DNS 原理入门

作者： 阮一峰

日期： 2016年6月16日

DNS 是互联网核心协议之一。不管是上网浏览，还是编程开发，都需要了解一点它的知识。

本文详细介绍DNS的原理，以及如何运用工具软件观察它的运作。我的目标是，读完此文后，你就能完全理解DNS。

一、DNS 是什么？

DNS （Domain Name System 的缩写）的作用非常简单，就是根据域名查出IP地址。你可以把它想象成一本巨大的电话本。

举例来说，如果你要访问域名 math.stackexchange.com ，首先要通过DNS查出它的IP地址是 151.101.129.69 。

如果你不清楚为什么一定要查出IP地址，才能进行网络通信，建议先阅读我写的《互联网协议入门》。

二、查询过程

虽然只需要返回一个IP地址，但是DNS的查询过程非常复杂，分成多个步骤。

工具软件 dig 可以显示整个查询过程。

```
$ dig math.stackexchange.com
```

上面的命令会输出六段信息。

```
; <<>> DiG 9.9.5-12.1-Debian <<>> math.stackexchange.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6328
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 4, ADDITIONAL: 4

;; QUESTION SECTION:
;math.stackexchange.com.                IN      A

;; ANSWER SECTION:
math.stackexchange.com. 600     IN      A       151.101.65.69
math.stackexchange.com. 600     IN      A       151.101.129.69
math.stackexchange.com. 600     IN      A       151.101.193.69
math.stackexchange.com. 600     IN      A       151.101.1.69

;; AUTHORITY SECTION:
stackexchange.com.      171469  IN      NS      ns-1832.awsdns-37.co.uk.
stackexchange.com.      171469  IN      NS      ns-1029.awsdns-00.org.
stackexchange.com.      171469  IN      NS      ns-463.awsdns-57.com.
stackexchange.com.      171469  IN      NS      ns-925.awsdns-51.net.

;; ADDITIONAL SECTION:
ns-463.awsdns-57.com.   171406  IN      A       205.251.193.207
ns-925.awsdns-51.net.   171393  IN      A       205.251.195.157
ns-1029.awsdns-00.org.  171469  IN      A       205.251.196.5
ns-1832.awsdns-37.co.uk. 171393 IN      A       205.251.199.40

;; Query time: 5 msec
;; SERVER: 192.168.1.253#53(192.168.1.253)
;; WHEN: Mon Jun 13 22:12:46 CST 2016
;; MSG SIZE  rcvd: 305
```

第一段是查询参数和统计。

```
; <<>> DiG 9.9.5-12.1-Debian <<>> math.stackexchange.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6328
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 4, ADDITIONAL: 4
```

第二段是查询内容。

```
;; QUESTION SECTION:
;math.stackexchange.com.                IN      A
```

上面结果表示，查询域名 math.stackexchange.com 的 A 记录， A 是address的缩写。

第三段是DNS服务器的答复。

```
;; ANSWER SECTION:
math.stackexchange.com. 600     IN      A       151.101.65.69
math.stackexchange.com. 600     IN      A       151.101.129.69
math.stackexchange.com. 600     IN      A       151.101.193.69
math.stackexchange.com. 600     IN      A       151.101.1.69
```

上面结果显示， math.stackexchange.com 有四个 A 记录，即四个IP地址。 600 是TTL值（Time to live 的缩写），表示缓存时间，即600秒之内不用重新查询。

第四段显示 stackexchange.com 的NS记录（Name Server的缩写），即哪些服务器负责管理 stackexchange.com 的DNS记录。

```
;; AUTHORITY SECTION:
stackexchange.com.      171469  IN      NS      ns-1832.awsdns-37.co.uk.
stackexchange.com.      171469  IN      NS      ns-1029.awsdns-00.org.
stackexchange.com.      171469  IN      NS      ns-463.awsdns-57.com.
stackexchange.com.      171469  IN      NS      ns-925.awsdns-51.net.
```

上面结果显示 stackexchange.com 共有四条NS记录，即四个域名服务器，向其中任一台查询就能知道 math.stackexchange.com 的IP地址是什么。

第五段是上面四个域名服务器的IP地址，这是随着前一段一起返回的。

```
;; ADDITIONAL SECTION:
ns-463.awsdns-57.com.   171406  IN      A       205.251.193.207
ns-925.awsdns-51.net.   171393  IN      A       205.251.195.157
ns-1029.awsdns-00.org.  171469  IN      A       205.251.196.5
ns-1832.awsdns-37.co.uk. 171393 IN      A       205.251.199.40
```

第六段是DNS服务器的一些传输信息。

```
;; Query time: 7 msec
;; SERVER: 192.168.1.253#53(192.168.1.253)
;; WHEN: Wed Jun 15 23:23:55 CST 2016
;; MSG SIZE  rcvd: 305
```

上面结果显示，本机的DNS服务器是 192.168.1.253 ，查询端口是53（DNS服务器的默认端口），以及回应长度是305字节。

如果不想看到这么多内容，可以使用 +short 参数。

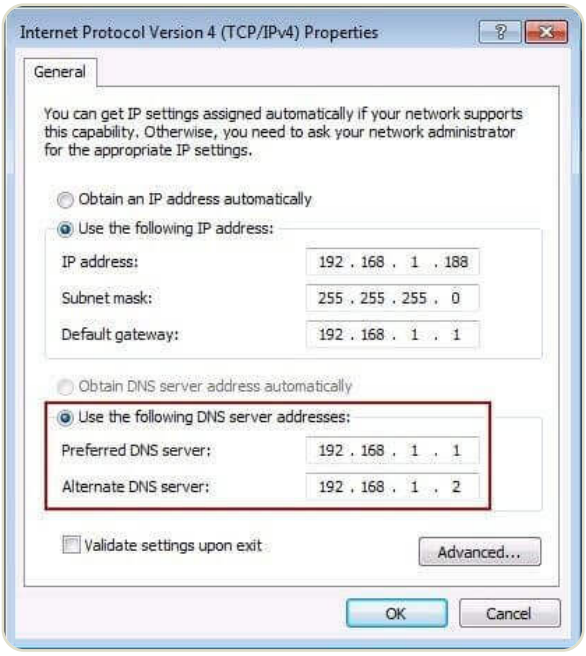
```
$ dig +short math.stackexchange.com

151.101.129.69
151.101.65.69
151.101.193.69
151.101.1.69
```

上面命令只返回 math.stackexchange.com 对应的4个IP地址（即 A 记录）。

三、DNS服务器

下面我们根据前面这个例子，一步步还原，本机到底怎么得到域名 math.stackexchange.com 的IP地址。

首先，本机一定要知道DNS服务器的IP地址，否则上不了网。通过DNS服务器，才能知道某个域名的IP地址到底是什么。



DNS服务器的IP地址，有可能是动态的，每次上网时由网关分配，这叫做DHCP机制；也有可能是事先指定的固定地址。Linux系统里面，DNS服务器的IP地址保存在 /etc/resolv.conf 文件。

上例的DNS服务器是 192.168.1.253 ，这是一个内网地址。有一些公网的DNS服务器，也可以使用，其中最有名的就是Google的 8.8.8.8 和Level 3的 4.2.2.2 。

本机只向自己的DNS服务器查询， dig 命令有一个 @ 参数，显示向其他DNS服务器查询的结果。

```
$ dig @4.2.2.2 math.stackexchange.com
```

上面命令指定向DNS服务器 4.2.2.2 查询。

四、域名的层级

DNS服务器怎么会知道每个域名的IP地址呢？答案是分级查询。

请仔细看前面的例子，每个域名的尾部都多了一个点。

```
;; QUESTION SECTION:
;math.stackexchange.com.                IN      A
```

比如，域名 math.stackexchange.com 显示为 math.stackexchange.com. 。这不是疏忽，而是所有域名的尾部，实际上都有一个根域名。

举例来说， www.example.com 真正的域名是 www.example.com.root ，简写为 www.example.com. 。因为，根域名 .root 对于所有域名都是一样的，所以平时是省略的。

根域名的下一级，叫做"顶级域名"（top-level domain，缩写为TLD），比如 .com 、 .net ；再下一级叫做"次级域名"（second-level domain，缩写为SLD），比如 www.example.com 里面的 .example ，这一级域名是用户可以注册的；再下一级是主机名（host），比如 www.example.com 里面的 www ，又称为"三级域名"，这是用户在自己的域里面为服务器分配的名称，是用户可以任意分配的。

总结一下，域名的层级结构如下。

```
主机名.次级域名.顶级域名.根域名

# 即

host.sld.tld.root
```

五、根域名服务器

DNS服务器根据域名的层级，进行分级查询。

需要明确的是，每一级域名都有自己的NS记录，NS记录指向该级域名的域名服务器。这些服务器知道下一级域名的各种记录。

所谓"分级查询"，就是从根域名开始，依次查询每一级域名的NS记录，直到查到最终的IP地址，过程大致如下。

1. 从"根域名服务器"查到"顶级域名服务器"的NS记录和A记录（IP地址）
2. 从"顶级域名服务器"查到"次级域名服务器"的NS记录和A记录（IP地址）
3. 从"次级域名服务器"查出"主机名"的IP地址

仔细看上面的过程，你可能发现了，没有提到DNS服务器怎么知道"根域名服务器"的IP地址。回答是"根域名服务器"的NS记录和IP地址一般是不会变化的，所以内置在DNS服务器里面。

下面是内置的根域名服务器IP地址的一个例子。

```
; formerly NS.INTERNIC.NET
;
.                        3600000  IN  NS    A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.      3600000      A     198.41.0.4
A.ROOT-SERVERS.NET.      3600000      AAAA  2001:503:BA3E::2:30
;
; formerly NS1.ISI.EDU
;
.                        3600000      NS    B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.      3600000      A     192.228.79.201
;
; formerly C.PSI.NET
;
.                        3600000      NS    C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET.      3600000      A     192.33.4.12
```

上面列表中，列出了根域名（ .root ）的三条NS记录 A.ROOT-SERVERS.NET 、 B.ROOT-SERVERS.NET 和 C.ROOT-SERVERS.NET ，以及它们的IP地址（即 A 记录） 198.41.0.4 、 192.228.79.201 、 192.33.4.12 。

另外，可以看到所有记录的TTL值是3600000秒，相当于1000小时。也就是说，每1000小时才查询一次根域名服务器的列表。

目前，世界上一共有十三组根域名服务器，从 A.ROOT-SERVERS.NET 一直到 M.ROOT-SERVERS.NET 。

六、分级查询的实例

dig 命令的 +trace 参数可以显示DNS的整个分级查询过程。

```
$ dig +trace math.stackexchange.com
```

上面命令的第一段列出根域名 . 的所有NS记录，即所有根域名服务器。

```
; <<>> DiG 9.9.5-12.1-Debian <<>> +trace math.stackexchange.com
;; global options: +cmd
.                       318820  IN      NS      b.root-servers.net.
.                       318820  IN      NS      h.root-servers.net.
.                       318820  IN      NS      e.root-servers.net.
.                       318820  IN      NS      l.root-servers.net.
.                       318820  IN      NS      g.root-servers.net.
.                       318820  IN      NS      k.root-servers.net.
.                       318820  IN      NS      j.root-servers.net.
.                       318820  IN      NS      c.root-servers.net.
.                       318820  IN      NS      m.root-servers.net.
.                       318820  IN      NS      i.root-servers.net.
.                       318820  IN      NS      a.root-servers.net.
.                       318820  IN      NS      d.root-servers.net.
.                       318820  IN      NS      f.root-servers.net.
```

根据内置的根域名服务器IP地址，DNS服务器向所有这些IP地址发出查询请求，询问 `math.stackexchange.com` 的顶级域名服务器 `com.` 的NS记录。最先回复的根域名服务器将被缓存，以后只向这台服务器发请求。

接着是第二段。

```
com.                    172800  IN      NS      a.gtld-servers.net.
com.                    172800  IN      NS      b.gtld-servers.net.
com.                    172800  IN      NS      c.gtld-servers.net.
com.                    172800  IN      NS      d.gtld-servers.net.
com.                    172800  IN      NS      e.gtld-servers.net.
com.                    172800  IN      NS      f.gtld-servers.net.
com.                    172800  IN      NS      g.gtld-servers.net.
com.                    172800  IN      NS      h.gtld-servers.net.
com.                    172800  IN      NS      i.gtld-servers.net.
com.                    172800  IN      NS      j.gtld-servers.net.
com.                    172800  IN      NS      k.gtld-servers.net.
com.                    172800  IN      NS      l.gtld-servers.net.
com.                    172800  IN      NS      m.gtld-servers.net.
```

上面结果显示 `.com` 域名的13条NS记录，同时返回的还有每一条记录对应的IP地址。

然后，DNS服务器向这些顶级域名服务器发出查询请求，询问 `math.stackexchange.com` 的次级域名 `stackexchange.com` 的NS记录。

```
stackexchange.com.      172800  IN      NS      ns-463.awsdns-57.com.
stackexchange.com.      172800  IN      NS      ns-925.awsdns-51.net.
stackexchange.com.      172800  IN      NS      ns-1029.awsdns-00.org.
stackexchange.com.      172800  IN      NS      ns-1832.awsdns-37.co.uk.
```

上面结果显示 `stackexchange.com` 有四条NS记录，同时返回的还有每一条NS记录对应的IP地址。

然后，DNS服务器向上面这四台NS服务器查询 `math.stackexchange.com` 的主机名。

```
math.stackexchange.com. 300     IN      A       151.101.65.69
math.stackexchange.com. 300     IN      A       151.101.193.69
math.stackexchange.com. 300     IN      A       151.101.129.69
math.stackexchange.com. 300     IN      A       151.101.1.69
stackexchange.com.      172800  IN      NS      ns-1029.awsdns-00.org.
stackexchange.com.      172800  IN      NS      ns-1832.awsdns-37.co.uk.
stackexchange.com.      172800  IN      NS      ns-463.awsdns-57.com.
stackexchange.com.      172800  IN      NS      ns-925.awsdns-51.net.
;; Received 252 bytes from 205.251.193.207#53(ns-463.awsdns-57.com) in 226 ms
```

上面结果显示， math.stackexchange.com 有4条 A 记录，即这四个IP地址都可以访问到网站。并且还显示，最先返回结果的NS服务器是 ns-463.awsdns-57.com ，IP地址为 205.251.193.207 。

七、NS 记录的查询

dig 命令可以单独查看每一级域名的NS记录。

```
$ dig ns com
$ dig ns stackexchange.com
```

+short 参数可以显示简化的结果。

```
$ dig +short ns com
$ dig +short ns stackexchange.com
```

八、DNS的记录类型

域名与IP之间的对应关系，称为"记录"（record）。根据使用场景，"记录"可以分成不同的类型（type），前面已经看到了有 A 记录和 NS 记录。

常见的DNS记录类型如下。

（1） A：地址记录（Address），返回域名指向的IP地址。

（2） NS：域名服务器记录（Name Server），返回保存下一级域名信息的服务器地址。该记录只能设置为域名，不能设置为IP地址。

（3） MX：邮件记录（Mail eXchange），返回接收电子邮件的服务器地址。

（4） CNAME：规范名称记录（Canonical Name），返回另一个域名，即当前查询的域名是另一个域名的跳转，详见下文。

（5） PTR：逆向查询记录（Pointer Record），只用于从IP地址查询域名，详见下文。

一般来说，为了服务的安全可靠，至少应该有两条 NS 记录，而 A 记录和 MX 记录也可以有多条，这样就提供了服务的冗余性，防止出现单点失败。

CNAME 记录主要用于域名的内部跳转，为服务器配置提供灵活性，用户感知不到。举例来说， facebook.github.io 这个域名就是一个 CNAME 记录。

```
$ dig facebook.github.io

...

;; ANSWER SECTION:
```

```
facebook.github.io. 3370    IN  CNAME   github.map.fastly.net.
github.map.fastly.net.  600 IN  A   103.245.222.133
```

上面结果显示， facebook.github.io 的CNAME记录指向 github.map.fastly.net 。也就是说，用户查询 facebook.github.io 的时候，实际上返回的是 github.map.fastly.net 的IP地址。这样的好处是，变更服务器IP地址的时候，只要修改 github.map.fastly.net 这个域名就可以了，用户的 facebook.github.io 域名不用修改。

由于 CNAME 记录就是一个替换，所以域名一旦设置 CNAME 记录以后，就不能再设置其他记录了（比如 A 记录和 MX 记录），这是为了防止产生冲突。举例来说， foo.com 指向 bar.com ，而两个域名各有自己的 MX 记录，如果两者不一致，就会产生问题。由于顶级域名通常要设置 MX 记录，所以一般不允许用户对顶级域名设置 CNAME 记录。

PTR 记录用于从IP地址反查域名。 dig 命令的 -x 参数用于查询 PTR 记录。

```
$ dig -x 192.30.252.153

...

;; ANSWER SECTION:
153.252.30.192.in-addr.arpa. 3600 IN    PTR pages.github.com.
```

上面结果显示， 192.30.252.153 这台服务器的域名是 pages.github.com 。

逆向查询的一个应用，是可以防止垃圾邮件，即验证发送邮件的IP地址，是否真的有它所声称的域名。

dig 命令可以查看指定的记录类型。

```
$ dig a github.com
$ dig ns github.com
$ dig mx github.com
```

九、其他DNS工具

除了 dig ，还有一些其他小工具也可以使用。

（1）host 命令

host 命令可以看作 dig 命令的简化版本，返回当前请求域名的各种记录。

```
$ host github.com

github.com has address 192.30.252.121
github.com mail is handled by 5 ALT2.ASPMX.L.GOOGLE.COM.
github.com mail is handled by 10 ALT4.ASPMX.L.GOOGLE.COM.
github.com mail is handled by 10 ALT3.ASPMX.L.GOOGLE.COM.
github.com mail is handled by 5 ALT1.ASPMX.L.GOOGLE.COM.
github.com mail is handled by 1 ASPMX.L.GOOGLE.COM.

$ host facebook.github.com

facebook.github.com is an alias for github.map.fastly.net.
github.map.fastly.net has address 103.245.222.133
```

host 命令也可以用于逆向查询，即从IP地址查询域名，等同于 dig -x <ip> 。

```
$ host 192.30.252.153

153.252.30.192.in-addr.arpa domain name pointer pages.github.com.
```

（2）nslookup 命令

nslookup 命令用于互动式地查询域名记录。

```
$ nslookup

> facebook.github.io
Server:     192.168.1.253
Address:    192.168.1.253#53

Non-authoritative answer:
facebook.github.io  canonical name = github.map.fastly.net.
Name:   github.map.fastly.net
Address: 103.245.222.133

>
```

（3）whois 命令

whois 命令用来查看域名的注册情况。

```
$ whois github.com
```

十、参考链接

- DNS: The Good Parts, by Pete Keen
- DNS 101, by Mark McDonnell

（完）

文档信息

- 版权声明：自由转载-非商用-非衍生-保持署名（创意共享3.0许可证）
- 发表日期： 2016年6月16日
- 更多内容： 档案 » 理解计算机
- 博客文集：《寻找思想之路》，《未来世界的幸存者》
- 社交媒体： twitter， weibo
- Feed订阅：