|  |
| --- |
| DNS实战 |
| **作者：**  **归档：学习笔记**  **2016/6/27** |
| **快捷键：**  Ctrl + 1 标题1  Ctrl + 2 标题2  Ctrl + 3 标题3  Ctrl + 4 实例  Ctrl + 5 程序代码  Ctrl + 6 正文 |
| **格式说明：**  蓝色字体：注释  黄色背景：重要  绿色背景：注意 |

****

**老男孩教育教学核心思想6重：重目标、重思路、重方法、重实践、重习惯、重总结**

**学无止境，老男孩教育成就你人生的起点！**

**联系方式:**

|  |  |
| --- | --- |
| **网站运维QQ交流群：** | |
| Linux 385168604 | 架构师 390642196 |
| Python 29215534 | 大数据 421358633 |
| **官方网站:** | |
| [**http://www.oldboyedu.com**](http://www.oldboyedu.com) |  |

目 录

[一、理论基础 1](#_Toc471644847)

[1.DNS的出现及演化 1](#_Toc471644848)

[2.DNS概述入门 1](#_Toc471644849)

[3.DNS域名解析过程 1](#_Toc471644850)

[4.DNS的分类 1](#_Toc471644851)

[5.DNS中记录类型 1](#_Toc471644852)

[6.DNS命名规范 2](#_Toc471644853)

[7、DIG、NSLOOKUP、HOST的介绍 2](#_Toc471644854)

[二、学习部署BIND9 4](#_Toc471644855)

[2.1.安装Master端 4](#_Toc471644856)

[2.2.安装Slave端： 8](#_Toc471644857)

[2.3.添加A、CNAME、MX、PTR记录 8](#_Toc471644858)

[2.4.使用DNS实现负载均衡 10](#_Toc471644859)

[2.5.配置DNS视图（智能DNS） 11](#_Toc471644860)

[三、构建企业级DNS 14](#_Toc471644861)

[3.1.硬件选型 14](#_Toc471644862)

[3.2.初始化系统 14](#_Toc471644863)

[3.3.构建高性能、高可用的DNS 14](#_Toc471644864)

[3.4.压测 15](#_Toc471644865)

[3.5.监控 16](#_Toc471644866)

[3.5.1.系统基础性能 16](#_Toc471644867)

[3.5.2.LOOPBACK的地址绑定状态监听 16](#_Toc471644868)

[3.5.3.DNS数据与MASTER一致性监控 16](#_Toc471644869)

[3.5.4.DNS响应时间 16](#_Toc471644870)

[3.5.5.DNS请求每秒请求数监控 17](#_Toc471644871)

[3.5.6.DNS可用性监控 18](#_Toc471644872)

[3.6.自动化 18](#_Toc471644873)

[3.6.1.SLASTACK安装、部署 18](#_Toc471644874)

[3.6.2.配置管理自动化 18](#_Toc471644875)

[3.6.3.BIND-BIZ的优缺点 18](#_Toc471644876)

[3.6.4.老师的方案 18](#_Toc471644877)

[3.7.安全 19](#_Toc471644878)

[3.8.日常运维规范 19](#_Toc471644879)

[四、DNS的发展趋势 19](#_Toc471644880)

[4.1.DNSMASQ 19](#_Toc471644881)

[4.2.HTTPDNS 19](#_Toc471644882)

[五、生产常见DNS问题以及排障方法 20](#_Toc471644883)

[5.1.DNS需要开通53的UDP和TCP 20](#_Toc471644884)

[5.2.运维同学突然告诉DNS不能用了 20](#_Toc471644885)

[5.3.突然收到报警，提示DNS虚地址不能正常解析 20](#_Toc471644886)

[5.4.开发找，说调用微信支付接口有超时现象，错误日志中报解析微信域名失败 20](#_Toc471644887)

[六、DNS面试题 21](#_Toc471644888)

[6.1.DNS中递归查询与迭代查询的区别 21](#_Toc471644889)

[6.2.DNS进行域名解析的过程 21](#_Toc471644890)

[八、附录 21](#_Toc471644891)

[本地DNS部署到公网： 22](#_Toc471644892)

# 一、理论基础

## 1.DNS的出现及演化

## 2.DNS概述入门

最大深度不得超过127层，每个节点都有一个可以长达63个字符的文本标号。

## 3.DNS域名解析过程

先看本身缓存记录，如果有就查询上级……

## 4.DNS的分类

主DNS服务器：就是一台存储着原始资料的DNS服务器。

从DNS服务器：使用自动更新方式从主DNS服务器同步数据的DNS服务器，也成辅助DNS服务器。

缓存服务器：不负责本地解析，采用递归方式转发客户机查询请求，并返回结果给客户机的DNS服务器，同时缓存查询回来的结果，也叫递归服务器。

转发器：这台服务器发现非本机的查询请求时，不再向根域发起请求，而是直接转发给指定的一台或多台服务器，自身不缓存查询结果。

## 5.DNS中记录类型

SOA：标注

NS：域名授权名称服务器，(记录每级域名对应的服务器在哪。)

MX：邮件交换

A：IPV4主机地址

AAAA：IPV6主机地址

PTR：反向解析

CNAME：别名

## 6.DNS命名规范

1、26个英文字母

2、"0,1,2,3,4,5,6,7,8,9"是个数字

3、"-"英文中的连词号

4、最多63字节长度

## 7、DIG、NSLOOKUP、HOST的介绍

yum install bind-utils

HOST命令也是大多数系统、软件库调用的解析命令，如php。

[root@iZwz91901kljyt0j66lycyZ ~]# host baidu.com

baidu.com has address 123.125.114.144

baidu.com has address 111.13.101.208

baidu.com has address 180.149.132.47

baidu.com has address 220.181.57.217

baidu.com mail is handled by 20 mx50.baidu.com.

baidu.com mail is handled by 20 mx1.baidu.com.

baidu.com mail is handled by 10 mx.n.shifen.com.

baidu.com mail is handled by 20 jpmx.baidu.com.

NSLOOKUP能够反映出是那个dnsserver返回的结果：

[root@iZwz91901kljyt0j66lycyZ ~]# nslookup baidu.com

Server: 100.100.2.138

Address: 100.100.2.138#53

Non-authoritative answer:

Name: baidu.com

Address: 180.149.132.47

Name: baidu.com

Address: 220.181.57.217

Name: baidu.com

Address: 111.13.101.208

Name: baidu.com

Address: 123.125.114.144

Dig返回整个解析过程详细路径，类似traceroute。

[root@iZwz91901kljyt0j66lycyZ ~]# dig baidu.com

; <<>> DiG 9.9.4-RedHat-9.9.4-38.el7\_3 <<>> baidu.com

;; global options: +cmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 29134

;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 5, ADDITIONAL: 6

;; OPT PSEUDOSECTION:

; EDNS: version: 0, flags:; udp: 4096

;; QUESTION SECTION:

;baidu.com. IN A

;; ANSWER SECTION:

baidu.com. 297 IN A 180.149.132.47

baidu.com. 297 IN A 123.125.114.144

baidu.com. 297 IN A 220.181.57.217

baidu.com. 297 IN A 111.13.101.208

;; AUTHORITY SECTION:

baidu.com. 66130 IN NS dns.baidu.com.

baidu.com. 66130 IN NS ns4.baidu.com.

baidu.com. 66130 IN NS ns3.baidu.com.

baidu.com. 66130 IN NS ns2.baidu.com.

baidu.com. 66130 IN NS ns7.baidu.com.

;; ADDITIONAL SECTION:

dns.baidu.com. 46088 IN A 202.108.22.220

ns2.baidu.com. 60074 IN A 61.135.165.235

ns3.baidu.com. 59995 IN A 220.181.37.10

ns4.baidu.com. 43924 IN A 220.181.38.10

ns7.baidu.com. 60074 IN A 180.76.76.92

;; Query time: 2 msec

;; SERVER: 100.100.2.138#53(100.100.2.138)

;; WHEN: Thu Jan 05 22:23:28 CST 2017

;; MSG SIZE rcvd: 272

# 二、学习部署BIND9

## 2.1.安装Master端

环境准备：

yum源

安装包：

yum install bind-utils bind bind-devel bind-chroot

主配置文件：

[root@m01 ~]# cat /etc/named.conf

options {

version "1.1.1";

listen-on port 53 { any; };

directory "/var/named/chroot/etc/";

pid-file "/var/named/chroot/var/run/named/named.pid";

allow-query { any; };

Dump-file "/var/named/chroot/var/log/binddump.db";

statistics-file "/var/named/chroot/var/log/named\_stats"; #可用于监控

zone-statistics yes; #配置成yes才会网上边文件记录

memstatistics-file "log/mem\_stats";

empty-zones-enable no;

forwarders {202.106.196.115;8.8.8.8; };

};

key "rndc-key" {

algorithm hmac-md5;

secret "Eqw4hC1GExUWeDkKBX/pBg==";

};

controls {

inet 127.0.0.1 port 953

allow { 127.0.0.1; } keys { "rnde-key"; };

};

logging {

channel warning {

file "/var/named/chroot/var/log/dns\_warning" versions 10 size 10m;

severity warning;

print-category yes;

print-severity yes;

print-time yes;

};

channel general\_dns {

file "/var/named/chroot/var/log/dns\_log" versions 10 size 100m;

severity info;

print-category yes;

print-severity yes;

print-time yes;

};

category default {

warning;

};

category queries {

general\_dns;

};

};

include "/var/named/chroot/etc/view.conf"; #智能DNS

编辑/etc/rndc.key

[root@m01 ~]# cat /etc/rndc.key

key "rndc-key" {

algorithm hmac-md5;

secret "Eqw4hC1GExUWeDkKBX/pBg==";

};

编辑/etc/rndc.conf

[root@m01 ~]# cat /etc/rndc.conf

key "rndc-key" { #调用哪个key

algorithm hmac-md5;

secret "Eqw4hC1GExUWeDkKBX/pBg==";

};

options {

default-key "rndc-key";

default-server 127.0.0.1;

default-port 953;

};

编辑/var/named/chroot/etc/view.conf

view "View" { #一个说明，标注这个名称

zone "lnh.com" {

type master; #还有backup，主从

file "lnh.com.zone"; #zone文件中的哪个zone，没写目录表示zone文件同目录

allow-transfer { #允许谁来向我同步数据，填写对应的slave

172.16.1.51;

};

notify yes; #配置发生变更，通知slave端同步

also-notify {

172.16.1.51;

};

};

};

编辑/var/named/chroot/etc/lnh.com.zone

$ORIGIN .

$TTL 3600 ;1 hour

lnh.com IN SOA op.lnh.com. dns.lnh.com. (

2000 ; serial

900 ; refresh (15 minutes)

600 ; retry (10 minutes)

86400 ; expire (1 day)

3600 ; minimum (1 hour)

)

NS op.lnh.com.

$ORIGIN lnh.com.

shanks A 1.2.3.4

op A 1.2.3.4

a A 1.2.3.4

修改目录权限，并启动服务

cd /var/ && chown -R named:named named/

/etc/init.d/named start

chkconfig named on

试一把能不能解析

[root@m01 /var/named/chroot/etc]# dig @127.0.0.1 shanks.lnh.com

; <<>> DiG 9.8.2rc1-RedHat-9.8.2-0.47.rc1.el6\_8.3 <<>> @127.0.0.1 shanks.lnh.com

; (1 server found)

;; global options: +cmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 45035

;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:

;shanks.lnh.com. IN A

;; ANSWER SECTION:

shanks.lnh.com. 3600 IN A 1.2.3.4

;; AUTHORITY SECTION:

lnh.com. 3600 IN NS op.lnh.com.

;; ADDITIONAL SECTION:

op.lnh.com. 3600 IN A 1.2.3.4

;; Query time: 1 msec

;; SERVER: 127.0.0.1#53(127.0.0.1)

;; WHEN: Sat Jan 7 12:23:30 2017

;; MSG SIZE rcvd: 81

注：更改配置使用rndc reload重载配置。

## 2.2.安装Slave端：

以下只把不同的配置写出：

编辑/var/named/chroot/etc/view.conf文件：

view "SlaveView" {

zone "lnh.com" {

type slave;

masters {172.16.1.61; };

file "slave.lnh.com.zone";

};

};

更改目录权限及启动服务：

cd /var/ && chown -R named:named named/

/etc/init.d/named start

chkconfig named on

## 2.3.添加A、CNAME、MX、PTR记录

编辑master节点/var/named/chroot/etc/lnh.com.zone，在文件末尾添加：

[root@m01 /var/named/chroot/etc]# cat lnh.com.zone

$ORIGIN .

$TTL 3600 ;1 hour

lnh.com IN SOA op.lnh.com. dns.lnh.com. (

**2001** ; serial #用于更新slave端配置

900 ; refresh (15 minutes)

600 ; retry (10 minutes)

86400 ; expire (1 day)

3600 ; minimum (1 hour)

)

NS op.lnh.com.

$ORIGIN lnh.com.

shanks A 1.2.3.4

op A 1.2.3.4

a A 1.2.3.4

**a A 172.16.1.100**

使用rndc reload

添加CNAME记录

cname CNAME a.lnh.com. #保证域名能解析

添加MX记录：

mx MX 5 172.16.1.100 #5表示优先级，越小越高

尝试解析：

[root@db01 etc]# host mx.lnh.com 127.0.0.1

Using domain server:

Name: 127.0.0.1

Address: 127.0.0.1#53

Aliases:

mx.lnh.com mail is handled by 5 172.16.1.100.lnh.com.

添加PTR记录：

编辑/var/named/chroot/etc/view.conf配置文件：添加一个新的zone

zone "16.172.in-addr.arpa" {

type master;

file "16.172.zone";

allow-transfer {

172.16.1.51;

};

notify yes;

also-notify {

172.16.1.51;

};

};

编辑16.172.zone文件：

[root@m01 /var/named/chroot/etc]# cat 16.172.zone

$TTL 3600 ; 1 hour

@ IN SOA op.lnh.com. dns.lnh.com. (

2004 ;serial

900 ;refresh(15 minutes)

600 ;retry(10 minutes)

86400 ;expire(1 day)

3600 ;minimum(1 hour)

)

NS op.lnh.com.

102.122 IN PTR a.lnh.com.

更改权限：

chown -R named:named 16.172.zone

编辑Slave端的view.conf,添加如下配置：

zone "16.172.in-addr.arpa" {

type slave;

masters {172.16.1.61; };

file "slave.16.172.zone";

};

使用rndc reload更新配置。slave端才会通过不新的zone配置。

解析测试：

[root@db01 etc]# host 172.16.122.102 127.0.0.1

Using domain server:

Name: 127.0.0.1

Address: 127.0.0.1#53

Aliases:

102.122.16.172.in-addr.arpa domain name pointer a.lnh.com.

## 2.4.使用DNS实现负载均衡

通过DNS轮询的方式，将同一个域名轮询到不同的主机。

a A 172.16.1.100

a A 172.16.1.102

测试轮询结果：

[root@m01 /var/named/chroot/etc]# nslookup a.lnh.com 127.0.0.1

Server: 127.0.0.1

Address: 127.0.0.1#53

Name: a.lnh.com

Address: 172.16.1.100

Name: a.lnh.com

Address: 172.16.1.102

Name: a.lnh.com

Address: 1.2.3.4

[root@m01 /var/named/chroot/etc]# nslookup a.lnh.com 127.0.0.1

Server: 127.0.0.1

Address: 127.0.0.1#53

Name: a.lnh.com

Address: 172.16.1.102

Name: a.lnh.com

Address: 1.2.3.4

Name: a.lnh.com

Address: 172.16.1.100

注：DNS策略很简单，不会对后端服务器进行监测。

## 2.5.配置DNS视图（智能DNS）

任意机房出现网络、节点故障，不影响业务。节省专线流量。负载均衡和DNS联动，负载un横配置DNS的域名。指定那个ip访问那个地址。

编辑master上的主配置文件/var/named/chroot/etc/named.conf。

key "group1" {

algorithm hmac-md5;

secret "i+uKnzR3JfSheMVLNWUH5Q==";

};

key "group2" {

algorithm hmac-md5;

secret "wPaGrqLp0XtAepn8Rc0U3w==";

};

acl group1 {

172.16.1.51;

};

acl group2 {

172.16.1.61;

};

注：这俩是新的节点，在最后的incloude上方。说明：group1就相当于idc1中的客户机，分组下的机器发起的请求，分到group1中。如果不再这里边的发起DNS请求走默认。

配置不同的key是为了做主从区分不同的域。记得在rndc.key和rndc.conf文件中配置相关key。

编辑/var/named/chroot/etc/view.conf配置文件：（清空源内容）

view "GROUP1" {

match-clients { key "group1";group1; };

server 172.16.1.51 { keys "group1";};

zone "viewlnh.com" {

type master;

file "group1.viewlnh.com.zone";

};

};

view "GROUP2" {

match-clients { key "group2";group2; };

server 172.16.1.51 { keys "group2"; };

zone "viewlnh.com" {

type master;

file "group2.viewlnh.com.zone";

};

};

#从服务器的view配置：

view "GROUP1" {

match-clients { key "group1";group1; };

server 172.16.1.61 { keys "group1";};

zone "viewlnh.com" {

type slave;

masters {172.16.1.61; };

file "slave.group1.viewlnh.com.zone";

};

};

view "GROUP2" {

match-clients { key "group2";group2; };

server 172.16.1.61 { keys "group2";};

zone "viewlnh.com" {

type slave;

masters {172.16.1.61; };

file "slave.group2.viewlnh.com.zone";

};

};

编辑/var/named/chroot/etc/group1.viewlnh.com.zone文件：

$ORIGIN .

$TTL 3600 ; 1 hour

viewlnh.com IN SOA op.viewlnh.com. dns.view.lnh.com. (

2005 ; serial

900 ; refresh (15 minutes)

600 ; retry (10 minutes)

86400 ; expire (1 day)

3600 ; minimum (1 hour)

)

NS op.viewlnh.com.

$ORIGIN viewlnh.com.

op A 172.16.1.61

view A 172.16.1.61

编辑/var/named/chroot/etc/group2.viewlnh.com.zone文件：

$ORIGIN .

$TTL 3600 ; 1 hour

viewlnh.com IN SOA op.viewlnh.com. dns.view.lnh.com. (

2005 ; serial

900 ; refresh (15 minutes)

600 ; retry (10 minutes)

86400 ; expire (1 day)

3600 ; minimum (1 hour)

)

NS op.viewlnh.com.

$ORIGIN viewlnh.com.

op A 172.16.1.51

view A 172.16.1.51

两个组解析到不同的地址上去。不同的客户端请求DNS分配到指定的group中去解析。

更改配置文件权限：

chown -R named:named group\*

#重载配置文件

rndc reload

分配配置的两个ip访问测试：

[root@db01 etc]# host view.viewlnh.com 172.16.1.61

Using domain server:

Name: 172.16.1.61

Address: 172.16.1.61#53

Aliases:

view.viewlnh.com has address 172.16.1.61

[root@m01 /var/named/chroot/etc]# host view.viewlnh.com 172.16.1.61

Using domain server:

Name: 172.16.1.61

Address: 172.16.1.61#53

Aliases:

view.viewlnh.com has address 172.16.1.51

# 三、构建企业级DNS

## 3.1.硬件选型

CPU：12c以上配置

内存：16G

网络：千兆

注：0延迟这个配置可以达到3万/s

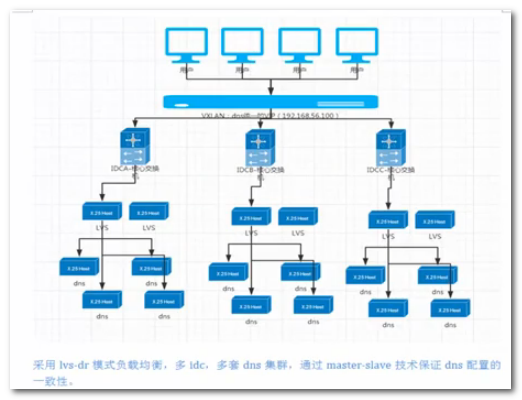
## 3.2.初始化系统

关闭selinux

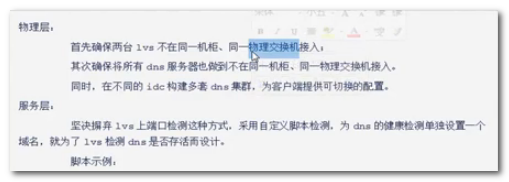
关闭iptables：

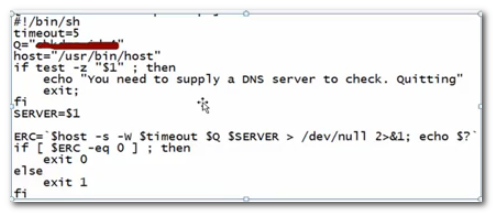
调整ulimit限制：echo -e '\* soft nproc 65546\n\* hard nproc 65536\n\* spft nofile 65536\n\* hard nofile65536\n' >> /etc/security/limits.conf

## 3.3.构建高性能、高可用的DNS



1、高可用





客户端层：

多idc之间的流量切换是通过客户端的健康监测cron实现的，脚本每分钟运行一次，分别检测每个dns集群虚地址的可用性。

2、高性能

通过lvs可以对每个集群做横向扩容，是否需要扩容的一句是对现有系统的压测结果，以及实时的监控数据。

亦或者可以在最靠近应用层处，加上一层cache-only集群，但前提是你的线上环境中，没有任何系统依赖于dns负载均衡。如果数据库实时读写的情况使用的dns做缓存，使用cache会出现脏数据，比如cache是15分钟切换，数据库是15分3秒，机器已经切换了，但是数据库还是连接并写入了旧的机器上。

## 3.4.压测

安装queryperf

下载bind源码：wget <http://ftp.isc.org/isc/bind9/9.7.3/bind-9.7.3.tar.gz>

解压：tar xf bind-9.7.3.tar.gz

进入目录：cd bind-9.7.3/contrib/queryperf/

编译:./configure

make

会在当前目录下出现queryperf，可以将他拷贝到/usr/bin/下

编辑域名记录文件（test.txt），格式如下：

[www.baidu.com](http://www.baidu.com) A

[www.baidu.com](http://www.baidu.com) A

[www.baidu.com](http://www.baidu.com) A

[www.baidu.com](http://www.baidu.com) A

[www.baidu.com](http://www.baidu.com) A #写多少就是多少线程。

压测命令：queryperf -d test.txt -s 8.8.8.8

按照3.1.中的硬件选型，也做了3.2.的初始化，这样在相同idc的压测下，单台dns server的qps可以达到35~40k。

## 3.5.监控

结合zabbix实现。

### 3.5.1.系统基础性能

CPU、内存、主机存活、磁盘空间、主机运行时间系统load

### 3.5.2.LOOPBACK的地址绑定状态监听

dnsserver在集群中充当realserver角色，dr中，需要绑定loopback，lvs特性。（MAC转换为后端真实服务器，所以后端真实服务器得知道包中的ip是自己。）

### 3.5.3.DNS数据与MASTER一致性监控

zabbix自定义discovery，扫出dns配置中所有zone（view.conf文件中包含了所有zone），然后分别对比slave和master每个zone的serial值，持续5分钟不一致时报警。

二是写脚本，没15分钟扫一遍master上所有域名解析结果，与每个slave的结果做对比，当出现结果不一致情况时，报警。

### 3.5.4.DNS响应时间

通过dig命令可以拿到响应时间。

[root@m01 ~]# **dig @127.0.0.1 view.viewlnh.com**

; <<>> DiG 9.8.2rc1-RedHat-9.8.2-0.47.rc1.el6\_8.3 <<>> @127.0.0.1 view.viewlnh.com

; (1 server found)

;; global options: +cmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: REFUSED, id: 55437

;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 0

;; WARNING: recursion requested but not available

;; QUESTION SECTION:

;view.viewlnh.com. IN A

**;; Query time: 1 msec #相应时间**

;; SERVER: 127.0.0.1#53(127.0.0.1)

;; WHEN: Sun Jan 8 10:34:15 2017

;; MSG SIZE rcvd: 34

### 3.5.5.DNS请求每秒请求数监控

在每台dns主机上，编写zabbix脚本，分析named\_stats文件(在named.conf配置文件中有配置路径)，获取每秒请求数。或使用rndc stats命令获取

#!/bin/bash

#rndc stats

STATS='/var/named/chroot/var/log/named\_stats'

if [[ $# -ne 1 ]]

then

echo "$0 [querys]"

exit 2

else

which=$1

fi

if [[ -f "${STATS}" ]]

then

echo > ${STATS}

rndc stats >/dev/null 2>&1

else

echo "${STATS} not found."

exit 2

fi

case ${which} in

querys)

RESULT=`awk '{if ($2=="QUERY") {print $1}}' ${STATS}`;;

\*)

echo "$0 {querys}"

exit2

;;

esac

echo ${RESULT}

执行rndc stats命令之后，会往文件中追加新信息，通过awk获取querys的值，最后一个肯定就是新的tail 1或者，执行前清空文件，在获取肯定是最新的。

### 3.5.6.DNS可用性监控

远端一组主机跑在fullnai下（提供高可用），通过host命令检测dnsserver的可用性，脚本与lvs健康检测脚本类似。

## 3.6.自动化

### 3.6.1.SLASTACK安装、部署

通过定制salstack配置，实现自动、批量安装、部署dns。

### 3.6.2.配置管理自动化

业界最多的是bind-biz，biz是指将所有的配置都存在mysql表中，对bind做特殊配置，使得每次bind接受的请求都去mysql中查询数据之后返回给用户。

### 3.6.3.BIND-BIZ的优缺点

优势：将数据全部存在数据库，符合运维开发的理念。

缺点：每次解析都要select数据库，性能底下；

增加了系统的耦合性，还需要保证mysql的高可用。

### 3.6.4.老师的方案

采用biz的数据库部分表结构。

用thinkphp实现对mysql的增删改查，和一些权限控制的页面，在该页面，用户可以完成对域名的增删改查操作，数据源在MySQL中。

通过saltstack+py实现对mysql中调数据，生成bind的配置文件，并检测文件格式，之后reload：

开源地址：<https://github.com/shanks1127/dns>

## 3.7.安全

时刻关注dns相关的漏洞、补丁；

选用稍大些的厂商作为域名服务器，我们是万网；

对服务器的登录日志做监控分析。

## 3.8.日常运维规范

DNS作为基础服务，在做好高可用、高性能、好扩容的基础上，任何时刻都不能掉以轻心！！！

确保所有监控均处于生效状态；

所有新机器，均在Saltstack上完成初始化和完成、部署操作、不能单独操作。

所有针对dns架构调整的操作，均需在流量低谷时操作；

对集群扩容操作时，务必对新加入节点做压测，同时重启服务器并监测重启后各项指标是否正常。

关注dns相关新闻，时刻跟进。

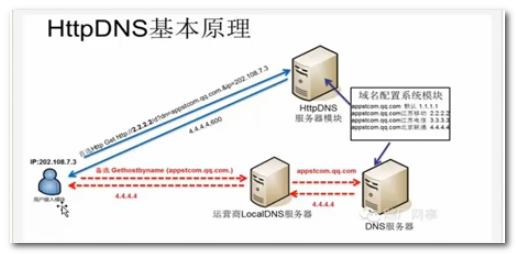
# 四、DNS的发展趋势

## 4.1.DNSMASQ

轻量级的DNS和DHCP配置工具，适用于小型网络，同时也有DNS功能，可以作为缓存使用。

## 4.2.HTTPDNS

普通DNS会有DNS劫持。再加上不同运营商不同节点的的缓存时间的不同。



适合http有客户端的请求。

# 五、生产常见DNS问题以及排障方法

## 5.1.DNS需要开通53的UDP和TCP

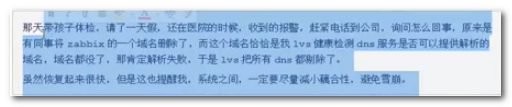
机器和idc机房出口。traceroute命令测试线路。

## 5.2.运维同学突然告诉DNS不能用了

查看dns进程是否存在，系统瓶颈，如果开着iptables，查看日志，发现有报netfilter表满。停掉iptables。

在iptables运行时，即便是没有任何规则，所有的报文也要过netfilter表，而dns这样高并发的 很容易把netfilter表填满，从而影响服务。

## 5.3.突然收到报警，提示DNS虚地址不能正常解析



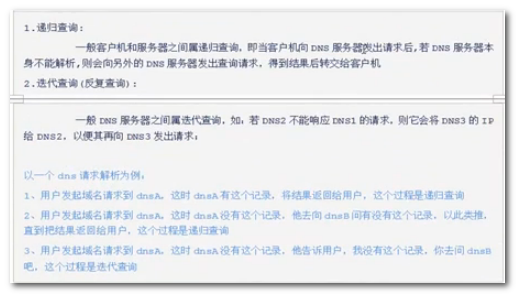
## 5.4.开发找，说调用微信支付接口有超时现象，错误日志中报解析微信域名失败

微信的forwarders请求的限制，本地的named.conf配置文件中的forwarders加入腾讯的dns全为ip。

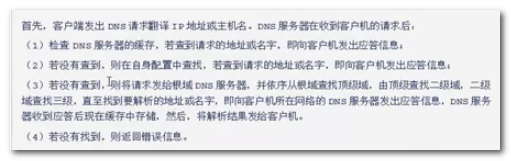
不要轻易更改线上，通过traceroute命令和host以及dig测试。

# 六、DNS面试题

## 6.1.DNS中递归查询与迭代查询的区别



## 6.2.DNS进行域名解析的过程



注：首先客户端先监测有没有缓存。没有的话监测hosts。

# 八、附录

通过源码安装bind：

./configure --sysconfdir=/etc/named --prefix=/usr/local/bind --enable-epoll --disable-ipv6 --disable-chroot --enable-threads

make && make install

# 本地DNS部署到公网：

工作的域名做NS记录对应到外网的域名解析，万网目前需要提工单，更改NS对应的ip。