

GNU/Linux DNS



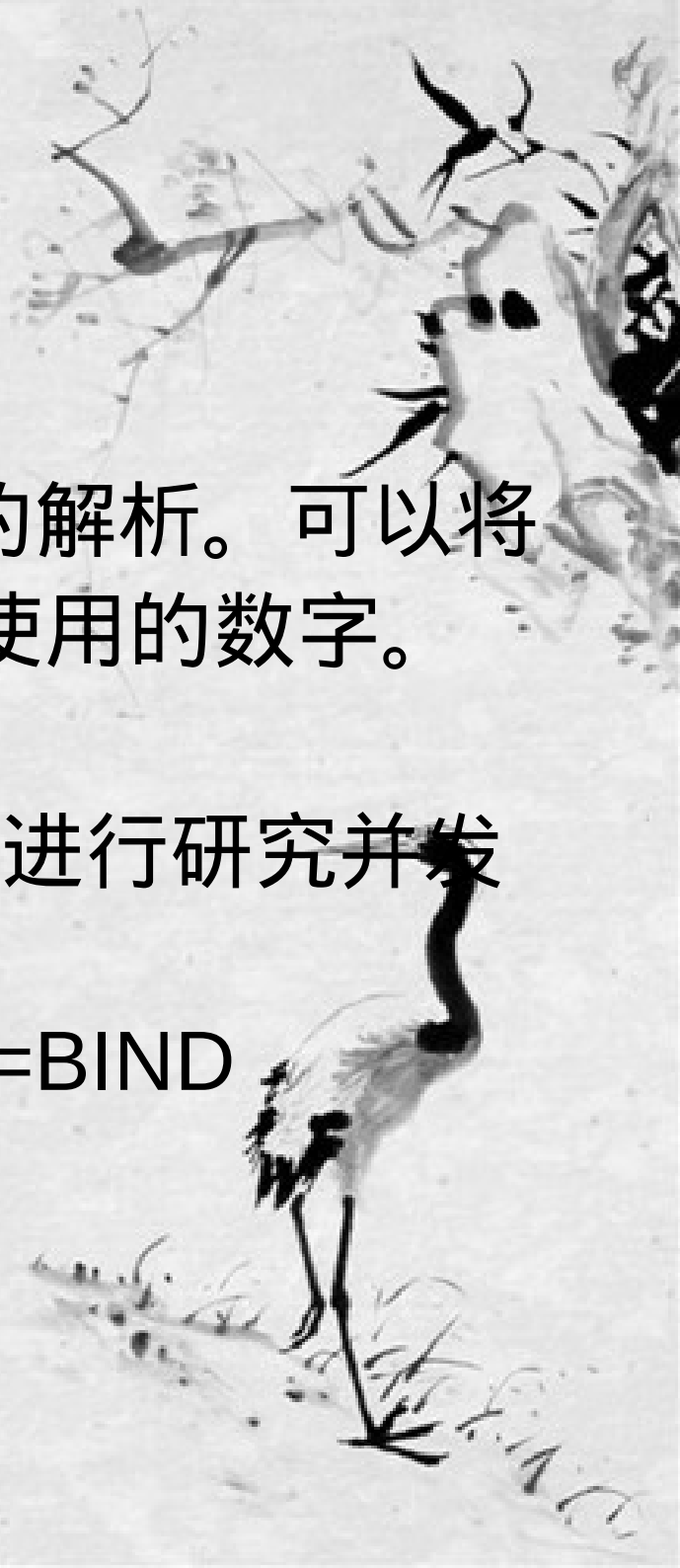
GNU/Linux

DNS(Domain Name System)

DNS 是完成主机名称到 IP 之间的解析。可以将计算机无法理解的文字转换为可以使用的数字。

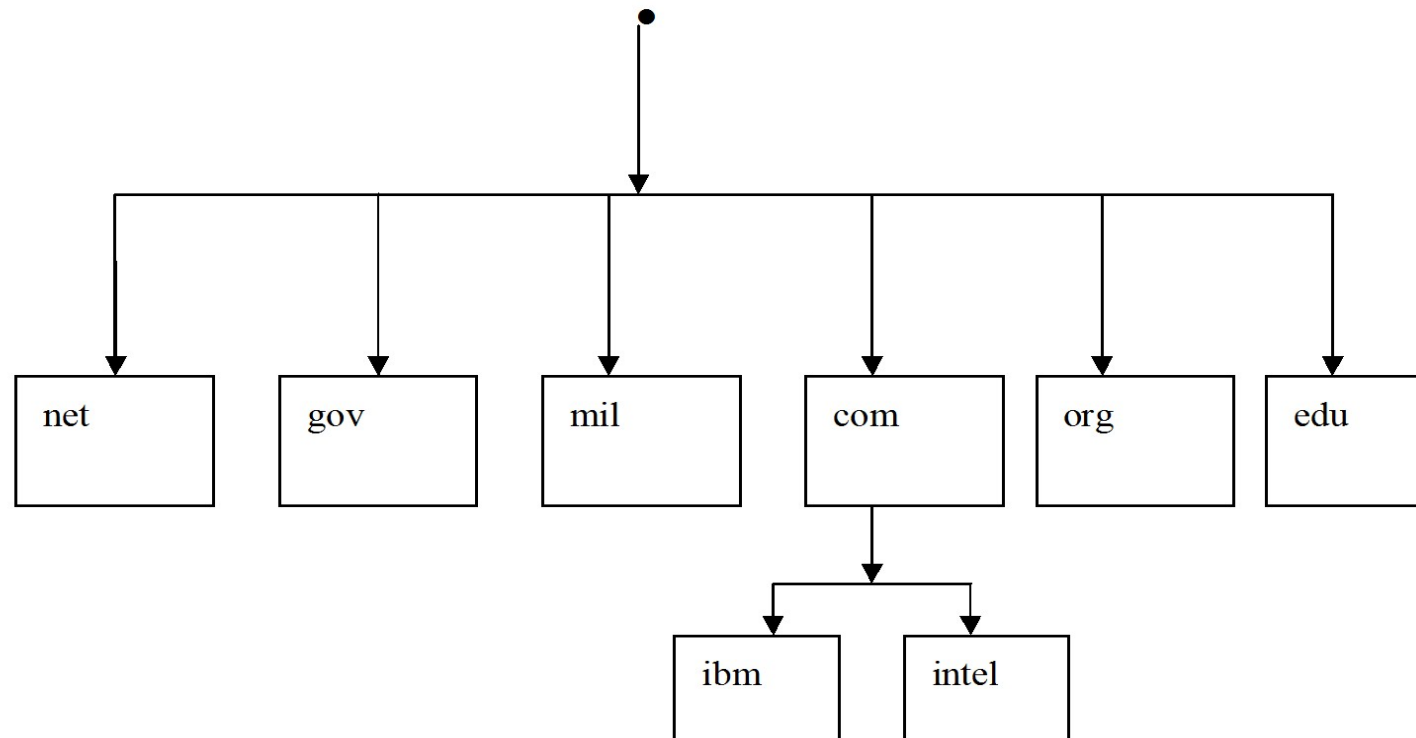
DNS 是由伯克利 (Berkeley) 大学进行研究并发布的。整套程序的名字为：

Berkeley Internet Name Domain=BIND



GNU/Linux

DNS 空间结构



GNU/Linux

DNS 术语

1. Root Domain(根域)

在 DNS 域名空间中，根域只有一个，它没有上级域，以圆点来 “.” 表示。全世界的 IP 地址与 DNS 域名空间都是有违于美国的 InterNIC （ Internet Network Information Center ） 负责管理或进行授权管理的。目前全世界有 13 台半根域服务器。

这些根域服务器中并没有保存全世界全部 Internet 网址，其中只保存顶级域的 “ DNS 服务器—— IP

GNU/Linux

DNS 术语

2. top-level domain(TLD, 顶级域)

在根域之下的第一级域便是顶级域，它以根域为上级域，其数目有限且不能轻易改变。

顶级域是由 InterNIC 统一管理的。在 FQDN 中，各级域之间都以原点 “.” 分隔，顶级域位于最右边。如 :www.gov.cn “.cn” 就是顶级域。顶级域一般分为地理域与机构域。

GNU/Linux

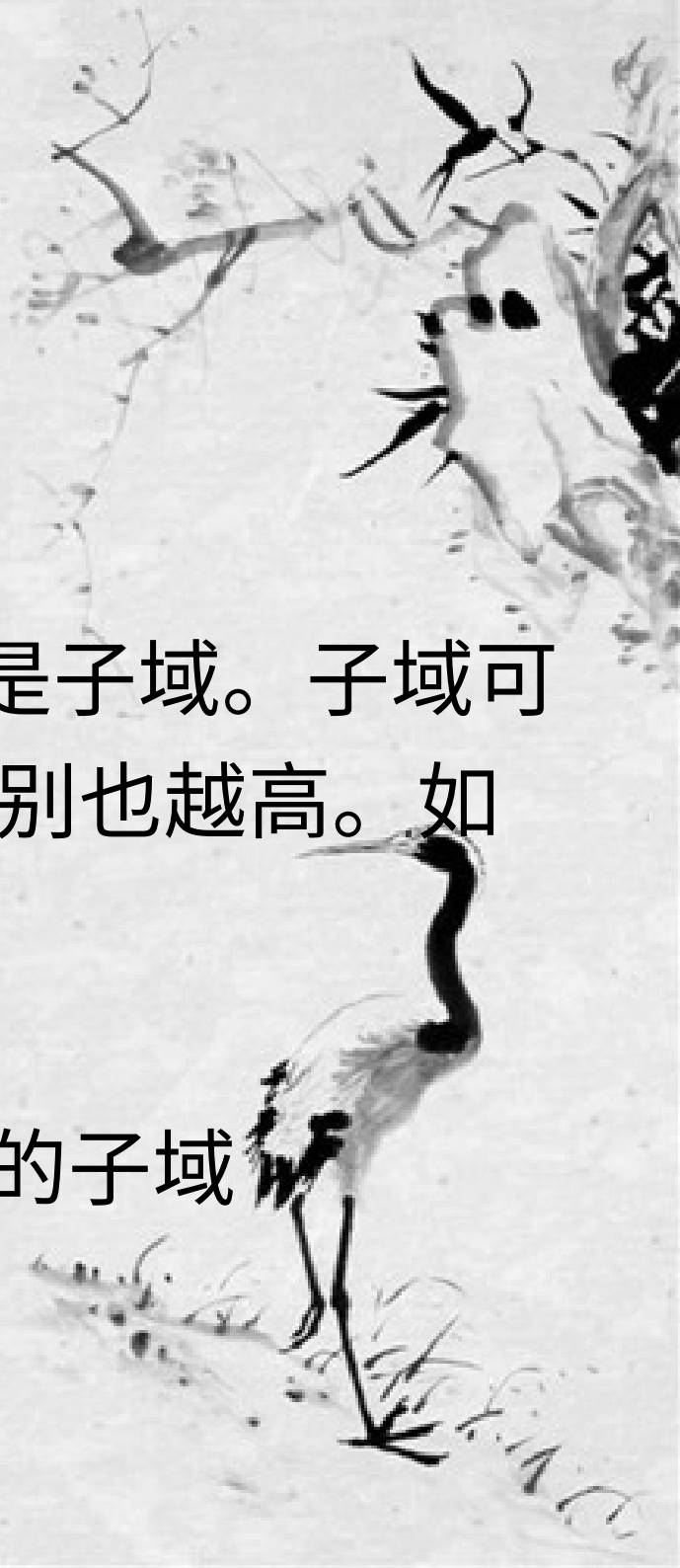
DNS 术语

3. subdomain(子域)

除了根域和顶域之外，其它域都是子域。子域可以分为很多层。按照离 TLD 越近级别也越高。如二级子域、三级子域等

如

www.gov.cn 中的 gov 即为 .cn 的子域

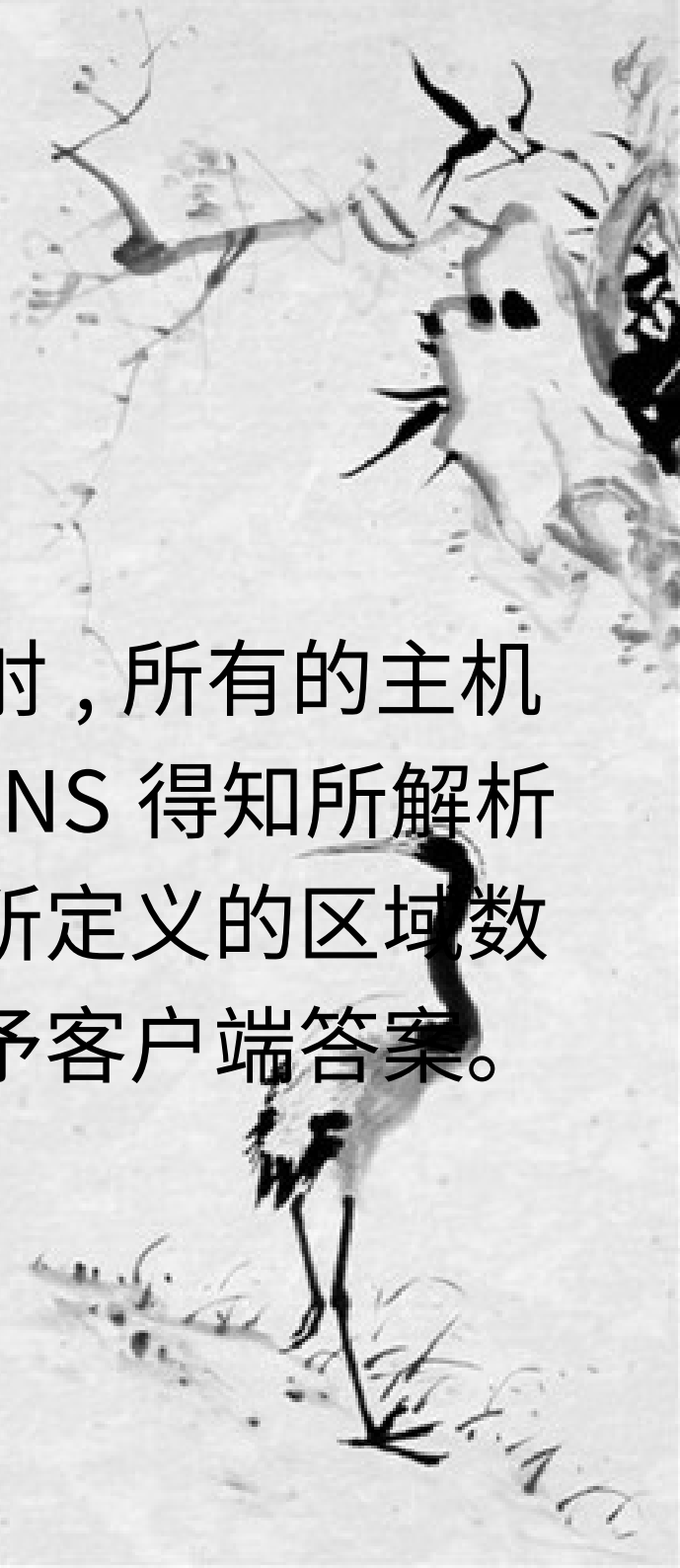


GNU/Linux

DNS 术语

4. zone(区域)

DNS 是直接解析 FQDN \leftrightarrow IP 映射，所有的主机或 IP 都在一个区域中声明。便于 DNS 得知所解析的内容归属于哪个区域，这样通过所定义的区域数据库文件来找到所对应的条目，给予客户端答案。



GNU/Linux

DNS 术语

5. 正解区域

在 DNS 中由 FQDN 解析为 IP 的功能区域被称为正解区域

6. 反解区域

在 DNS 中由 IP 解析为 FQDN 的功能区域被称为反解区域



GNU/Linux

DNS 术语

7. DNS 服务类型

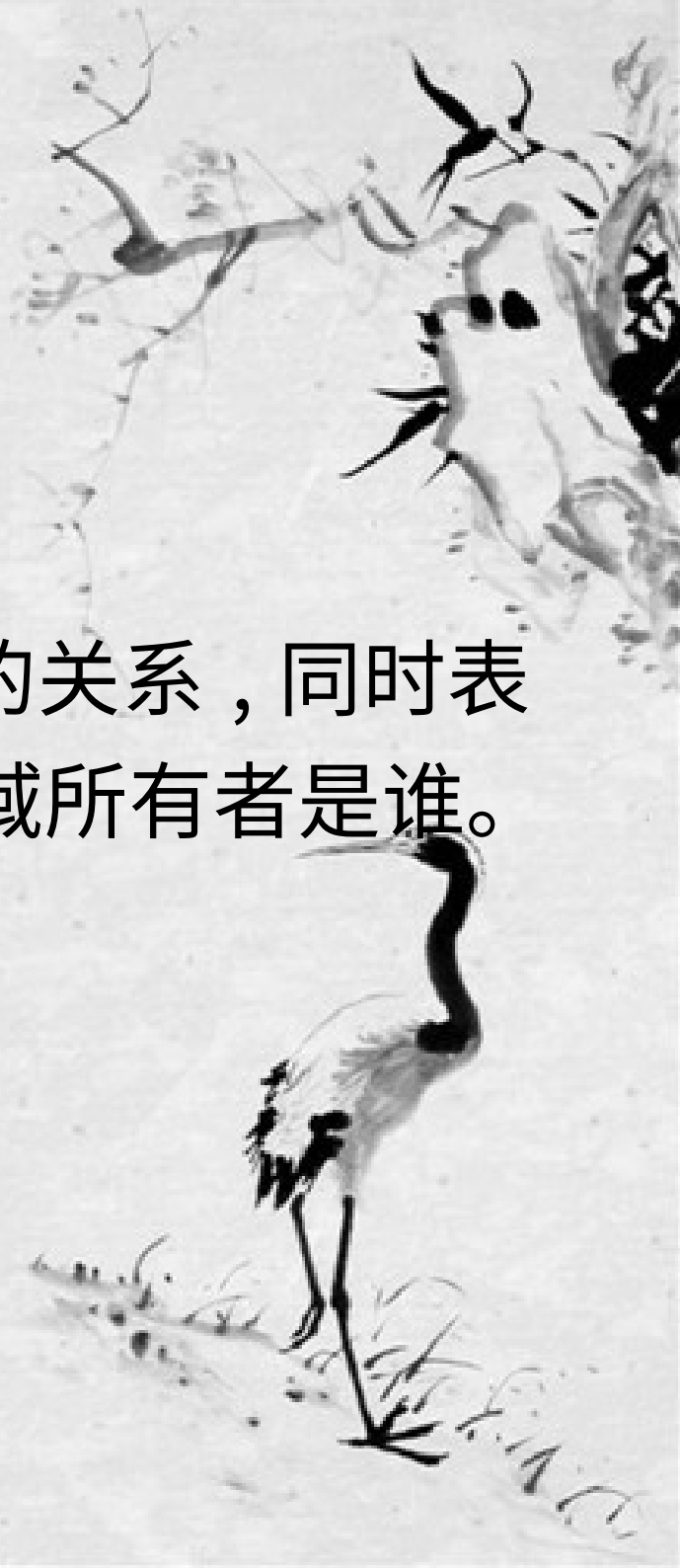
DNS 服务类型	服务类型	说明
hint	线索服务器	指定根服务的问题，通过 hint 来联系到根并刷新根服务器列表
master	主服务器	主域服务器，能够提供授权解析服务
slave	辅服务器	主域的冗余，每个主域可以有一个到多个 slave 服务器。slave 的数据直接通过 master 服务器进行复制，更新。并通过 master 服务器的定义，来控制 slave server 的版本更新时间及废弃时间等
forward	转发服务器	将本地网络中所需要解析的请求转发至指定 DNS 服务器
cache	缓存服务器	将所得到的解析信息存放在本地缓存中，以便于下次客户端直接读取，加快解析速度

GNU/Linux

DNS 术语

8. SOA(start of Authority)

SOA 记录表明 DNS 服务器之间的关系，同时表明所声明的对此具有修改权利的区域所有者是谁。



GNU/Linux

DNS 术语

9. NS(Name Server)

表示那台主机对此区域拥有解析权力。 NS 即权威 DNS 服务器。

对于辅 DNS 来说 ,NS 及主 DNS.

对于 DNS Client 而言 ,所指定的 DNS Server 即是它的权威服务器



GNU/Linux

DNS 术语

10. IN(class 字段, IN 所对应的为 Internet)

11. TTL(Time-To-Live): DNS 服务器缓存时间, 单位: 秒

12. A 记录: 正解记录。即 FQDN->IP

13. PTR 记录: 反解记录。即 IP->FQDN(IPv4 和 v6)



GNU/Linux

DNS 术语

14.AAAA 记录 :FQDN->IPv6 记录

15.MX 记录 : 邮件服务器 FQDN. 邮件服务可以配置多个, 以优先级控制。数字越小越优先

16.domain: 声明本地 DNS 为指定域的主域服务器

17.search: 声明本地 DNS/Client 为指定域的成员

GNU/Linux

DNS 术语

18. 区域命名：区域命名必须符合 FQDN 规范，且字符最大值不得超过 255 个字符。区域命名必须唯一。

19. 所有者：资源所有者。一般来说为 DNS 服务器的 FQDN

20: 指定的类型数据 :A/PTR/SOA/MX 所对应的值

GNU/Linux

DNS 术语

21.RNAME:DNS 区域负责人的邮件地址

22.CNAME:FQDN 的别名

23. 序列号：一个 32bit 的值，从 1-4294967295
如果值到最大时将重新轮换。此值主要定义了主
DNS 的修改次数。便于辅 DNS 用以对比、更新。


GNU/Linux



DNS 术语

24. 刷新时间：定义辅服务器每个多少时间连接主服务器检查更新情况

25. 重试时间：如辅服务器没有成功连接到主服务器，则每经过多少时间再次连接主服务器



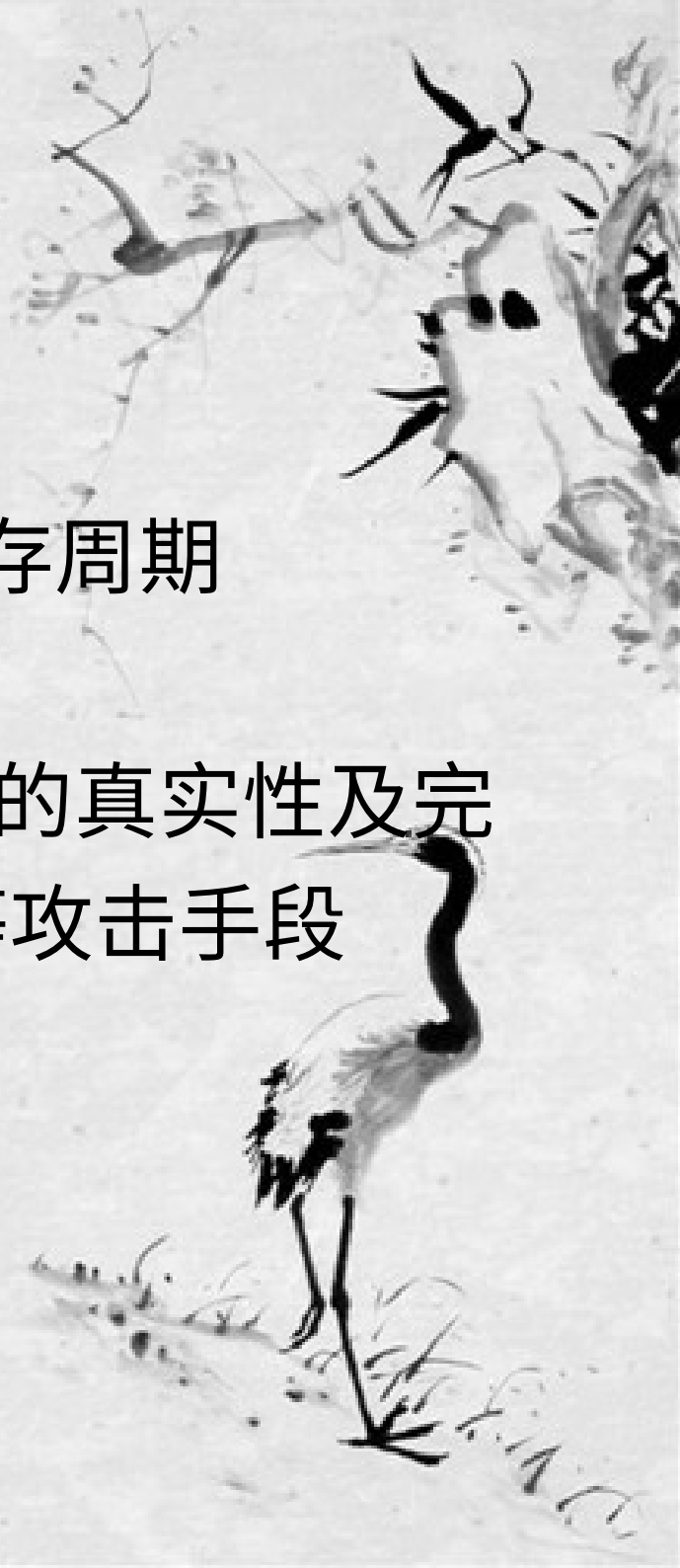
26. 废弃时间：在经过指定时间，辅服务器仍然无法连接到主服务器，则不在提供 DNS 解析服务

GNU/Linux

DNS 术语

27. 最小时间：指定辅 DNS 缓存生存周期

28.DNSSEC: 用于 DNS 之间的数据的真实性及完整性的效验。可以避免 DNS 劫持等攻击手段



GNU/Linux

FQDN 解析方式

1. 静态 :/etc/hosts

优点：

- 1) 如此文件中有相关的记录则优先 DNS 被直接解析相应的 IP
- 2) 速度快



GNU/Linux

FQDN 解析方式

1. 静态 :/etc/hosts

缺点：

1) 维护起来极度麻烦



GNU/Linux

FQDN 解析方式

2. 动态

使用 DNS 服务器完成解析。客户端不必存储大量的解析条目，仅通过 DNS 就可完成解析请求。



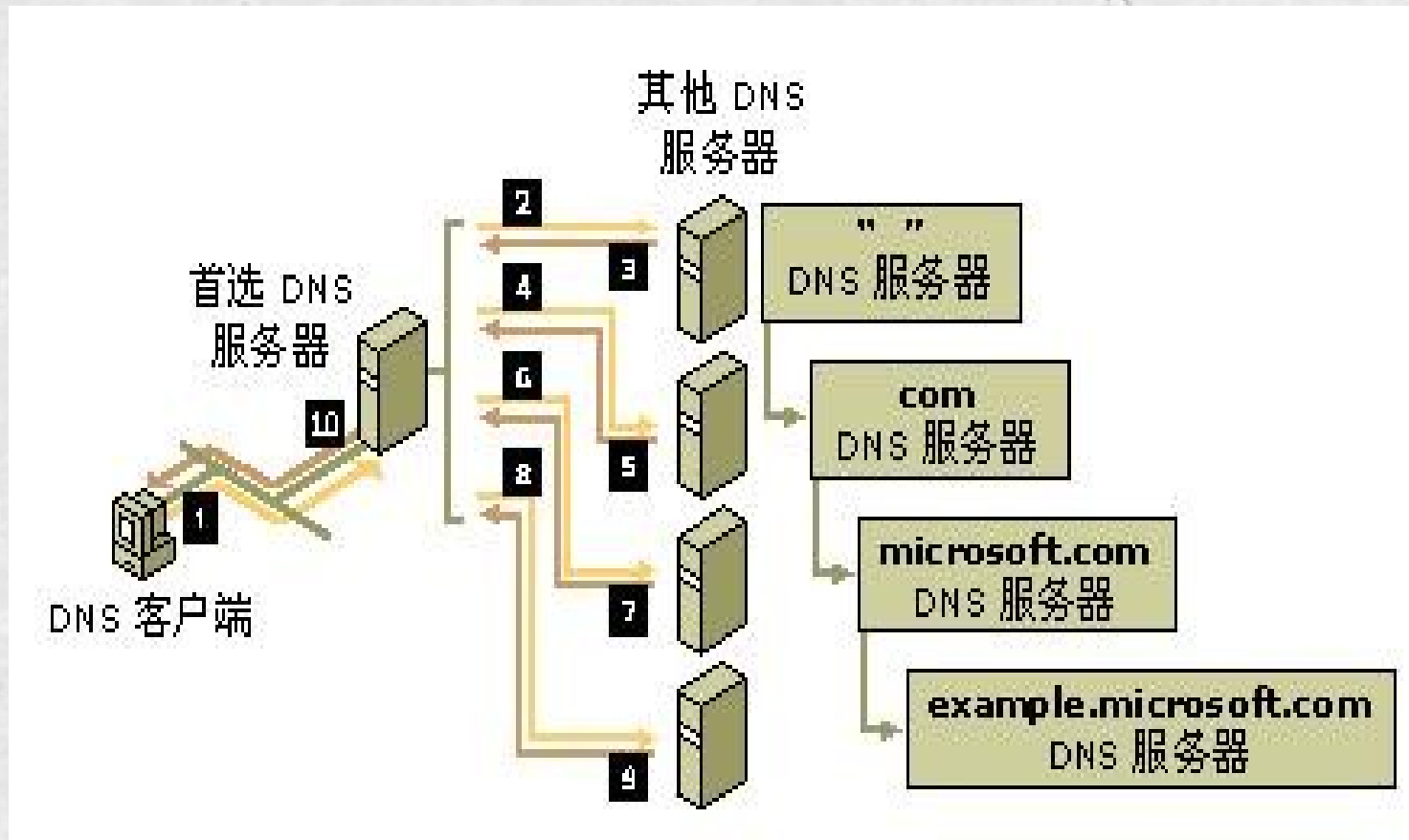
GNU/Linux

DNS 查询 FQDN 条目方式

查询模式	
迭代查询	DNS 服务器返回它能够提供的最佳答案， 否则提供一个指针，通知客户端向其它 dns 查询
递归查询 通常	DNS 服务器提供一个最终的结果。 它代表客户机对其它服务器进行独立的迭代查询

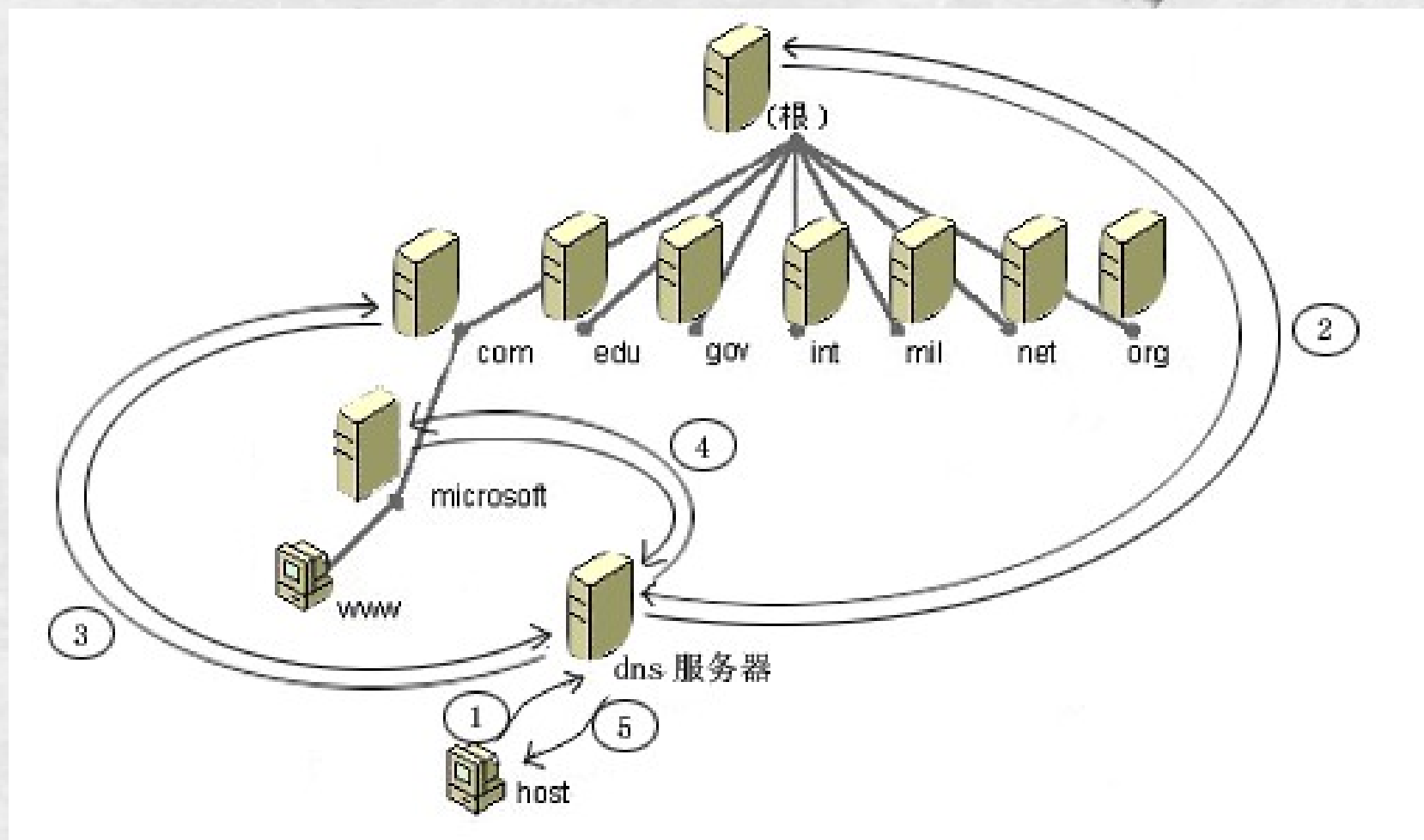
GNU/Linux

迭代查询



GNU/Linux

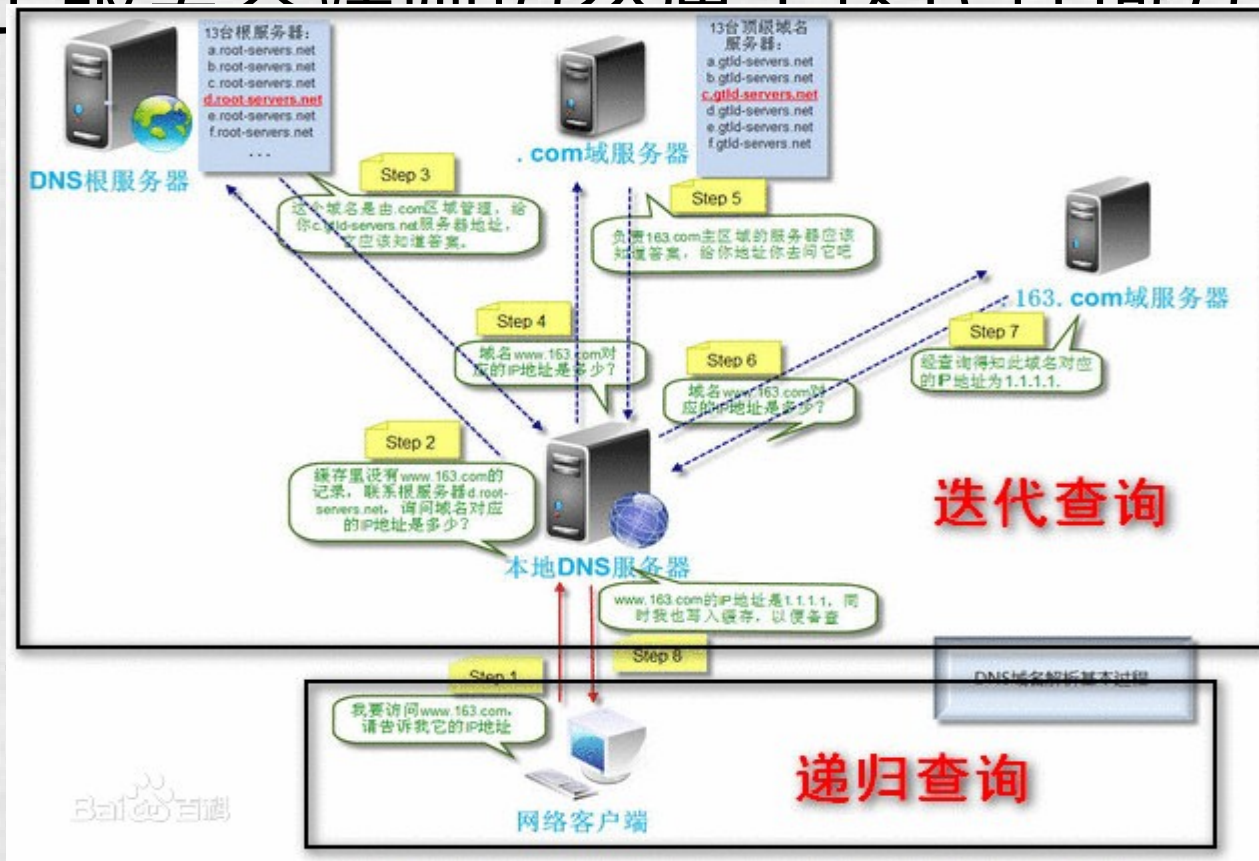
递归查询



GNU/Linux

迭代与递归查询

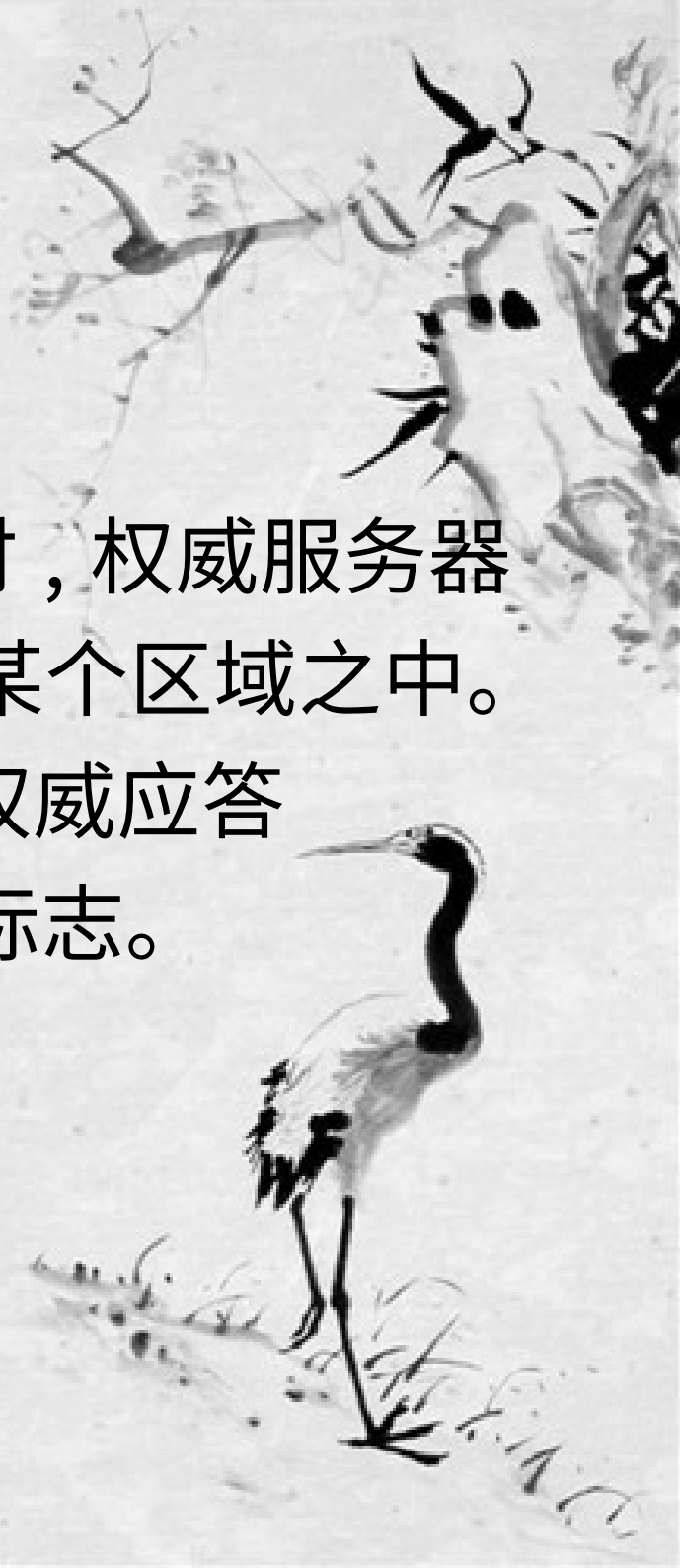
迭代与递归的主要动作是客户端查询行为而定性的。对于服务器端则仍然属于迭代查询方法



GNU/Linux

本地授权数据

查询到达一个权威 DNS 服务器时，权威服务器首先确定是否被查询的信息驻留在某个区域之中。如果存在，权威服务器将响应一个权威应答(AA)。所有提供的数据都将有 AA 标志。



GNU/Linux

本地缓存未授权数据

如果权威 DNS 服务器没有权威区域的记录，则通过自己最近获得的缓存来查询应答。但此资源不属于 AA 应答。



GNU/Linux

递归获取未授权数据

如果权威 DNS 服务器在缓存中亦为查找出条目，将完成一个迭代查询的过程，通过迭代流程完成条目的查询，并最终将条目信心缓存至服务器本地，而后将结果递归给客户端



GNU/Linux

DNS 所使用的协议及端口

tcp:53

udp:53



GNU/Linux

配置主 DNS 服务器

1. 安装软件包

```
#yum install bind
```

2. 打开配置文件

```
#vim /etc/named.conf
```



GNU/Linux

配置主 DNS 服务器

3. 了解 named.conf

4. 配置 named.conf 正解区域

1) 修改成最小配置



GNU/Linux

配置主 DNS 服务器

4. 配置 named.conf 正解区域

2) 增加区域

view “internal” { ←view 命名区域，限制允许的网络段 /IP 访问
match-clients {
 localhost;
 192.168.11.0/24;
};



GNU/Linux

配置主 DNS 服务器

4. 配置 named.conf 正解区域

2) 增加区域

```
zone “.” IN {  
    type hint;  
    file “named.ca” ;  
};
```



GNU/Linux

配置主 DNS 服务器

4. 配置 named.conf 正解

2) 增加区域

```
zone "niliu.edu" IN { ← 增加 niliu.edu 区域  
    type master; ← 服务器类型为 master  
    file "niliu.db"; ← 区域数据库文件  
};  
}; ← 结束 view 模式
```



GNU/Linux

配置主 DNS 服务器

4. 配置 named.conf 正解区域

3) 创建 niliu.db

```
#cd /var/named
```

```
#vim niliu.db
```

```
$TTL 1D
```

```
@      IN      SOA      rh7s1.niliu.edu. root.rh7s1.niliu.edu. (  
                                0      ;serial  
                                1D     ;refresh  
                                1H     ;retry  
                                1W     ;expire
```



GNU/Linux

配置主 DNS 服务器

4. 配置 named.conf 正解区域

3) 创建 niliu.db

```
@      IN      NSrh7s1.niliu.edu.
```

```
rh7s1 IN A 192.168.10.1
```

```
rh7s2 IN A 192.168.10.2
```

```
rh7s3 IN A 192.168.10.3
```

保存退出



GNU/Linux

配置主 DNS 服务器

4. 配置 named.conf 正解区域

4) 修改数据库属主 / 属组

```
#chown named.named niliu.db
```

5) 客户端配置

```
#vim /etc/resolv.conf
```

```
search niliu.edu
```

```
nameserver dns_server_ip
```

```
#dig FODN
```



GNU/Linux

DNS 客户端解析工具

命令 :nslookup

功能 : 查询 DNS 解析

语法格式 :nslookup [选项] [Server]



GNU/Linux

Nslookup:

正解

```
#nslookup
```

```
>www.niliu.edu
```

或

```
#nslookup www.niliu.edu
```



GNU/Linux

Nslookup:

反解

```
#nslookup
```

```
>192.168.1.123
```

或

```
#nslookup 192.168.1.123
```



GNU/Linux

Nslookup:

查询指定类型

```
#nslookup
```

```
>set type=mx
```

```
>niliu.edu
```

或

```
#nslookup -q=mx niliu.edu
```



GNU/Linux

命令 :dig

功能 : 查询 DNS 解析

语法格式 :dig [选项] [server]



GNU/Linux

查询 FQDN->IP

```
#dig www.niliu.edu
```

查询 IP->FQDN

```
#dig -x 192.168.1.111
```

查看解析过程

```
#dig www.niliu.edu +trace
```



GNU/Linux

查询指定 zone 的资源类型

```
#dig niliu.edu mx
```

```
#dig niliu.edu ns
```

```
#dig niliu.edu soa
```



GNU/Linux

命令 :host

功能 : 查询 DNS 解析

语法格式 :host [选项] [server]



GNU/Linux

查询 FQDN->IP

```
#host www.niliu.edu
```

查询 IP->FQDN

```
#host 192.168.1.111
```

查看指定资源

```
#host -t mx niliu.edu
```



GNU/Linux

命令 :gethostip

所需包 :syslinux

功能 : 查看 FQDN 与 IP 的映射

语法格式 :gethostip [选项] [hostname/IP]



GNU/Linux

查看 www.niliu.edu 的 IP

#gethostip www.niliu.edu

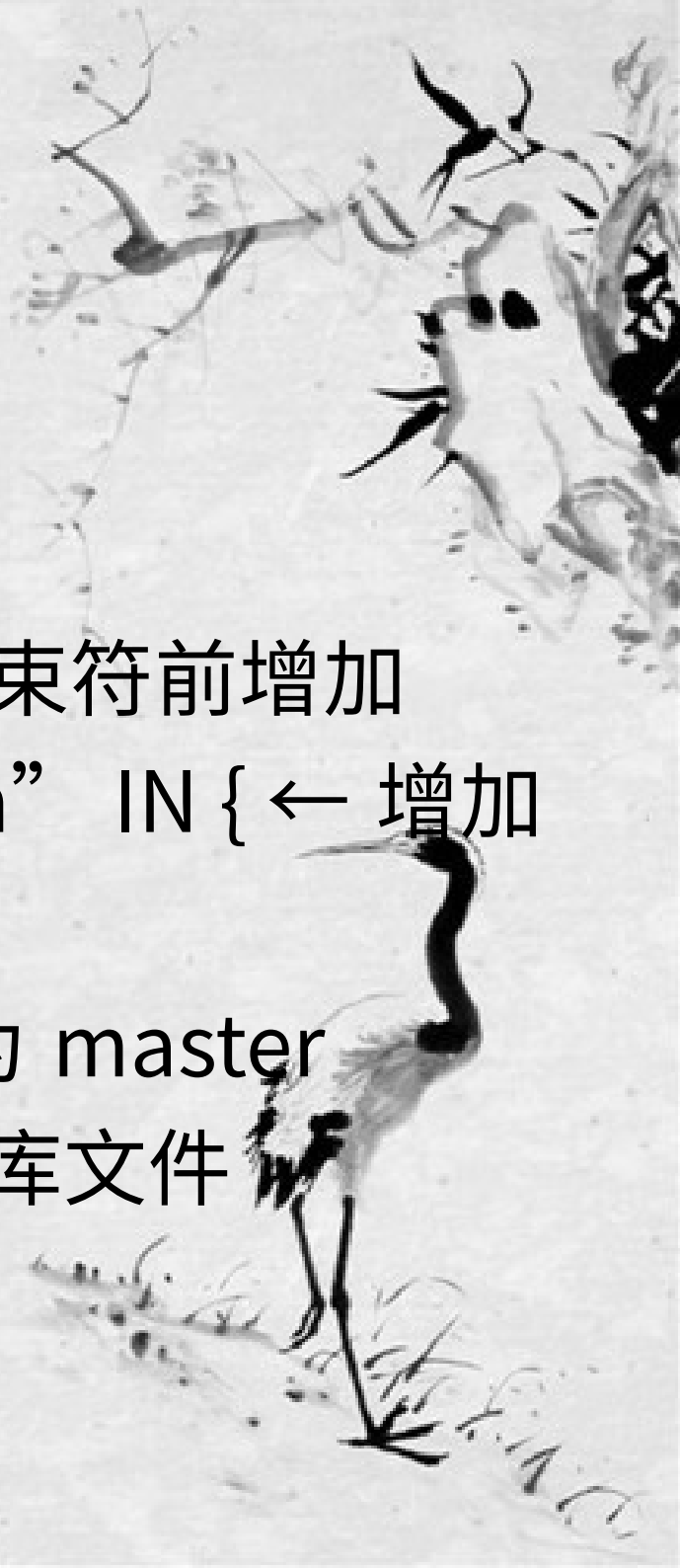


GNU/Linux

配置主 DNS 服务器

5. 配置 named.conf 反解区域

- 1) 在新建正解区域后面 ,view 结束符前增加
zone “10.168.192.in-addr.arpa” IN { ← 增加
niliu.edu 区域
type master; ← 服务器类型为 master
file “db.niliu” ; ← 区域数据库文件
};
}; ←view 结束符



GNU/Linux

配置主 DNS 服务器

5. 配置 named.conf 反解区域

2) 创建 db.niliu

```
#cd /var/named
```

```
#vim db.niliu
```

```
$TTL 1D
```

```
@      IN      SOA      rh7s1.niliu.edu. root.rh7s1.niliu.edu. (  
                                0      ;serial  
                                1D     ;refresh  
                                1H     ;retry  
                                1W     ;expire
```



GNU/Linux

配置主 DNS 服务器

5. 配置 named.conf 反解区域

3) 创建 niliu.db

@ IN NSrh7s1.niliu.edu.

1 IN PTR rh7s1.niliu.edu.

2 IN PTR rh7s2.niliu.edu.

3 IN PTR rh7s3.niliu.edu.

保存退出



GNU/Linux

配置主 DNS 服务器

5. 配置 named.conf 反解区域

4) 修改数据库属主 / 属组

```
#chown named.named db.niliu
```

5) 检测 named 配置文件正确

```
#named-checkconf
```

6) 启动 named

```
#systemctl start named
```



GNU/Linux

配置主 DNS 服务器

5. 配置 named.conf 反解区域

7) 客户端配置

```
#vim /etc/resolv.conf
```

```
search niliu.edu
```

```
nameserver dns_server_ip
```

```
#dig -x ip
```



GNU/Linux

配置辅 DNS 服务器

1. 安装所需文件

```
#yum install bind
```

2. 配置辅 DNS

```
#vim /etc/named.conf
```

配置最小化



GNU/Linux

配置辅 DNS 服务器

3. 配置 named.conf 正解

1) 增加区域

```
zone "niliu.edu" IN { ← 增加 niliu.edu 区域  
    type slave; ← 服务器类型为 slave;  
    masters { 192.168.10.1; }; ← 指向 Master 服  
务器 IP  
    file "slaves/niliu.db" ; ← 区域数据库文件  
};
```

GNU/Linux

配置辅 DNS 服务器

3. 配置 named.conf 正解

1) 增加区域

zone “10.168.192.in-addr.arpa” IN { ← 增加
niliu.edu 区域

type slave; ← 服务器类型为 slave;

masters { 192.168.10.1; }; ← 指向 Master 服
务器 IP

file “slaves/db.niliu” ; ← 区域数据库文件

};

GNU/Linux

配置辅 DNS 服务器

3. 配置 named.conf 正解

2) 主 DNS 服务器配置

(1) 如配置文件最小化，可不用加此项

(2) 如 slave 服务器无法生成数据库文件且存放目录正确，则加入此句

```
allow-transfer { slave_dns_ip; };
```

/* 词句可加在 Option 中，或某个区域中



GNU/Linux

配置辅 DNS 服务器

3. 配置 named.conf 正解区域

3) 检测 named 配置文件正确

```
#named-checkconf
```

4) 启动 named

```
#systemctl start named
```

```
#systemctl status named
```



GNU/Linux

配置辅 DNS 服务器

3. 配置 named.conf 正解区域

5) 客户端配置

```
#vim /etc/resolv.conf
```

```
search niliu.edu
```

```
nameserver slave_dns_server_ip
```

```
#dig -x ip
```



GNU/Linux

配置转发 DNS 服务器

转发服务器属于一种中间转发器，其指向一个真实的 DNS 服务器，当客户端请求转发服务器做解析操作时，转发服务器先查看本地缓存中有否记录，若有则直接反馈给客户端；若没有再询问真实 DNS，得到结果后存入本地缓存中，再反馈给客户端。则当下次客户端再次询问相同地址时就不需要再次询问其他 DNS 服务器了。

GNU/Linux

配置转发 DNS 服务器

1. 安装 bind 程序

```
#yum install bind -y
```

2. 配置文件最小化



GNU/Linux

配置转发 DNS 服务器

3. 转发格式

转发器是在 `named.conf` 文件中的 `options` 区段设置的。主要用到两个配置选项：

`forwarders` : 指定要把查询请求转发到的远程域名服务器的 IP 地址。

```
forwarders { ip_addr [port ip_port]; [ ip_addr  
[port ip_port]; ... ] }
```

GNU/Linux

配置转发 DNS 服务器

3. 转发格式

forward : 启用域名转发功能。

forward only



GNU/Linux

配置转发 DNS 服务器

3. 转发格式

```
forwarders {  
    DNS_IP_1;  
    DNS_IP_2;  
};
```

forwarders 指令用于设置将 DNS 请求转发到哪个服务器，可以指定多个服务器的 IP 地址。



GNU/Linux

配置转发 DNS 服务器

3. 转发格式

`forward first | only;`

`forward` 指令用于设置 DNS 转发的工作方式:

`forward first` 设置优先使用 forwarders DNS 服务器做域名解析, 如果查询不到再使用本地网卡设置的 DNS 服务器做域名解析。

`forward only` 设置只使用 forwarders DNS 服务器做域名解析, 如果查询不到则返回 DNS 客户端

GNU/Linux

配置转发 DNS 服务器

4. 配置 named.conf

Forwarders { 192.168.10.1; }; ← 开启转发器
，并指定主 DNS 的地址

forward only; ← 仅转发

保存退出，重启 named 服务

如需清空 BIND 缓存



GNU/Linux

指定 dns 区域特性

```
zone "niliu.edu" IN {  
    type master;  
    allow-transfer { 指定辅 dns_ip; };  
    allow-update { none; }; ← 不允许动态更新  
};
```



GNU/Linux

指定 dns 区域特性

```
zone "niliu.edu" IN {  
    type forward:  
    forwarders { 指定 dns_srv_ip; };  
    forward only:  
};
```



GNU/Linux

dns 委派

DNS 委派又是一个比较重要的概念。到底什么 DNS 委派？就是把解析某个区域的权利转交给另外一台 DNS 服务器。

比如 A 主机可以解析 .edu 域而 A 主机上还需要解析 .edu 的一个子域 .niliu.edu 如果全部由 A 主机解析的话，那很可能 A 主机的负载会过重。此时一台主机 B 来专门解析 .niliu.edu 域。这样 A 主机的负载就会减小从而加快解析速度。这很合理，也很容易实现

GNU/Linux

dns 委派

可是这里存在一个问题。我们再次回顾解析一个完整域名的过程。假设我们现在需要解析 `www.niliu.edu`。DNS 肯定先找根域，然后再找 A 主机，因为 `.edu` 域的都由 A 主机解析，但是 A 主机只解析 `.edu` 域，并不解析 `.niliu.edu` 这个子域。如何解决这个问题，其实很简单。只需要在 A 主机的 `/etc/named.conf` 文件中声明 `.niliu.edu` 域。

GNU/Linux

dns 委派实现

Host1:

```
#vi /etc/named.conf
options {
    directory "/var/named";
};
zone "edu." {
    type master;
    file "named.edu" ;
```



GNU/Linux

dns 委派实现

Host1:

```
#cd /var/named
```

```
#vim named.edu
```

```
$TTL 1D
```

```
@      IN      SOA      rh7s1.niliu.edu. root.rh7s1.niliu.edu. (  
                                0      ;serial  
                                1D     ;refersh  
                                1H     ;retry  
                                1W     ;expire  
                                3H )   ;minimum
```



GNU/Linux

dns 委派实现

Host1:

创建 named.edu

@ IN NSrh7s1.niliu.edu.

niliu.edu. IN NSrh7s2.niliu.edu. ← 委派

rh7s2.niliu.edu. IN A 192.168.10.2

rh7s1 IN A 192.168.10.1

rh7s3 IN A 192.168.10.3

保存退出

//* 将 niliu.edu. 委派给 rh7s2.niliu.edu. 管理



GNU/Linux

dns 委派实现

Host1:

```
#chown named. named.niliu
```

```
#systemctl restart named
```



GNU/Linux

dns 委派实现

Host2:

```
#vi /etc/named.conf
options {
    directory "/var/named";
};
zone "niliu.edu." {
    type master;
    file "niliu.db" ;
```



GNU/Linux

dns 委派实现

Host2:

```
#cd /var/named
```

```
#vim niliu.db
```

```
$TTL 1D
```

```
@      IN      SOA      rh7s2.niliu.edu. root.rh7s2.niliu.edu. (  
                                0      ;serial  
                                1D     ;refersh  
                                1H     ;retry  
                                1W     ;expire  
                                3H )   ;minimum
```



GNU/Linux

dns 委派实现

Host2:

创建 niliu.db

```
@      IN      NSrh7s2.niliu.edu.  
rh7s2  IN  A    192.168.10.2  
rh7s3  IN  A    192.168.10.3  
@      IN  A    192.168.10.2
```

保存退出

/* 将 niliu.edu. 委派给 rh7s2.niliu.edu. 管理



GNU/Linux

dns 委派实现

Host2:

```
#chown named. named.niliu
```

```
#systemctl restart named
```



GNU/Linux

清除 DNS 缓存

1) 在 dns 服务上使用 rndc 方法
建立 rndc 密钥

```
#rndc-confgen -r /dev/urandom -a
```

```
#chown root:named /etc/rndc.key
```

```
#chmod 644 /etc/rndc.key
```

```
#systemctl restart named
```

```
#rndc flush
```



GNU/Linux

清除 DNS 缓存

2) 用 dns 缓存程序 nscd(name service cache daemon) 负责管理 dns 缓存

```
#systemctl restart nscd
```

