学习了理论怎么够，我们继续发车，通过一些命令和抓包来实际看看域名解析的奥秘。

一、nslookup命令

如何查看域名对应的IP地址呢？除了使用ping命令外，如果需要获取更多的信息，我们可以使用简单的nslookup命令。

nslookup命令用于查询DNS的记录，查看域名解析是否正常，在网络故障的时候用来诊断网络问题。

下面是常用的命令：

```
域名    # 进行正向解析
IP地址    # 进行反向解析(有些地址不能反向解析)
lserver + IP地址    # 更换DNS服务器
set all    # 显示当前的查询设置
set type=MX    # 查询邮件交换记录
set type=ns    # 查询某子域的域名服务器
set type=AAAA        #查询域名对应的IPv6地址(查询IPv6站点有效:如
bbs6.ustc.edu.cn)
```

交互式中可以进行一些设置，比如设置DNS服务器地址为电信的114.114.114.114。

当然了，如果是为了查看域名解析结果和所用的DNS服务器地址，直接使用非交互式也可，即输入：

```
nslookup  www.baidu.com
```

```
[C:\~]$ nslookup www.baidu.com
非权威应答:
服务器:  UnKnown
Address:  192.168.101.1

名称:    www.a.shifen.com
Addresses:  180.101.49.11
          180.101.49.12
Aliases:  www.baidu.com
```

首先192.168.101.1是使用的DNS服务器地址，实际上这个地址是我的路由器，也就是说路由器在这里又充当了DNS服务器的角色，天呐，再次感谢你，路由器！

下面出现了 www.a.shifen.com 域名，这是什么？下面两个IP地址，又是什么？

```
[C:\~]$ ping www.baidu.com

正在 Ping www.a.shifen.com [180.101.49.11] 具有 32 字节的数据:
来自 180.101.49.11 的回复: 字节=32 时间=6ms TTL=52
来自 180.101.49.11 的回复: 字节=32 时间=6ms TTL=52
来自 180.101.49.11 的回复: 字节=32 时间=6ms TTL=52
来自 180.101.49.11 的回复: 字节=32 时间=6ms TTL=52

180.101.49.11 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 6ms, 最长 = 6ms, 平均 = 6ms

[C:\~]$
```

从ping的结果来看，180.101.49.11就是域名解析后的IP地址，我们实际抓包看下，在执行nslookup命令之前我们开启wireshark，抓出来的结果为：

要理解上面的抓包，不得不先介绍下“A记录”和“CNAME记录”两个概念。

二、A记录和CNAME记录

什么是A记录?

比如我有一个私人域名叫做 oursnail.cn ，对应的公网IP为111.231.119.253，这里的111.231.119.253即A记录。

> A记录和AAAA记录：二者都是指向一个IP地址，但对应的IP版本不同。A记录指向IPv4地址，AAAA记录指向IPv6地址。AAAA记录是A记录的升级版本。

什么是CNAME记录?

CNAME记录又成为Alias记录，即别名的意思，有什么用呢?

假设我们有一个云服务器，申请的域名为 `oursnail.cn` ，可以同时提供网页、邮件和地图三个服务，分别对应三个三级域名： `www.oursnail.cn` 、`mail.oursnail.cn`以及`map.oursnail.cn`，由于是对应的同一台主机，只有一个公网IP为111.231.119.253。

那么这里的域名解析怎么做呢？容易想到的方案是做三个A记录：

```
www.oursnail.cn  A 111.231.119.253
mail.oursnail.cn A 111.231.119.253
map.oursnail.cn A 111.231.119.253
```

但是假设有一天，我的公网IP地址发生了变化呢？是不是需要同时修改这三个DNS的解析记录?

还有一个方案，我们引入别名，按照下面这样配置：

```
www.oursnail.cn  A 111.231.119.253
mail.oursnail.cn CNAME  www.oursnail.cn
map.oursnail.cn CNAME  www.oursnail.cn
```

即DNS解析到`mail.oursnail.cn`或`map.oursnail.cn`两个域名的时候，会“跳”到 `www.oursnail.cn` 上，从而通过 `www.oursnail.cn` 域名对应的A记录找到最终的111.231.119.253。当IP发生变化时，也只是改变www对应的A记录即可，无需改动其他两个。

或许有朋友觉得，这个真的有必要吗？大型网站的管理员会告诉你真的有必要，当IP发生变化的时候，没有人愿意去修改N条A记录。

回到上面的抓包截图，我们可以看到我们希望查询到 `www.baidu.com` 对应的A记录，第一行是查询的请求，第二行是返回结果。

首先将 `www.baidu.com` 指到别名 `www.s.shifen.com` 上，这个 `www.s.shifen.com` 域名解析出来是两个A记录，分别为180.101.49.12和180.101.49.11。

从ping的结果上可以看到，我们最终只会使用到其中一个IP，我们这里用到了180.101.49.11。接下来我们的主机将与此IP建立三次握手的TCP连接，从而能够访问到百度。

但是这里出现的两个A记录该如何理解呢?

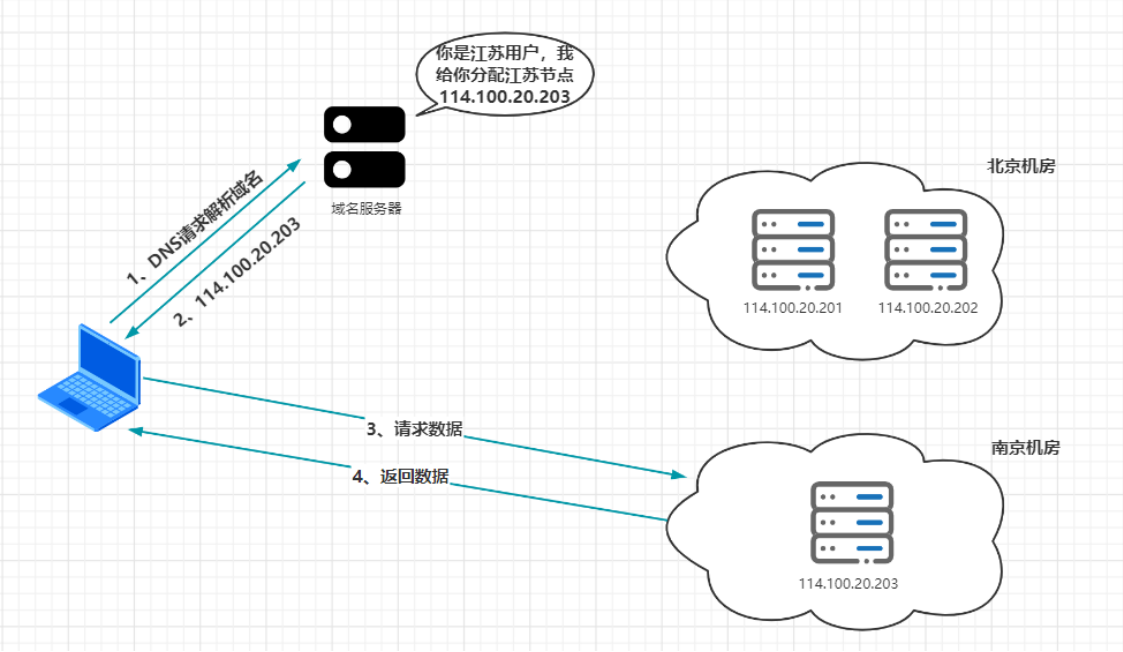
三、DNS还有妙用

上面发现A记录出现了两个，这里针对一个域名出现两个或多个A记录时，代表着什么呢?

DNS除了能解析域名之外还具有负载均衡的功能，比如我的域名 www.oursnail.cn ，添加多个A记录：

```
www.oursnail.cn  A 114.100.20.201
www.oursnail.cn  A 114.100.20.202
www.oursnail.cn  A 114.100.20.203
```

每次域名解析请求可以根据负载均衡算法计算出一个合适的IP地址并返回，比如以用户IP判断出地区，根据地区找到离他最近的服务器为其服务（这里只是说的其中一种可能算法方便理解，下面以七牛云CDN举例用）。



其实这就是CDN的原理，不过CDN一般都会引入CNAME机制，如果不用CNAME，那么我们这个域名解析为多个A记录的工作量将会变得无比巨大。

我们以七牛云为例，说明它的CNAME如何为CDN加速比如图片加速、视频加速服务的。

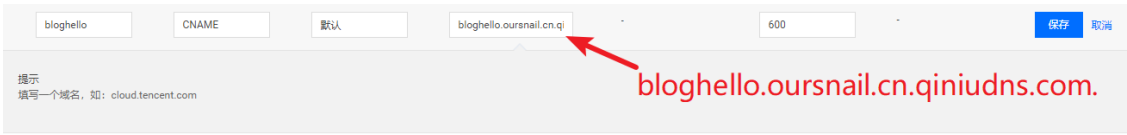
比如我这里的的笔记图片是上传到七牛云上的，如：

```
http://bloghello.oursnail.cn/avatar.png
```

我将图片上传后，七牛云如何为我实现加速的呢？七牛云也有一个对应的域名，这个域名需要用户在七牛云上创建，比如我这里创建为：

```
bloghello.oursnail.cn.qiniudns.com
```

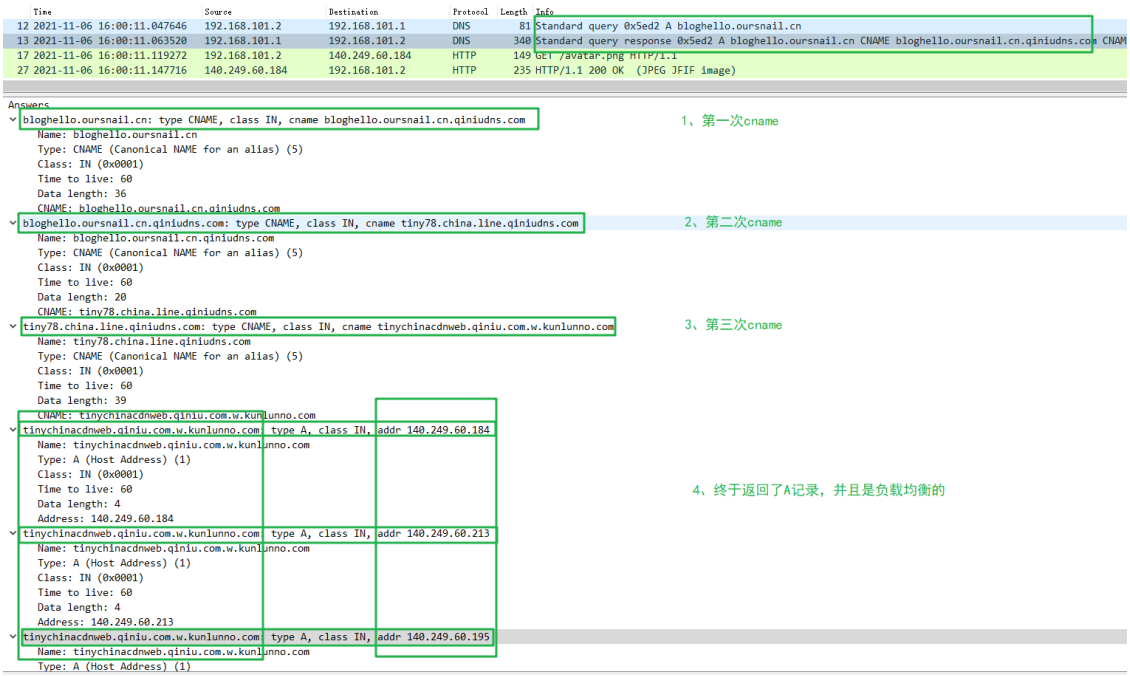
七牛云要求你必须拥有自己的域名作为图片域名使用，比如我这里使用的是 `bloghello.oursnail.cn` ，该三级域名是我在腾讯云服务器上配置的，这里就需要在云服务器上配置该域名的CNAME记录：



经过上面一波操作后，我访问图片的过程就变成了：

- **第一步：将自身域名 `bloghello.oursnail.cn` 解析为别名，即七牛云对应的域名 `bloghello.oursnail.cn.qiniudns.com`，后续如有其他别名则继续解析。**
- **第二步：七牛云内部根据负载均衡算法，将离用户最近的一个CDN节点IP返回回来，给用户提供服务，这个过程被称为CDN调度。**

我们来抓包看下以上过程：



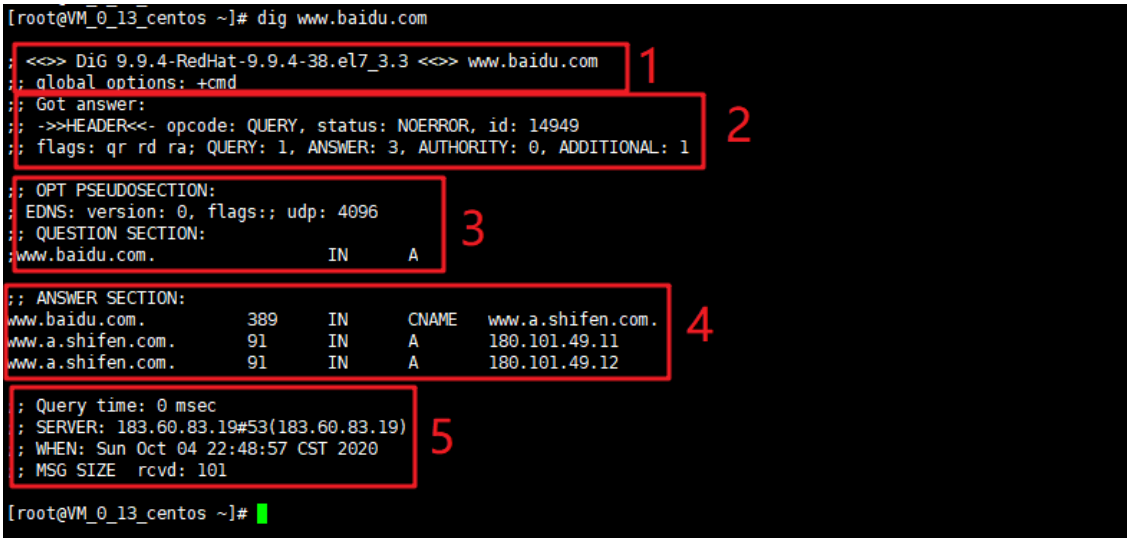
可以看到，经过了三次cname，才返回了A记录。这个过程就是一个CDN调度的过程，用最近的节点向用户提供服务，提高资源响应速度。

对于简单的应用，DNS解析为一个或多个IP地址，客户端通过多个IP地址进行简单的轮询实现简单的负载均衡即可；对于复杂的应用，尤其是跨地域跨运营商的大型应用，则需要更加复杂的全局负载均衡机制，因而需要专门的设备或者服务器来做这件事情，这就是全局负载均衡器（GSLB，Global Server Load Balance），并且GSLB往往不止一层，有可能是这样的形式：第一层GSLB根据用户本地DNS地址得知用户网络所属运营商，比如明确是移动还是电信还是联通后，第二层GSLB根据用户本

地DNS地址得知用户所属地区，比如江苏南京还是上海，从离用户位置比较近的区域中获得可提供服务的IP列表。

四、dig命令

dig命令可以更详细地查询DNS相关信息，我们还是以查询 `www.baidu.com` 为例：



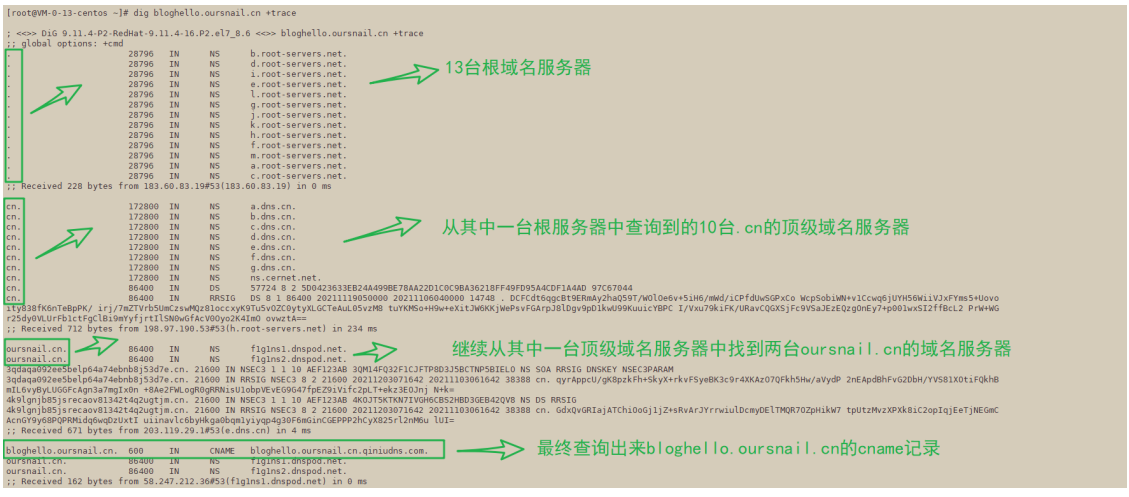
- 第一部分显示 dig 命令的版本和输入的参数。
- 第二部分显示服务返回的一些技术详情，比较重要的是 status。如果 status 的值为 NOERROR 则说明本次查询成功结束。
- 第三部分中的 "QUESTION SECTION" 显示我们要查询的域名。
- 第四部分的 "ANSWER SECTION" 是查询到的结果。
- 第五部分则是本次查询的一些统计信息，比如用了多长时间，查询了哪个 DNS 服务器，在什么时间进行的查询等等。

那么我们同理来看看 bloghello.oursnail.com 的 dig 结果：

```
[root@VM_0_13_centos ~]# dig bloghello.oursnail.cn

; <<>> DiG 9.9.4-RedHat-9.9.4-38.el7_3.3 <<>> bloghello.oursnail.cn
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59231
;; flags: qr rd ra; QUERY: 1, ANSWER: 19, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;bloghello.oursnail.cn.         IN      A

;; ANSWER SECTION:
bloghello.oursnail.cn.  10      IN      CNAME   bloghello.oursnail.cn.qiniudns.com.
bloghello.oursnail.cn.qiniudns.com. 300 IN CNAME tiny78.china.line.qiniudns.com.
tiny78.china.line.qiniudns.com. 164 IN  CNAME   tinychinacdnweb.qiniu.com.w.kunlunno.com.
tinychinacdnweb.qiniu.com.w.kunlunno.com. 60 IN A 114.80.187.103
tinychinacdnweb.qiniu.com.w.kunlunno.com. 60 IN A 114.80.187.104
tinychinacdnweb.qiniu.com.w.kunlunno.com. 60 IN A 114.80.187.67
tinychinacdnweb.qiniu.com.w.kunlunno.com. 60 IN A 114.80.187.68
tinychinacdnweb.qiniu.com.w.kunlunno.com. 60 IN A 114.80.187.69
tinychinacdnweb.qiniu.com.w.kunlunno.com. 60 IN A 114.80.187.71
tinychinacdnweb.qiniu.com.w.kunlunno.com. 60 IN A 114.80.187.72
tinychinacdnweb.qiniu.com.w.kunlunno.com. 60 IN A 114.80.187.74
tinychinacdnweb.qiniu.com.w.kunlunno.com. 60 IN A 114.80.187.75
tinychinacdnweb.qiniu.com.w.kunlunno.com. 60 IN A 114.80.187.76
tinychinacdnweb.qiniu.com.w.kunlunno.com. 60 IN A 114.80.187.77
tinychinacdnweb.qiniu.com.w.kunlunno.com. 60 IN A 114.80.187.84
tinychinacdnweb.qiniu.com.w.kunlunno.com. 60 IN A 114.80.187.90
tinychinacdnweb.qiniu.com.w.kunlunno.com. 60 IN A 114.80.187.98
tinychinacdnweb.qiniu.com.w.kunlunno.com. 60 IN A 114.80.187.99
tinychinacdnweb.qiniu.com.w.kunlunno.com. 60 IN A 114.80.187.102

;; Query time: 88 msec
;; SERVER: 183.60.83.19#53(183.60.83.19)
;; WHEN: Sun Oct 04 22:51:58 CST 2020
;; MSG SIZE  rcvd: 437

[root@VM_0_13_centos ~]#
```

可以完全复现以上对七牛云域名解析的抓包过程，体现了三次cname和多个A记录的情况。

我们在上面提到了迭代查询，我们可以通过在dig后面增加“+trace”参数强迫客户端采取迭代查询模式，请看：



从抓包的角度来看，显然迭代查询要复杂一点，不过迭代查询和递归查询的最终结果是一样的。

也可看出dig命令的强大，nslookup、dig等命令是平时工作中排查网络的两大瑞士军刀，十分好用，我们的视野需要开拓，不能仅仅知道ping和telnet就结束了。

好了，关于DNS，想分享的大概就是这些，涵盖的内容还是较多的，所以提及到DNS实际上应该有很多话题的，而不仅仅是一句简单的域名解析。这里涉及到的很多名词和工具都将伴随我们很久很久，我们有必要经常使用并加以实践。