务域：
    - 1.业务域一般都是真实可用的；
    - 2.业务域则为一个真正需要对外提供服务的域名；

2.4.1 自定义区域配置文件

- 区域 `zone` 文件定义在 `/etc/named.conf` 配置；
- 也可以配置在自定义的其他文件里，并在 `named.conf` 里 `include`
- 注意文件的权限，属主 `root` 属组 `named` 文件权限 `640`

```
zone "oldxu.com" IN {
    type master|slave;          //自定义区域类型
    file /path/to/zonefile;     //绝对路径和相对路径
    allow-update {ip|none};      //允许哪个ip可以使用nsupdate动态更新
区域文件
};
```

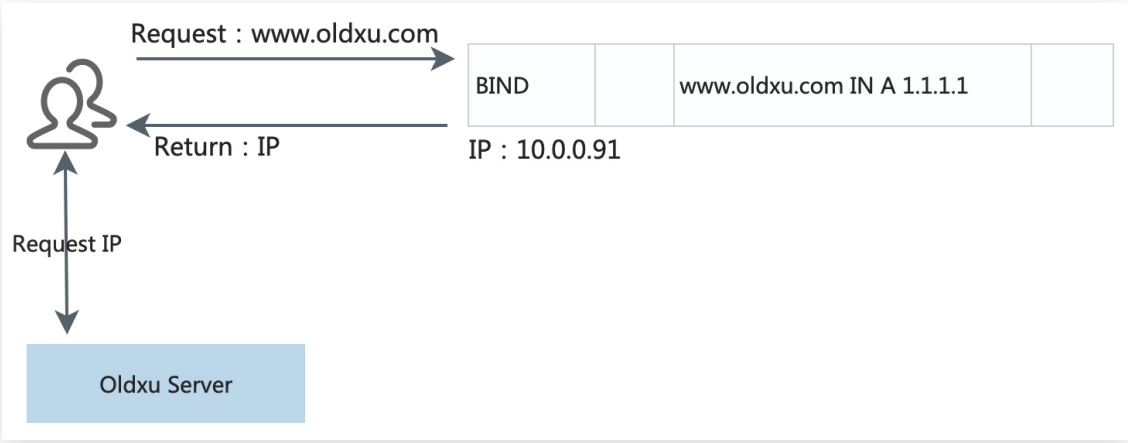
2.4.2 自定义区域数据库文件

- 范例以及编写注意事项：
  - 1.严格注意语法书写，其格式非常严格；
  - 2.记录不准许折行书写；
  - 3.单行记录开头不准许空格或tab开头；

```
[root@dns-master ~]# cat /var/named/oldxu.com.zone
$TTL 600    ; DNS失效时间，单位秒；

;区域名称    IN  SOA dns.oldxu.com. 管理员邮箱
;区域名称可以使用过@表示，@表示当前域
oldxu.com. IN SOA ns.oldxu.com. qq.oldxu.com. (
    2021041514  ; 序列号(serial number) 十进制，不能超过10位，通常使用日期时间戳，例如2018121601
    10800  ; 刷新时间(refresh time) 即每隔多久到主服务器检查一次
    900    ; 重试时间(retry time) 即刷新不成功多久后重试，应该小于refresh time
    604800 ; 过期时间(expire time) 当辅助DNS无法联系主DNS时，辅助DNS在多长时间内认为其缓存是有效的。
    86400  ; 权威应答的ttl(netgative answer ttl) ;缓存DNS服务器可以缓存记录多长时间
)

;给客户端返回NS记录，oldxu.com. 域名由哪几台权威服务器提供解析
oldxu.com.  IN  NS  ns1.oldxu.com.
oldxu.com.  IN  NS  ns2.oldxu.com.

;A记录，配置权威域名的真实IP地址；
ns1.oldxu.com. IN   A   10.0.0.91
ns2.oldxu.com. IN   A   10.0.0.92

;真正的域名解析
www.oldxu.com.  IN  A   1.1.1.1
web.oldxu.com.  IN  A   2.2.2.2
```

2.5 BIND实战场景-1

- 用户通过 `DNS` 服务器 `10.0.0.91` 解析 `www.oldxu.com`
  - 1.添加 `oldxu.com` 区域配置文件
  - 2.添加区域数据库文件，配置NS记录，返回 `DNS` 权威服务器地址
  - 3.该域的 `DNS` 权威服务器为 `10.0.0.91`
  - 4.添加该域的 `A` 记录解析



2.5.1 新增区域配置文件

1.在主配置文件 `/etc/named.conf`，新增一个 `job.com` 区域配置

```
[root@dns-master ~]# cat  /etc/named.conf
...
//自行配置权威域名解析oldxu.com
zone "oldxu.com" IN {
    type master;                        //
    file "oldxu.com.zone";  //具体解析记录配置存储至那个文件
...
```

2.检查配置文件语法

```
[root@dns-master ~]# named-checkconf
```

2.5.2 新增区域数据库文件

1.添加区域数据库文件

```
[root@dns-master ~]# cat  /var/named/oldxu.com.zone
$TTL 600
oldxu.com. IN SOA ns.oldxu.com. qq.oldxu.com. (
    2021041514
    10800
    900
    604800
    86400
)

;给客户端返回NS记录，该域名由哪台权威服务器提供解析
oldxu.com.   IN  NS  ns1.oldxu.com.
ns1.oldxu.com. IN   A   10.0.0.91
```

```
;域名解析记录
www.oldxu.com.  IN  A   1.1.1.1
test.oldxu.com. IN  A   2.2.2.2
```

2.检查区域数据库文件配置

```
[root@dns-master ~]# named-checkzone oldxu.com
/var/named/oldxu.com.zone
```

3.重载 `DNS` 服务器

```
[root@dns-master ~]# rndc reload
```

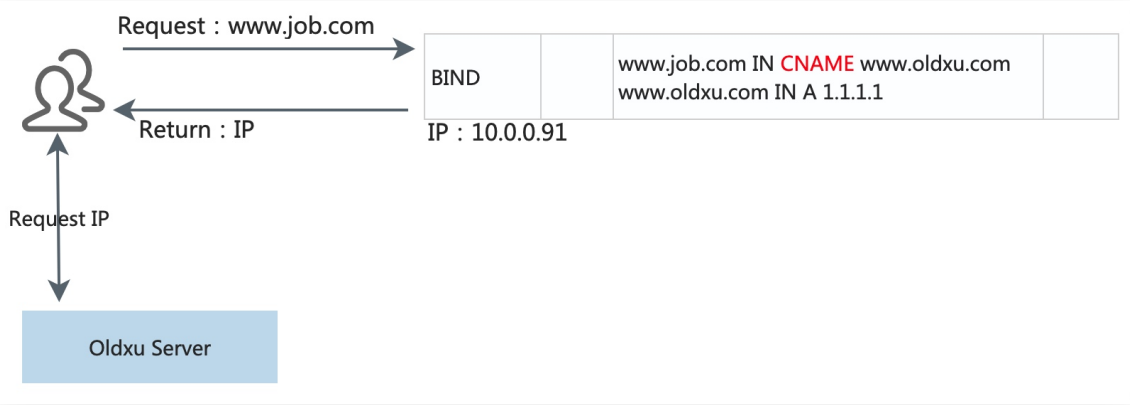
2.5.3 客户端测试解析域名

- 客户端测试域名解析

```
[root@client ~]# dig www.oldxu.com  @10.0.0.91
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.oldxu.com.          IN  A

;; ANSWER SECTION:
www.oldxu.com.       7200    IN  A   1.1.1.1

;; AUTHORITY SECTION:
oldxu.com.       7200    IN  NS  dns1.oldxu.com.

;; ADDITIONAL SECTION:
dns1.oldxu.com.      7200    IN  A   10.0.0.91
```

2.6 BIND实战场景-2

- 用户通过 `DNS` 服务器 `10.0.0.91` 解析 `www.job.com`
    - 1.添加 `job.com` 区域配置文件
    - 2.添加区域数据库文件，配置NS记录，返回 `DNS` 权威服务器地址
    - 3.该域的 `DNS` 权威服务器为 `10.0.0.91`
    - 4.添加该域的 `CNAME` 解析记录，解析至 `www.oldxu.com` 域名上



2.6.1 新增区域配置文件

1.在主配置文件 `/etc/named.conf`，新增一个 `job.com` 区域配置

```
[root@dns-master ~]# vim /etc/named.conf
//添加job.com权威的域名解析
zone "job.com" IN {
        type master;
        file "job.com.zone";
};
```

2.检查配置文件语法

```
[root@dns-master ~]# named-checkconf
```

2.6.2 新增区域数据库文件

1.配置 `job.com`的 `CNAME` 解析记录

```
[root@dns-master ~]# cat  /var/named/job.com.zone
$TTL 600
job.com. IN SOA ns1.job.com. qq.job.com. (
    2021041515
    10800
    900
    604800
    86400
)

job.com.         IN   NS  ns1.job.com.
dns1.job.com.    IN   A    10.0.0.91

;CNAME解析
www.job.com.     IN  CNAME   www.oldxu.com.
```

2.检查区域数据库文件配置

```
[root@dns-master ~]# named-checkzone job.com
/var/named/job.com.zone
```

3.重载 `DNS` 服务器

```
[root@dns-master ~]# rndc reload
```

2.6.3 客户端测试解析域名

```
[root@client ~]# dig www.job.com
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 1,
ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.job.com.            IN  A

;; ANSWER SECTION:
www.job.com.        7200    IN  CNAME   www.oldxu.com.
www.oldxu.com.      7200    IN  A   1.1.1.1

;; AUTHORITY SECTION:
oldxu.com.      7200    IN  NS  dns1.oldxu.com.

;; ADDITIONAL SECTION:
ns1.oldxu.com.      7200    IN  A   10.0.0.91
```

2.7 BIND实战场景-3

- 配置反向解析 `PTR：IP-->FQDN`
  - 1.反向区域文件名称为 `逆向网络地址` 加 `.in-addr.arpa.` 后缀组成
  - 2.反向区域数据库文件，例如 `10.0.0.200` 的 `name` 为 `200`，完全格式为 `200.0.0.10.in-addr.arpa.`
- 建议维护一个主机域，然后在维护主机域的反向解析；（对于业务域则无需维护反向解析）

2.7.1 新增区域配置文件

1.在主配置文件 `/etc/named.conf`，新增一个反向区域配置

```
[root@dns-master ~]# vim /etc/named.conf
//配置反向解析
zone "0.0.10.in-addr.arpa" IN {
        type master;
        file "0.0.10.zone";
};
```

2.检查配置文件语法

```
[root@dns-master ~]# named-checkconf
```

2.7.2 新增区域数据库文件

1.配置反向解析数据文件

```
[root@dns-master ~]# cat /var/named/0.0.10.zone
$TTL 7200
@ IN SOA 0.0.10.in-addr.arpa. qq.oldxu.com. (
    2021041515
    10800
    900
    604800
    86400
)

;给客户端返回NS记录，该域名由哪台权威服务器提供解析
@       IN      NS      ns1.oldxu.com.

;权威DNS的反向解析
91      IN  PTR ns1.oldxu.com.

; www反向解析
200     IN  PTR www.oldxu.com.
```

2.7.3 客户端测试解析域名

```
[root@client ~]# dig -x 10.0.0.200
;; QUESTION SECTION:
;200.0.0.10.in-addr.arpa.   IN  PTR

;; ANSWER SECTION:
200.0.0.10.in-addr.arpa. 7200   IN  PTR www.oldxu.com.

;; AUTHORITY SECTION:
0.0.10.in-addr.arpa.    7200    IN  NS  ns1.oldxu.com.

;; ADDITIONAL SECTION:
ns1.oldxu.com.      7200    IN  A   10.0.0.91
```

2.8 DNS客户端工具

	通用性	使用难易度
nslookup	支持多平台、应用广泛	使用简单、易懂
dig	常用语linux系统	比较专业
host	较多	简单、明了

2.8.1 host

- 查询SOA记录

```
[root@client ~]# host -t SOA baidu.com
baidu.com has SOA record dns.baidu.com. sa.baidu.com. 2012144174
300 300 2592000 7200
```

- 查询NS记录

```
[root@client ~]# host -t NS baidu.com
baidu.com name server ns3.baidu.com.
baidu.com name server ns7.baidu.com.
baidu.com name server ns4.baidu.com.
baidu.com name server dns.baidu.com.
baidu.com name server ns2.baidu.com.
```

- 查询A记录

```
[root@client ~]# host -t A baidu.com
baidu.com has address 220.181.38.148
baidu.com has address 39.156.69.79
```

2.8.2 nslookup

- 解析域名对应的IP

```
[root@client ~]# nslookup www.baidu.com
Server:     223.5.5.5
Address:    223.5.5.5#53

Non-authoritative answer:
www.baidu.com   canonical name = www.a.shifen.com.
Name:   www.a.shifen.com
Address: 110.242.68.3
Name:   www.a.shifen.com
Address: 110.242.68.4
```

- 解析域名SOA记录

```
[root@client ~]# nslookup
> set q=soa                    # 设定q=soa|ns|a
> baidu.com
Server:     223.5.5.5
Address:    223.5.5.5#53

Non-authoritative answer:
baidu.com
    origin = dns.baidu.com
    mail addr = sa.baidu.com
    serial = 2012144174
    refresh = 300
    retry = 300
    expire = 2592000
    minimum = 7200
```

2.8.3 dig

- 通过哪个 `dns` 来解析域名，正向解析

```
[root@client ~]# dig @223.5.5.5 www.baidu.com
```

- 通过 `ip` 解析对应的域名，反向解析

```
[root@client ~]# dig -x 39.104.16.126 @223.5.5.5
```

- 通过 `dig` 仅查询a记录

```
[root@client ~]# dig -t a baidu.com
```

- 通过 `dig` 查看区域传送配置

```
[root@client ~]# dig -t AXFR oldxu.com @10.0.0.91
```

3.DNS递归查询

3.1 什么是递归查询

- 如果你要建立一个授权域名服务器服务器，仅提供本地已存在域名解析即可；那么不要开启 `recursion` 功能。
- 如果你要建立一个递归 `DNS` 服务器, 那么需要开启 `recursion` 功能。
- 如果你的递归DNS服务器有公网IP地址, 你必须开启访问控制功能，只有合法用户才可以发询问。

3.2 递归查询配置参数

参数	选项	作用
recurison	yes/no	是否开启递归查询请求
allow-recursion	{address_match_list | any | none };	限制客户端递归请求的范围

3.3 递归查询场景实践

3.3.1 开启递归查询

- BIND默认配置中的 `recurison` 参数是启用的;
    - 1.配置仅允许 `10.0.0.0` 网段用户可以查询（可选）；
    - 2.使用客户端查询系统中存在的域名;
    - 3.使用客户端查询系统中不存在的域名（让`BIND`进行递归查询）；

1.修改 `/etc/named.conf` 配置文件

```
[root@dns-master ~]# vim /etc/named.conf
recursion yes;
# allow-recursion  {10.0.0.0/24;172.16.1.0/24};

[root@dns-master ~]# rndc reload
```

1.查询存在 `BIND` 服务中的域名；

```
[root@client ~]# dig www.oldxu.com @10.0.0.91
...
;; ANSWER SECTION:
www.oldxu.com.      7200    IN  A   1.1.1.1

;; AUTHORITY SECTION:
oldxu.com.      7200    IN  NS  dns1.oldxu.com.

;; ADDITIONAL SECTION:
dns1.oldxu.com.     7200    IN  A   10.0.0.91
...
```

2.查询不存在 `BIND` 服务中的域名；能获得正确返回，原因是`BIND`进行了递归查询；

```
[root@client ~]# dig www.qq.com @10.0.0.91
...
;; QUESTION SECTION:
;www.qq.com.            IN  A

;; ANSWER SECTION:
www.qq.com.     300 IN  CNAME   ins-r23tsuuf.ias.tencent-
cloud.net.
ins-r23tsuuf.ias.tencent-cloud.net. 108 IN A    61.241.54.232
ins-r23tsuuf.ias.tencent-cloud.net. 108 IN A    61.241.54.211

;; AUTHORITY SECTION:
tencent-cloud.net.  172501  IN  NS  ns-open3.qq.com.
tencent-cloud.net.  172501  IN  NS  ns-open2.qq.com.
tencent-cloud.net.  172501  IN  NS  ns-open1.qq.com.
```

3.3.1 关闭递归查询

1.修改 `/etc/named.conf` 配置文件，关闭递归查询；

```
[root@dns-master ~]# vim /etc/named.conf
recursion no;
[root@dns-master ~]# rndc reload
```

2.查询本地存在域名会成功返回；因为本地无需进行递归查询，即可返回权威解析；

```
[root@client ~]# dig www.oldxu.com @10.0.0.91
;; QUESTION SECTION:
;www.oldxu.com.          IN   A

;; ANSWER SECTION:
www.oldxu.com.       7200    IN   A    1.1.1.1

;; AUTHORITY SECTION:
oldxu.com.       7200    IN   NS   dns1.oldxu.com.

;; ADDITIONAL SECTION:
dns1.oldxu.com.      7200    IN   A    10.0.0.91
```

3.由于本地无法返回权威结果，同时也禁用了递归查询，所以查询本地不存在域名会有 `WARNING` 提示；

```
[root@client ~]# dig www.qq.com @10.0.0.91
# 递归请求不可用
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.qq.com.             IN   A
```

4.DNS主辅同步

4.1 DNS主辅同步概念

- 辅助DNS是DNS容灾备份服务：在主DNS和辅DNS之间建立区域数据传输机制，当主DNS遇到故障或者服务中断时，辅DNS仍可以继续提供解析服务，因此保障业务稳定运行。
- 辅助DNS的优势：
  - 容灾备份，降低业务中断风险：主DNS系统故障，辅助DNS可继续提供域名解析服务，保障业务可用性。
  - 负责均衡，流量均摊降低负载：当辅助DNS与主DNS同时对外提供解析服务时，可以达到流量负载均衡的效果。

- 阿里云dns
  - 223.5.5.5
  - 223.6.6.6

4.2 DNS主从同步原理

4.3 DNS主从同步场景

4.3.1 主从环境准备

- 1.确保防火墙规则开放（建议关闭）
- 2.保持主从服务器时钟一致；
- 3.搭建完主从后，若修改主服务器域配置，`Serail Number`必须递增，否则不同步；

4.3.2 主辅同步配置要点

- DNS 主辅同步配置要点：
  - 主`DNS`的`named.conf`里配置`allow-transfer`和`also-notify`选项；
  - 辅助`DNS`主配置文件`option`段添加`masterfile-format text`，否则同步的文件为`data`类型；
  - 辅助`DNS`添加区域配置文件，类型为`slave`，同时指向`masters`参数指向`master`地址；
  - 辅助`DNS`不可主动修改`DNS`数据库文件；

4.3.3 Master服务器配置

1.添加区域配置文件

```
[root@dns-master ~]# vim  /etc/named.conf
options {
    allow-transfer {10.0.0.92;};    //允许哪个地址能同步Master配置信息
    also-notify {10.0.0.92;};        //主动通知辅助DNS域名变更
}

//自行配置权威域名解析
zone "oldxu.com" IN {
        type master;
        file "oldxu.com.zone";
        notify yes;
};
```

2.添加区域数据文件

```
[root@dns-master ~]# cat /var/named/oldxu.com.zone
$TTL 600    ; DNS失效时间，单位秒；
oldxu.com. IN SOA ns.oldxu.com. qq.oldxu.com. (
    2021041515
    10800
    900
    604800
    86400
)

;给客户端返回NS记录，由于是主辅模式，所以需要两台解析
oldxu.com.  IN  NS  ns1.oldxu.com.
oldxu.com.  IN  NS  ns2.oldxu.com.

;将两条NS记录指向两台权威的DNS地址
ns1.oldxu.com. IN   A   10.0.0.91
ns2.oldxu.com. IN   A   10.0.0.92

;域名解析
www.oldxu.com.  IN  A   1.1.1.1
```

3.检测语法，重启服务

```
[root@dns-master ~]# named-checkconf
[root@dns-master ~]# rndc reload
```

4.3.4 Slave服务器配置

1.安装bind服务

```
[root@dns-slave ~]# yum install bind-utils -y
```

2.修改主配置文件;

```
[root@dns-slave ~]# vim /etc/named.conf
options {
    ...
    listen-on port 53 { any; };
    allow-query     { any; };
    masterfile-format text;
    ...
```

3.添加区域配置文件，类型为 `slave`，然后指向 `master`地址;

```
[root@dns-slave ~]# vim /etc/named.conf
zone "oldxu.com" IN {
        type slave;
        file "slaves/oldxu.com.zone";
        masters {10.0.0.91;};
};
```

4.检查语法，重启服务

```
[root@dns-slave ~]# named-checkconf
[root@dns-slave ~]# rndc reload
```

4.3.4 测试主从解析

- 使用 `dig` 进行域名解析，先通过 `master`解析，然后在使用 `slave` 测试解析

1.通过 `Master` 解析

```
[root@client ~]# dig www.oldxu.com @10.0.0.91
;; ANSWER SECTION:
ops.oldxu.com.      7200    IN  A   1.1.1.1

;; AUTHORITY SECTION:
oldxu.com.      7200    IN  NS  ns2.oldxu.com.
oldxu.com.      7200    IN  NS  ns1.oldxu.com.

;; ADDITIONAL SECTION:
ns1.oldxu.com.      7200    IN  A   10.0.0.91
ns2.oldxu.com.      7200    IN  A   10.0.0.92
```

2.通过 `Slave` 解析

```
[root@client ~]# dig www.oldxu.com @10.0.0.92
;; ANSWER SECTION:
ops.oldxu.com.      7200    IN  A   1.1.1.1

;; AUTHORITY SECTION:
oldxu.com.      7200    IN  NS  ns1.oldxu.com.
oldxu.com.      7200    IN  NS  ns2.oldxu.com.

;; ADDITIONAL SECTION:
ns1.oldxu.com.      7200    IN  A   10.0.0.91
ns2.oldxu.com.      7200    IN  A   10.0.0.92
```

4.3.5 测试主从同步

- 新增一条记录；然后滚动 serial

1.`Master` 节点新增一条记录

```
[root@dns-master ~]# cat  /var/named/oldxu.com.zone
$TTL 7200    ; DNS失效时间，单位秒；
oldxu.com. IN SOA ns.oldxu.com. qq.oldxu.com. (
    2021041516
    10800
    900
    604800
    86400
)

;给客户端返回NS记录，由于是主辅模式，所以需要两台解析
oldxu.com.  IN  NS  ns1.oldxu.com.
oldxu.com.  IN  NS  ns2.oldxu.com.
```

```
;将两条NS记录指向两台权威的DNS地址
ns1.oldxu.com. IN   A   10.0.0.91
ns2.oldxu.com. IN   A   10.0.0.92

;域名解析
www.oldxu.com.  IN  A   1.1.1.1
dev.oldxu.com.  IN  A   2.2.2.2
ttt.oldxu.com.  IN  A   3.3.3.3

[root@dns-master ~]# rndc reload
```

2.使用 `Master` 以及 `Slave` 地址测试新添加的记录是否能解析

```
[root@client ~]# dig ttt.oldxu.com @10.0.0.91
[root@client ~]# dig ttt.oldxu.com @10.0.0.92
;; ANSWER SECTION:
ttt.oldxu.com.      7200    IN  A   3.3.3.3

;; AUTHORITY SECTION:
oldxu.com.      7200    IN  NS  ns2.oldxu.com.
oldxu.com.      7200    IN  NS  ns1.oldxu.com.

;; ADDITIONAL SECTION:
ns1.oldxu.com.      7200    IN  A   10.0.0.91
ns2.oldxu.com.      7200    IN  A   10.0.0.92
```

4.3.6 配置DNS高可用

- `Linux` 配置

```
[root@client]# cat /etc/resolv.conf
# Generated by NetworkManager
nameserver 10.0.0.91
nameserver 10.0.0.92
```

- Windows 配置

5.DNS子域授权

5.1 什么是DNS子域授权

- A服务负责（oldxu.com）域名解析，授权B服务器管理（ops.oldxu.com）的域名解析；
  - 父域：oldxu.com
  - 子域：ops.oldxu.com
    - www.ops.oldxu.com
    - test.ops.oldxu.com

5.2 DNS子域授权环境

环境	域名	节点地址
父域（Master）	oldxu.com	10.0.0.91
父域（Slave）	oldxu.com	10.0.0.92
子域	ops.oldxu.com	10.0.0.93

5.3 DNS子域授权场景

5.3.1 父域配置（Master）

1.区域配置文件此前可以复用之前主辅配置，所以无需修改；

```
[root@dns-master ~]# cat /etc/named.conf
options {
    allow-transfer {10.0.0.92;};
    also-notify {10.0.0.92;};
}

zone "oldxu.com" IN {
    type master;
    file "oldxu.com.zone";
    notify yes;
};
```

2.区域文件数据库文件，将 ops 三级子域授权给子域服务器；

```
[root@dns-master ~]# vim /var/named/oldxu.com.zone
$TTL 7200    ; DNS失效时间，单位秒；
oldxu.com. IN SOA ns.oldxu.com. qq.oldxu.com. (
    2021041516
```

```
        10800
        900
        604800
        86400
)

;给客户端返回NS记录，该域名由哪台权威服务器提供解析
oldxu.com.   IN  NS  ns1.oldxu.com.
oldxu.com.   IN  NS  ns2.oldxu.com.

;权威DNS的地址,由于权威服务器在本机，所以填写本机IP
ns1.oldxu.com. IN   A   10.0.0.91
ns2.oldxu.com. IN   A   10.0.0.92

;真正的域名解析
www.oldxu.com.   IN  A   1.1.1.1
dev.oldxu.com.   IN  A   2.2.2.2
ttt.oldxu.com.   IN  A   3.3.3.3

;子域配置（通常子域也应该是主从模式，如果为主从，则应该授权两台NS记录）
ops.oldxu.com.       IN  NS  ns1.ops.oldxu.com.
ns1.ops.oldxu.com.   IN  A   10.0.0.93
```

3.检查配置文件，区域文件，重载服务；

```
[root@dns-master ~]# named-checkconf
[root@dns-master ~]# rndc reload
```

5.3.2 子域配置（Other）

1.安装 `bind` 服务

```
[root@dns-slave ~]# yum install bind-utils -y
```

2.修改配置文件，然后增加子域的 `zone` 配置文件

```
[root@dns-son ~]# vim /etc/named.conf
options {
    ...
        listen-on port 53 { 127.0.0.1;any; };
        allow-query     { localhost; any; };
    ...

//增加如下配置
zone "ops.oldxu.com" IN {
        type master;
        file "ops.oldxu.com.zone";
};
```

3.添加区域数据数据库文件

```
[root@dns-son ~]# cat /var/named/ops.oldxu.com.zone
$TTL 7200
ops.oldxu.com. IN SOA ns.ops.oldxu.com. qq.oldxu.com. (
    2021041516
    10800
    900
    604800
    86400
)

ops.oldxu.com.      IN  NS  ns1.ops.oldxu.com.
ns1.ops.oldxu.com.  IN  A   10.0.0.93


;配置子域解析记录
www.ops.oldxu.com.  IN  A   4.4.4.4
bbs.ops.oldxu.com.  IN  A   5.5.5.5
```

4.检查语法，启动服务

```
[root@dns-son ~]# named-checkconf
[root@dns-son ~]# systemctl start named
[root@dns-son ~]# systemctl enable named
```

5.3.3 结果验证

1.客户端通过获取子域对应的解析（使用子域的IP）；

```
[root@client ~]# dig www.ops.oldxu.com @10.0.0.93
;; QUESTION SECTION:
;www.ops.oldxu.com.      IN   A

;; ANSWER SECTION:
www.ops.oldxu.com.   7200    IN   A    4.4.4.4

;; AUTHORITY SECTION:
ops.oldxu.com.       7200    IN   NS   ns1.ops.oldxu.com.

;; ADDITIONAL SECTION:
ns1.ops.oldxu.com.   7200    IN   A    10.0.0.93
```

2.客户端通过获取子域对应的解析（使用父域的IP）；

```
[root@client ~]# dig www.ops.oldxu.com @10.0.0.91
[root@client ~]# dig www.ops.oldxu.com @10.0.0.92
;; QUESTION SECTION:
;www.ops.oldxu.com.      IN   A

;; ANSWER SECTION:
www.ops.oldxu.com.   4388    IN   A    4.4.4.4

;; AUTHORITY SECTION:
ops.oldxu.com.       7200    IN   NS   ns1.ops.oldxu.com.
```

3.客户端获取父域的解析（使用子域的IP）；

```
# 无法正常解析；
[root@client ~]# dig  www.oldxu.com @10.0.0.93
```

- 问题：由于父域与子域互相维护不同的区域配置，它们之间并不存在任何的联系，所以子域在解析父域的域名时，它并不会直接通过父域来获取权威的解析记录，那它会怎么做呢?
  - 第一步：它会先找顶点根域；
  - 第二步：寻找找 `com` 域对应的 `DNS` 服务器；
  - 第三步：寻找 `oldxu` 域对应的 `DNS` 服务器，而后获取 `www` 对应的解析记录；
  - 这种查找模式是由 `DNS` 的机制所决定的；
- 解决的方法：明确告诉子域，让其能找到父域进项查询解析，而无需查找根域；（需要配置 `DNS` 的转发）

6.DNS转发模式

6.1 什么是DNS转发

- 转发指的是将域名查询请求，转至某一台服务器解析（被转发的服务器必须允许为当前服务器做递归）
- 转发分为两类；
  - 区域转发：仅转发对某特定区域的解析请求；
  - 全局转发：针对本地没有通过 `zone` 定义的区域查询请求，统统转发；

6.2 DNS转发示例

- 区域转发示例配置：

```
zone "ZONE_NAME" IN {
    type forward;
    forward { first | only };
    forwarders { SERVER_IP; };
};
```

- 全局转发示例配置：

```
options {
    ...
    forward { first | only };
    forwarders { SERVER_IP; };
    ...
}
```

- 转发参数含义：
  - `forwarders`：转发给哪台服务器；可以多台；
  - `forwarder only`：仅转发
  - `forwarder first`：优先转发给对应的服务器查询，如转发器未响应，则自行迭代查询

6.3 DNS区域转发实践

6.3.1 子域配置转发

1.在子域服务器上，添加父域的域名，然后配置转发；

```
[root@dns-son ~]# vim /etc/named.conf
zone "oldxu.com" IN {
        type forward;
        forward first;
        forwarders { 10.0.0.91; 10.0.0.92; };
};
```

2.检查语法，重载服务；

```
[root@dns-slave ~]# named-checkconf
[root@dns-slave ~]# rndc reload
```

6.3.1 测试转发效果

- 解析父域的域名，通过子域的地址；

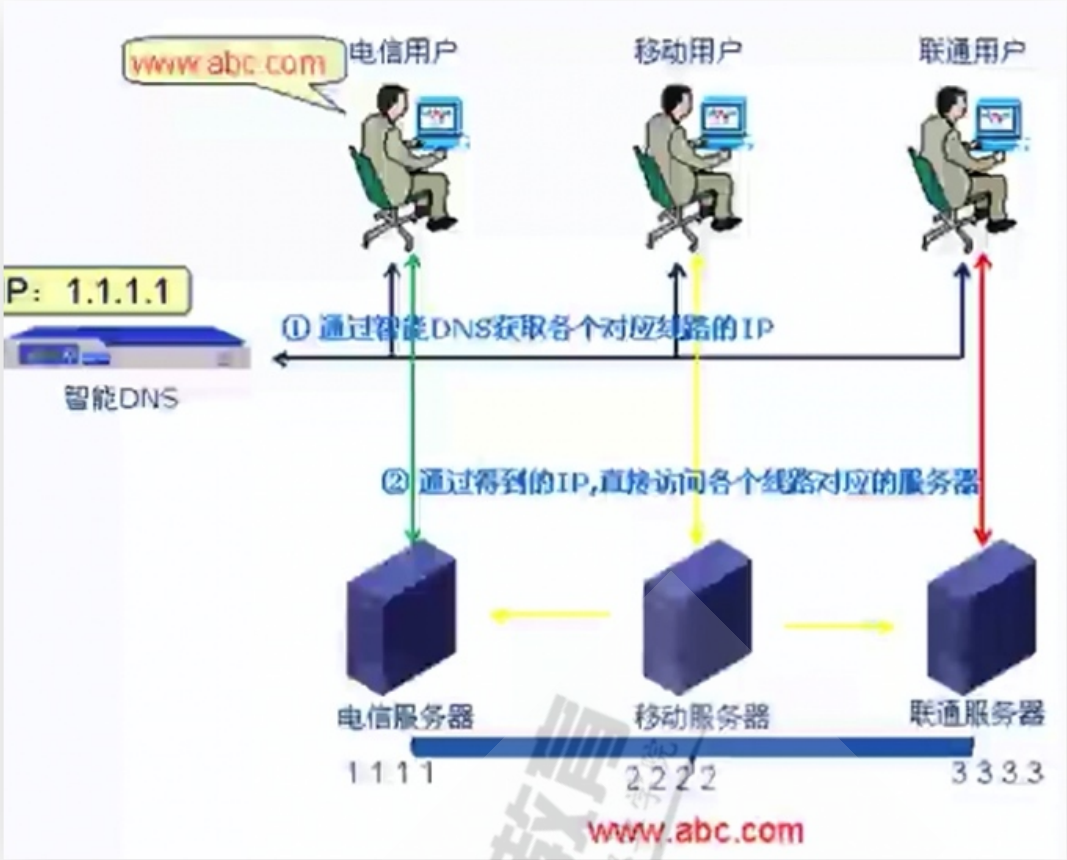
```
[root@dns-son ~]# dig www.oldxu.com @10.0.0.93 +short
1.1.1.1
```

7.智能DNS概述

- 智能DNS就是根据用户的来源地域，自动智能化判断来路IP返回给用户，而不需要用户进行选择；

7.1 什么是智能DNS

- DNS解析，不判断访问者来源，会随机选择其中一个IP地址返回给访问者。
- 智能DNS解析，会判断访问者的来源，为不同的访问者智能返回不同的IP地址，可使访问者在访问网站时可获取用户指定的IP地址，能够减少解析时延，并提升网站访问速度的功效。
- 比方一个企业的站点三个运营商的带宽都有：电信、联通、移动；同样来自三个不同 运营商网络的访问用户，假设电信用户访问企业网址的时候，智能DNS会自动根据IP判断，再从电信返回给电信用户，其他的也同理；
    - 电信用户：访问 `www.oldxu.com` 返回`1.1.1.1`
    - 联通用户：访问 `www.oldxu.com` 返回`2.2.2.2`
    - 移动用户：访问 `www.oldxu.com` 返回`3.3.3.3`



7.2 ACL访问控制列表

- ACL访问控制列表，是用来限制哪些主机可以通过DNS查询，哪些不可以；
- 系统默认内置了四种ACL
  - `any`：允许所有主机节点查询；
  - `none`：拒绝所有主机节点查询；
  - `localhost`：仅允许本地接口网络主机查询；
  - `localnet`：本地子网所有IP；
- 当然内置的可能无法满足企业需求，所以我们也可以自定义ACL规则；

```
// 简单ACL规则定义
acl "ips" {                              //定义一个名为ips的ACL
  10.0.0.1; 10.0.0.2; 192.168.1.1;  //包含3个单个IP
  172.16.1.0/24;
  ...
 };

 //复杂acl规则定义
 acl "all_rule" {            //定义一个名为all_rule的ACL
  "ips";                     //可以嵌套包含其他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）

 };
```

定义的acl规则如何使用

```
allow-update { "ips"; };              //允许谁能更新
allow-transfer { "all_rule"; };       //允许谁能同步
...
```

7.3 BIND-VIEW功能

- `view` 语句定义了视图功能，视图是BIND9提供的强大的新功能，允许DNS服务器根据不同的客户端，请求相同的域名，但返回不同的解析结果；
- `view`语法示例

```
view view_name [class] {
    match-clients { address_match_list } ;
    match-destinations { address_match_list } ;
    match-recursive-only { yes_or_no } ;
    [ view_option; ...]
};
```

7.3 场景1-根据不同环境解析

- 维护一个内网的主机域；根据不同IP请求，返回不同的解析结果；

1.修改 `/etc/named.conf` 配置文件

```
[root@dns-master ~]# vim  /etc/named.conf
...
...

//模拟测试业务地址段
acl "env-test" {
    10.0.0.5;
};

//模拟生产业务地址段
acl "env-prod" {
    10.0.0.6;10.0.0.200;
};
```

```
view "env-test-project" {
    match-clients { "env-test"; };
    recursion yes;
    zone "mg.com" {
        type master;
        file "env-test.mg.com.zone";
    };
};
view "env-prod-project" {
    match-clients { "env-prod"; };
    recursion yes;
    zone "mg.com" {
        type master;
        file "env-prod.mg.com.zone";
    };
};

view "default" {
    match-clients { any; };
    recursion yes;
    zone "." IN {
    type hint;
    file "named.ca";
    };
    include "/etc/named.rfc1912.zones";
};

#include "/etc/named.rfc1912.zones";
include "/etc/named.root.key";
```

2.准备测试环境的 `zone` 区域配置文件

```
[root@dns-master ~]# cat /var/named/env-test.mg.com.zone
$TTL 7200    ; DNS失效时间，单位秒；
mg.com. IN SOA ns1.mg.com. qq.mg.com. (
    2021041518
    10800
    900
    604800
    86400
)

;给客户端返回NS记录，该域名由哪台权威服务器提供解析
mg.com. IN  NS  ns1.mg.com.
```

```
;权威DNS的地址,由于权威服务器在本机，所以填写本机IP
ns1.mg.com. IN  A   10.0.0.91

;真正的域名解析
www.mg.com. IN  A   1.1.1.1
```

3.准备生产环境的 `zone` 区域配置文件

```
[root@dns-master ~]# cat /var/named/env-prod.mg.com.zone
$TTL 7200   ; DNS失效时间，单位秒;
mg.com. IN SOA ns1.mg.com. qq.mg.com. (
    2021041517
    10800
    900
    604800
    86400
)

;给客户端返回NS记录，该域名由哪台权威服务器提供解析
mg.com. IN  NS  ns1.mg.com.

;权威DNS的地址,由于权威服务器在本机，所以填写本机IP
ns1.mg.com. IN  A   10.0.0.91

;真正的域名解析
www.mg.com. IN  A   2.2.2.2
```

4.使用不同客户端进行测试;

```
# 10.0.0.5解析结果
[root@lb01 ~]# dig www.mg.com @10.0.0.91 +short
1.1.1.1

# 10.0.0.6解析结果
[root@route ~]# dig www.mg.com @10.0.0.91 +short
2.2.2.2
```

7.4 场景2-智能DNS

```
...
...

//电信IP访问控制列表
```

```
acl "telecomip"{ telecom_IP; ... };

//联通IP访问控制列表
acl "netcomip"{ netcom_IP; ... };


view "telecom" {
    match-clients { "telecomip"; };
    zone "ZONE_NAME" IN {
        type master;
        file "ZONE_NAME.telecom.zone";
    };
};

view "netcom" {
    match-clients { "netcomip"; };
    zone "ZONE_NAME" IN {
        type master;
        file "ZONE_NAME.netcom.zone";
    };
};

view "default" {
    match-clients { any; };
    zone "ZONE_NAME" IN {
        type master;
        file "ZONE_NAME.zone";
    };
};
```