OSI参考模型:适用于所有的网络,先有模型,后有协议
应用层:应用程序产生数据,和用户的接口,如http协议
表示层:编码转换,压缩,解压缩,加密,解密等
会话层:建立维护拆除会话

传输层:PDU:数据段 规定了应用程序的接口 TCP和UDP
TCP:传输控制协议:可靠地传输,面向连接的传输,效率低
TCP控制位:
URG:紧急指针有效位,标记紧急数据
ACK:确认序列号有效位
PSH:通知接收端把数据交给进程,不要在缓冲区中停留
RST:请求重新建立三次握手
SYN:请求建立三次握手
FIN:请求断开连接

TCP流控机制:滑动窗口

TCP计时器:重传计时器 坚持计时器 保活计时器 时间等待计时器

UDP:用户数据报协议,面向无连接协议,效率高

端口号:
1-1023:知名端口,管理端口
1024-49151:注册端口
49152-65535:随机端口

网络层:PDU:数据包 设备:路由器 逻辑地址:IP地址 网络层作用:为数据传输选择一条最优的路径

路由器工作原理:根据路由表转发数据,路由表分为直连路由和非直连路由
直连路由:为接口配置IP地址并保证为UP状态
非直连路由:通过静态指定或动态学习
IP地址是不可靠的地址,采用尽力而为的发送方式,把数据包送到目标地址所有的网络
IP地址分类:网络地址和主机地址,私有地址范围
A:10.0.0.0-10.275.275.275 子网掩码为8位
B:172,.16.0.0-172.31.275.275 子网掩码为16位
C:192.168.0.0-192.168.275.275 子网掩码为24

D类地址是组播地址 E类地址用作科研
IP包头格式:
版本:4位,标识IP地址的版本,常用的是IPv4和IPv6
首部长度:4位,可变长度,根据可选项变化
优先级与服务类型:8位,提供第三层的服务质量
总长度:16位,IP包头总长度
标识符:16位,为分段数据打上标记,方便到目标地址重新组装
标志:3位,第一位没有设置,第二位是DF,决定是否分片,0代表部分片,1代表分片;第三位是MF:代表更多的片,只有数据的最后一片值为0
段偏移量:13位,代表数据片在整个数据中的位置
TTL:8位,生命周期,数据包经过路由器个数,数据包每经过一个路由器,TTL值减一,当TTL值为0时,数据还没有到达目标地址,该数据将被丢弃,一般设置为32或64
协议号:8位,标识上层协议号
首部校验和:16位,验证发送端和接收端数据的完整性
源地址:32的IP地址
目标地址:32的IP地址
可选项:松散路由 严格路由 路由标记 时间戳
填充:把IP包头每行都填充为32位

数据链路层:PDU:数据帧 设备:交换机 MAC地址:48位,前24是厂商的地址,后24位是厂商为设备分配的地址
协议:ARP:地址解析协议(把IP地址转换成MAC地址) RARP:反向地址解析协议(把MAC地址转换成IP地址)
帧格式:
目标地址:MAC地址 源地址:MAC地址 类型:标识上层协议类型 数据:上层数据 FCS:CRC32
MTU:最大传输单元,1500字节

交换机工作原理:根据MAC地址表转发数据
学习:只学习数据帧中的源地址
广播:当交换机接收数据帧后,查看MAC地址表中有没有数据帧中的目标MAC地址,如果没有,进行广播发送数据帧,交换机的广播除去源端口之外的所有端口都能接收到,所以交换机端口同处于一个广播域
转发:当交换机接收数据帧后,查看MAC地址表中有没有数据帧中的目标MAC地址,如果有目标MAC地址,进行单播转发
更新:交换机默认更新时间为300秒,从MAC地址最后一次通讯完成开始计时

广播域:能够接收同一个广播数据包所有节点的集合
CSMA/CD—带冲突检测的载波监听多路访问:解决链路冲突
CSMA/CD:先听后发,边听边发,冲突停发,等待重发

物理层:介质和信号 介质:双绞线 光纤 同轴电缆 信号:模拟信号和数字信号
物理层关心的问题:电气特性 规程特性 功能特性 物理特性 PDU:比特流 物理层从数据链路层接收数据,为数据传输提供安全的通信信道
物理层设备:集线器 HUB等

集线器工作原理:共享通道,所有的数据都是广播

TCP/IP协议模型:只适用于TCP/IP网络,先有协议,后有模型
应用层
传输层
网络层
数据链路层
物理层

IP:PORT=socket,套接字会生成进程

DNS:Domain Name System(域名系统)
域名:www.baidu.com(主机名)

主机---百度:一对一通讯,IP地址,IP地址定位一个节点

FQDN:Full Quaclifiled Domain Name

FQDN--IP:解析(DNS完整解析过程中需要有数据库)

DNS:FQDN---IP:正向解析过程
IP---FQDN:反向解析过程

/etc/nsswitch.conf:
hosts: files dns myhostname
files: /etc/hosts
dns: DNS服务器

/etc/hosts:
192.168.1.1 www.baidu.com www

IANA:FQDN---IP
ICNAA:民间组织,管理顶级域名

ping www.baidu.com www.baidu.com---IP转换

stub resolver:名称解析器

IANA组织设置全新的架构来解析计算机名和IP地址对应关系,就是DNS

DNS采用结构化分布式数据库进行管理

tom:中国 北京 海淀区 知春路 40：tom
中国 上海 徐家汇 西藏路 50：tom

www.baidu.com.

DNS层次结构从根开始:
根 .:全球有13个根节点
TLD(Top Level Domain):顶级域
  组织域:.com .net .gov .edu等
  国家域: .cn .hk .tw .jp等
二级域:baidu sina sohu等
三级域:sports auto等

DNS数据库层次结构:
  上级只知道直属下级的存在
  主机只知道跟的存在

DNS查询解析结果时:目标DNS服务器给返回给源DNS服务器:解析结果和缓存时间(TTL)

DNS查询方式:递归查询:发送一次请求(只接受本地客户端递归查询)
        迭代查询:发送多次请求,根服务器不为任何服务器提供迭代查询
DNS查询是分段的:客户端向DNS服务提交的查询是递归查询
          DNS服务器查询是迭代查询

DNS查询方式:内部主机查询外部主机,外部主机查询内部主机,内部主机查询内部主机
  内部查询外部主机:非权威答案
  外部查询内部主机:权威答案
        否定答案和肯定答案 TTL(缓存时间长度)

DNS:一个IP地址可以对应多个主机名,一个主机名可以对应多个IP地址

DNS解析的结果:只有自己直属下级的解析结果,才是权威答案

DNS类型:
  主DNS:可以完成数据的修改
  辅助DNS:辅助DNS服务器向主DNS服务器请求数据更改
    serial number:请求序列号
    refresh:请求更新时间
    retry:请求重试时间(该时间一定小于更新时间)

expire:过期时间

minittl:否定答案的缓存时间

如果在expire(过期时间)内,主DNS依然没有响应辅助DNS请求,辅助DNS也将停止工作

缓存DNS服务器:只做区域的高速DNS缓存查询

DNS转发器:不负责查询,只负责转发

时间单位: D(天) W(周) H(小时) 默认的时间单位是秒

DNS:利用数据库中的条目去解析,Resource Record(RR)

DNS主要资源记录类型(RRT:Resource Record Type)

A:正向解析记录,把FQDN解析为IP地址

www.gooann.com. 600 IN A 192.168.1.1

或简写为

www 600 IN A 192.168.1.1

PRT(Pointer):反正指针,把IP地址解析成FQDN

192.168.1.1 600 IN PTR www.gooann.com.

或简写为:

1 600 IN PRT www.gooann.com.

MX(Mail Exchanger):邮件交换记录 abc@gooann.com

优先级:0-99,数字越小优先级越高

gooann.com. 600 IN MX 10 mail.gooann.com.

gooann.com. 600 IN MX 20 mail2.gooann.com.

mail.gooann.com. 600 IN A 192.168.1.2

mail2.gooann.com. 600 IN A 192.168.1.3

NS(Name Server):名称服务器,指明该区域中哪台主机是DNS服务器(通常位于DNS服务器区域数据库中第二行)

gooann.com. 600 IN NS ns1.gooann.com.

gooann.com. 600 IN NS ns2.gooann.com.

gooann.com. 600 IN NS ns3.gooann.com.

ns1.gooann.com. 600 IN A 192.168.10.10

ns2.gooann.com. 600 IN A 192.168.10.20

ns3.gooann.com. 600 IN A 192.168.10.30

SOA(start of authority):起始授权机构,指明该区域中哪台DNS是主DNS服务器,必须位于区域数据库文件中的第一行

gooann.com. 600 IN SOA ns1.gooann.com. admin.gooann.com. (

1000 ; serial number

300 ; refresh
200 ; retry
500 ; expire
30 ; minittl
)

CNAME:(Canonical Name):别名

www.gooann.com. 600 IN A 192.168.10.1

www2.gooann.com. 600 IN CNAME www.gooann.com.

区域文件格式:

名称(name) 缓存时间(TTL:可省略) IN(intenet) RRT(资源类型) 数据(value)

申请域名:gooann.com IP:192.168.10.0/24

www.gooann.com 10.1
ftp.gooann.com 10.2
ns1.gooann.com 10.3
ns2.gooann.com. 10.4
mail.gooann.com 10.5

DNS正向区域数据库:

gooann.com. 600 IN SOA ns1.gooann.com. admin.gooann.com. (
1000 ; serial number
300 ; refresh
200 ; retry
500 ; expire
30 ; minittl
)

gooann.com. 600 IN NS ns1.gooann.com.

gooann.com. 600 IN NS ns2.gooann.com.

ns1.gooann.com. 600 IN A 192.168.10.3

ns2.gooann.com. 600 IN A 192.168.10.4

www.gooann.com. 600 IN A 192.168.10.1

ftp.gooann.com. 600 IN A 192.168.10.2

mail.gooann.com. 600 IN A 192.168.10.5
gooann.com. 600 IN MX 10 mail.gooann.com.

DNS反向区域数据库:(10.168.192.in-addr.apra)

```
gooann.com. 600 IN SOA ns1.gooann.com. admin.gooann.com. (
                                                    1000 ; serial number
                                                    1H  ; refresh
                                                    200  ; retry
                                                    500  ; expire
                                                    30   ; minittl
                                                    )
gooann.com. 600 IN NS ns1.gooann.com.
gooann.com. 600 IN NS ns2.gooann.com.
3.10.168.192.in-addr.arpa 600 IN PTR ns1.gooann.com.
4.10.168.192.in-addr.arpa 600 IN PTR ns1.gooann.com.
1.10.168.192.in-addr.arpa 600 IN PTR www.gooann.com.
2.10.168.192.in-addr.arpa 600 IN PTR ftp.gooann.com.
5.10.168.192.in-addr.arpa 600 IN PTR mail.gooann.com.
gooann.com. 600 IN MX 10 mail.gooann.com.
```

域:domain(逻辑概念)
区域:zone(物理概念)

主DNS和辅助DNS:
refresh:1H 10:00更新完成 主DNS 10:05更新数据

主DNS和辅助DNS数据同步过程:区域传送
区域传送类型:
全区域传送:axfr
增量区域传送:ixfr
区域类型:
主DNS:master
辅助DNS:slave
提示区域(根提示):hint,命名方式:a-root.server 202.106.0.20
转发区域:forward

DNS实现工具:bind
Berkeley Internet Name Domain
ISC:dhcp www.isc.org

BIND:rpm包和编译安装源代码

bind.x86_64:主程序包,版本9.9.4
bind-libs:DNS库程序包
bind-utils:DNS工具包
bind-devel:为DNS二次开发提供头和库文件

bind主程序包的列表:
/etc/named.conf:DNS主配文件(定义了全局设置和区域的设置)
/etc/named.rfc1912.zones:头文件包(新定义的区域设置)
/etc/rndc.conf:Remote Name Domain Controller(远程控制工具rndc的配置文件)
/etc/rndc.key:rndc的key文件
/run/named:目录存放PID文件
/usr/lib/systemd/system/named-setup-rndc.service:named-setup-rndc.service服务配置文件
/usr/lib/systemd/system/named.service:DNS服务配置文件
/usr/sbin:存放运行脚本
/usr/share/doc/bind-9.9.4:存放DNS帮助文档
/var/named:存放DNS区域文件

bind-utils:提供了四个工具:dig host nslookup nsupdate

bind-libs:提供DNS库文件支持

DNS主配文件/etc/named.conf详细解释:
```
options {   //影响zone设置
        listen-on port 53 { 127.0.0.1; };   //监听端口和IP
        listen-on-v6 port 53 { ::1; };      //监听IPv6端口和IP
        directory  "/var/named";            //区域配置文件的目录
```

```
        dump-file       "/var/named/data/cache_dump.db";  //缓存文件
        statistics-file "/var/named/data/named_stats.txt";  //静态缓存(很少用)
        memstatistics-file "/var/named/data/named_mem_stats.txt"; //内存缓存(很少用)
        allow-query     { localhost; };  //允许查询的客户端
        recursion yes;   //是否允许递归查询

        dnssec-enable yes;   //安全相关
        dnssec-validation yes;   //安全相关
        bindkeys-file "/etc/named.iscdlv.key";  //安全相关

        managed-keys-directory "/var/named/dynamic";   //key文件

        pid-file "/run/named/named.pid";   //PID文件
        session-keyfile "/run/named/session.key";  //会话Key文件
};

logging {    //日志相关
        channel default_debug {
                file "data/named.run";
                severity dynamic;
        };
};

zone "." IN {   //zone选项设置区域
        type hint;   //区域类型:hint master slave forward
        file "named.ca";   //区域文件
};

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

DNS服务器监听端口:

- tcp:53 负责DNS服务器之间主从复制
- udp:53 负责客户端查询
- tcp:953 rndc监听端口

套接字socket:服务器中负责响应客户端请求的端口
IP:PORT 套接字

named-checkconf:检查DNS主配文件的语法

named-checkzone:检查DNS区域配置文件语法
named-checkzone "zone_name" "zone_config_files"

netstat:显示网络连接状况,路由表等信息
netstat -antpu 或者 netstat -tlnp
-a:监听所有端口
-n:以数字形式显示
-t:监听tcp的状态
-u:监听udp的状态
-p:显示PID和应用程序命令
-l:只显示被监听的套接字

gooann.net 192.168.10.0/24:
ns:ns1.gooann.net 192.168.10.10
www:192.168.10.10
ftp:192.168.10.20

1:在主配文件中添加zone
2:设置区域配置文件

dig:
-t:指定查询类型:
@DNS服务器地址:指定在哪台服务器上查询
+[no]recure:是否启用递归查询
+[no]trace:追踪递归查询过程

定义给某个网段进行递归查询:修改/etc/named.conf,在options字段中添加:allow-recursion { 网段地址/掩码长度; };
定义不给任何网段进行递归查询:修改/etc/named.conf,在options字段中添加:recursion no;

dig baidu.com:命令解析

; <<>> DiG 9.9.5-3ubuntu0.6-Ubuntu <<>> baidu.com
dig这个程序的版本号和要查询的域名
;; global options: +cmd
表示可以在命令后面加选项
;; Got answer:
以下是获取信息的内容
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60954
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 5, ADDITIONAL: 6
这个是返回信息的头部:
opcode: 操作码,QUERY,代表是查询操作;
status: 状态,NOERROR,代表没有错误;
id: 编号,60954,16bit数字,在dns协议中,通过编号匹配返回和查询.
flags: 标志,如果出现就表示有标志,如果不出现,就表示为设置标志:
qr query,查询标志,代表是查询操作
rd recursion desired,代表希望进行递归查询操作;
ra recursive available在返回中设置,代表查询的服务器支持递归查询操作;
aa Authoritative Answer权威回复,如果查询结果由管理域名的域名服务器而不是缓存服务器提供的,则
称为权威回复;
QUERY 查询数,1代表一个查询,对应下面QUESTION SECTION的记录数
ANSWER 结果数,4代表有4个结果,对应下面的ANSWER SECTION中的记录数
AUTHORITY 权威域名服务器记录数,5代表该域名有5个权威域名服务器,可供域名解析用。对应
下面AUTHORITY SECTION
ADDITIONAL 格外记录数,6代表有6项格外记录。对应下面 ADDITIONAL SECTION。
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096

这个不知道啥意思

;; QUESTION SECTION:

;baidu.com. IN A

查询部分,从做到右部分意义如下:

1、要查询的域名,这里是baidu.com.,'.'代表根域名,com顶级域名,baidu二级域名

2、class,要查询信息的类别,IN代表类别为IP协议,即Internet。还有其它类别,比如chaos等,由于

现在都是互联网,所以其它基本不用。

3、type,要查询的记录类型,A记录(Address),代表要查询ipv4地址。AAAA记录,代表要查询ipv6地址。

;; ANSWER SECTION:

baidu.com. 211 IN A 123.127.114.144

baidu.com. 211 IN A 111.13.101.208

baidu.com. 211 IN A 220.181.57.217

baidu.com. 211 IN A 180.149.132.47

回应部分,回应都是A记录,A记录从左到右各部分意义:

1、对应的域名,这里是baidu.com.,'.'代表根域名,com顶级域名,baidu二级域名

2、TTL,time ro live,缓存时间,单位秒。76,代表缓存域名服务器,可以在缓存中保存76秒该记录。

3、class,要查询信息的类别,IN代表类别为IP协议,即Internet。还有其它类别,比如chaos等由于现在都是互联网,所以其它基本不用。

4、type,要查询的记录类型,A记录,代表要查询ipv4地址。AAAA记录,代表要查询ipv6地址。

5、域名对应的ip地址。

;; AUTHORITY SECTION:

baidu.com. 52340 IN NS dns.baidu.com.

baidu.com. 52340 IN NS ns3.baidu.com.

baidu.com. 52340 IN NS ns2.baidu.com.

baidu.com. 52340 IN NS ns7.baidu.com.

baidu.com. 52340 IN NS ns4.baidu.com.

权威域名部分,回应都是NS记录(Name Server),NS记录从左到右各部分意义:

1、对应的域名,这里是baidu.com.,'.'代表根域名,com顶级域名,baidu二级域名

2、TTL,time ro live,缓存时间,单位秒。63948,代表缓存域名服务器,可以在缓存中保存63948秒

该记录。
3、class,要查询信息的类别,IN代表类别为IP协议,即Internet。还有其它类别,比如chaos等,由于
现在都是互联网,所以其它基本不用。
4、type,要查询的记录类型,NS,Name Server,NS记录,代表该记录描述了域名对应的权威域名
解析服务器
5、域名对应域名对应的权威域名解析服务器。由于ns2.baidu.com.是baidu.com.的子域名,而解析子
域名,又需要主域名的信息,为了打破这个死循环,需要在下面的额外记录中提供该服务器的ip地址。

;; ADDITIONAL SECTION:

dns.baidu.com.	55285	IN	A	202.108.22.220
ns2.baidu.com.	60827	IN	A	61.135.165.235
ns3.baidu.com.	79196	IN	A	220.181.37.10
ns4.baidu.com.	79196	IN	A	220.181.38.10
ns7.baidu.com.	55194	IN	A	119.75.219.82

额外记录部分,这里都是A记录,A记录从左到右各部分意义:
1、对应的域名,这里是dns.baidu.com.,'.'代表根域名,com顶级域名,baidu二级域名,dns是三级域名。
2、TTL,time ro live,缓存时间,单位秒。13284,代表缓存域名服务器可以在缓存中保存13284秒该记录。
3、class,要查询信息的类别,IN代表类别为IP协议,即Internet。还有其它类别,比如chaos等,由于
现在都是互联网,所以其它基本不用。
4、type,要查询的记录类型,A记录,代表要查询ipv4地址。AAAA记录,代表要查询ipv6地址。
5、域名对应的ip地址。

;; Query time: 2 msec
查询耗时
;; SERVER: 127.0.1.1#53(127.0.1.1)
查询使用的服务器地址和端口,其实就是本地DNS域名服务器
;; WHEN: Sun Dec 27 19:27:16 CST 2015

查询的时间
;; MSG SIZE rcvd: 272
回应的大小。收到(rcve, recieved)276字节。

DNS服务器区域传送(主从复制)

host:测试DNS工具
    host host_name:查看host_name的IP地址
    host IP:查看该IP地址对应的主机名
nslookup:
    -q=type(MX SOA NS等类型)
    server IP:更改查询DNS地址

DNS子域和区域转发:

gooann.net:分成市场部和财务部单独管理

子域:为了方便管理域,父域划分更多的子域,并制定子域中的权威服务器

子域DNS服务器:有自己的区域数据库文件,子域的名称是继承的父域的名称

公司申请的域名:gooann.net:有主从DNS
    申请子域名:市场部:market.gooann.net:有主从DNS
    申请子域名:财务部:fin.gooann.net:有主从DNS

192.168.10.10:父域(gooann.net)主DNS
192.168.10.10:父域(gooann.net)从DNS
192.168.10.30:子域(market.gooann.net)主DNS
192.168.10.40:子域(market.gooann.net)从DNS
192.168.10.50:子域(fin.gooann.net)主DNS
192.168.10.60:子域(fin.gooann.net)从DNS

子域配置:
1:在父域的区域数据库文件中添加记录:

子域名城　　IN　NS　子域DNS服务器名称
子域DNS服务器名称 IN　A IP

区域转发:
forward {only|first}
only:先转发给转发器,如果转发器解析不了,则放弃
firest:先转发给转发器,如果转发器解析不了,则进行递归查询

options区域中添加:全局转发
forward only|first;
forwarders { ip; };

zone "zone_name"　　IN {　//仅转发某个区域
type　forward;
forward only|first;
forwarders { ip; };
};

allow-recursion { list_ip;}　{ 192.168.10.0/24; 1.0.0.0/8; .....}
allow-transfer { list_ip;}
allow-query { list_ip;}
list:IP地址
none:
any:所有的

/etc/named.conf:ACL
acl acl_name {
192.168.10.0/24;
172.168.0.0/18;
10.0.0.0/8;
};

options {
directory "/var/named";
allow-recursion　　{ acl_name; };

};

泛域名解析:在区域数据库文件中添加A记录
* IN A ip //不常用

智能DNS:BIND+VIEW
CDN:内容分发网络(根据解析地址的不同,为客户端分配相应的内容资源)
split brain:脑裂
web缓存对象:缓存静态内容(动态web对象可以通过技术映射成静态内容)

VIEW:视图
所有的zone都必须放在VIEW中;跟区域只需要定义在能够递归客户端的view中

BIND DLZ:dynamic load zone:动态加载区域,把DNS区域存放到数据库中

bind日志功能:
在默认情况下,BIND9 把日志消息写到 /var/log/messages文件中,主要就是启动,关闭的日志记录和一些严重错误的消息;而将调试日志信息写入 BIND 服务器工作目录中的 named.run 文件。
BIND 9 的日志是可以灵活配置的,要详细记录服务器的运行状况,要在配置文件 named.conf 中使用 logging 语句来定制自己所需要的日志记录。

BIND 日志的常用术语
在讲述 logging 语句的语法之前,先要熟悉一些常用术语
channel(通道) 日志输出方式,如:syslog(记录到/var/log/messages中)、文本文件、标准错误输出或 /dev/null
category(类别) 日志的消息类别,如:查询消息或动态更新消息等
module(模块) 产生消息的来源模块名称
facility(设备) syslog 设备名
severity(严重性) 消息的严重性等级

logging 语句的语法为:

```
logging {
channel channel_name {                  // 定义通道
file log_file [versions number | unlimited] [size sizespec]; | syslog optional_facility; |
null; | stderr;      // 定义输出方式
severity log_severity;            // 定义消息严重性
[print-time boolean;]              // 是否在消息中添加时间前缀,仅用于 file 日志
[print-severity boolean;]         // 是否在消息中添加消息严重性前缀
[print-category boolean;]         // 是否在消息中添加消息类别名前缀
};
category category_name {                // 定义类别
channel_name;
……
};
};
```

配置日志时,首先要定义通道,然后将不同的日志类别的数据指派到指定的通道上输出。

BIND 9 的默认配置是:

```
logging {
// 由于使用了默认通道,所以没有通道定义部分
category “default” { “default_syslog”; “default_debug”; };
};
```

channel 语句

channel 语句用于定义通道。指定应该向哪里发送日志数据,需要在以下四种之间则其一:

file : 输出到纯文本文件

log_file 指定一个文件名

version 指定允许同时存在多少个版本的该文件,比如指定 3 个版本(version 3),就会保存 query.log、query.log0、query.log1 和query.log2。

size 指定文件大小的上限,如果只设定了size 而没有设定 version,当文件达到指定的文件大小上限时,服务器停止写入该文件。如果设定了version,服务器会进行循环,如把 log_file 变成 log_file.log1,log_file.log1 变成 log_file.log2 等,然后建立一个新的 log_file.log 进行写入。

syslog optional_facility:输出到 syslog,其中optional_facility是syslog的设备名,通常为以下几个:

daemon:local0 到 local7

null:输出到空设备
stderr:输出到标准错误输出,默认为屏幕

severity 语句用于指定消息的严重性等级,log_severity 的取值为(照严重性递减的顺序);
critical
error
warning
notice
info
debug [level]
dynamic 是一个特殊的值,它匹配服务器当前的调试级别
定义了某个严重性级别后,系统会记录包括该级别以及比该级别更严重的级别的所有消息。
比如定义级别为 error,则会记录 critical 和error 两个级别的信息。
对于系统管理员来说,一般记录到 info 级别就可以了。

BIND 9 预制了如下四个默认通道;

```
channel “default_syslog” {
  syslog daemon;     // 发送给 syslog 的 daemon 设备
  severity info;     // 只发送此 info 及其更高优先级的信息
  };
channel “default_debug” { // 只有当服务器的 debug 级别非 0 时,才产生输出。
  file “named.run”; // 写入工作目录下的 named.run 文件
  severity dynamic; // 按照服务器当前的debug 级别记录日志
  };
channel “default_stderr” {
  stderr;            // 写到stderr
  severity info;    // 只发送此 info 及其更高优先级的信息
  };
channel “null” {
  null;              // 丢弃所有发到此通道的信息
  };
```

category 语句是指定哪一种类别的信息使用哪个或者哪几个已经定义了的通道输出。
BIND 9 中可用的类别名(category_name)有:

client 处理客户端请求。
config 配置文件分析和处理。
database 同BIND内部数据库相关的消息,用来存储区数据和缓存记录。
default 匹配所有未明确指定通道的类别。
dnssec 处理 DNSSEC 签名的响应。
general 包括所有未明确分类的 BIND 消息。
lame-servers 发现错误授权,即残缺服务器。
network 网络操作。
notify 区更新通知消息。
queries 查询日志
resolver 名字解析,包括对来自解析器的递归查询信息。
security 批准/非批准的请求。
update 动态更新事件。
xfer-in 从远程名字服务器到本地名字服务器的区传送。
xfer-out 从本地名字服务器到远程名字服务器的区传送。

例如要记录查询消息,可以在 named.conf 中添加如下配置:

```
logging {
    channel query_log {
            file “/var/log/bind/query.log” versions 3 size 20m;
            severity info;
            print-time yes;
            print-category yes;
        };

channel xfer_log {
file “/var/log/bind/xfer.log” versions 3 size 20m;
severity debug 3;
print-time yes;
print-category yes;
};
            category queries { query_log; };
            category xfer-out { xfer_log; };
        };
};
```

这样服务器会在工作目录(directory 语句所指定的目录,Ubuntu 为：/var/cache/bind)下创建 query.log 文件,并把运行过程产生的 queries 消息写如到此文件中。

rndc:remote name domain controller:远程域名服务器控制器
/etc/rndc.conf:rndc配置文件
/etc/rndc.key:rndc秘钥文件

192.168.10.20远程控制192.168.10.10:
192.168.10.10:配置
rndc-confgen -a //生成秘钥文件
rndc-confgen > /etc/rndc.conf //生成rndc配置文件

rndc.conf解析:

```
# Start of rndc.conf          //配置文件开始
        key "rndc-key" {     //定义使用key名称,如rndc-key
        algorithm hmac-md5;    //加密算法
        secret "8pycNW/k9tI+4EfR7JHs2g==";   //生成密文
    };

    options {
        default-key "rndc-key";     //使用的秘钥,秘钥名称要和key定义的名字相同
        default-server 127.0.0.1;  //监听哪个IP地址
        default-port 953;   //rndc默认端口号,为tcp的953端口
    };
# End of rndc.conf   //配置文件结束
```

添加到named.conf文件中的内容解析:

```
  key "rndc-key" {
    algorithm hmac-md5;
    secret "8pycNW/k9tI+4EfR7JHs2g==";
        };

    controls {
    inet 127.0.0.1 port 953    //定义监听的IP地址和远程控制的端口
```

allow { 127.0.0.1; } keys { "rndc-key"; }; //定义允许远程控制的主机和key文件
};

queryperf:对DNS服务器进行压力测试
建立测试文件:格式 www.gooann.net A
-d:指定测试文件
-s:指定测试DNS服务器地址

dnstop:监控dns
-4:使用IPv4
-R:请求的数据包
-Q:回复的数据包
dnstop -4 NIC_name -R -Q

http协议:超文本传输协议hypertext transfer protocol
超文本:带有超级链接的文本
常用server:httpd nginx IIS等
httpd官网:httpd.apache.org
最新版本:2.4.X版本
MPM:multi processer modules多处理模块:prefork woker event

keepalive:持久连接

APR:apache portable runtime:apache可移植运行平台

安装apache依赖包
yum -y install openssl-devel pcre-devel zlib-devel libtool expat expat-devel
安装apr(Apache Portable Runtime)
tar xvzf apr-1.5.2.tar.gz -C /usr/src
cd /usr/src/apr-1.5.2
./configure --prefix=/usr/local/apr
make
make install

安装apr-util

```
# tar xvzf apr-util-1.5.4.tar.gz -C /usr/src
# cd /usr/src/apr-util-1.5.4
# ./configure --prefix=/usr/local/apr-util --with-apr=/usr/local/apr
# make
# make install
```

编译安装apache2.4.27

```
# tar xvzf httpd-2.4.27.tar.gz -C /usr/src
# cp -rf /usr/src/apr-1.5.2/ /usr/src/httpd-2.4.27/srclib/apr
# cp -rf /usr/src/apr-util-1.5.4/ /usr/src/httpd-2.4.27/srclib/apr-util
# cd /usr/src/httpd-2.4.27
# ./configure --prefix=/usr/local/httpd --enable-so --enable-ssl --enable-cgi --
enable-rewrite --enable-deflate --with-zlib --with-pcre --with-apr=/usr/local/apr --
with-apr-util=/usr/local/apr-util --enable-modules=most --enable-mpms-
shared=all --with-included-apr -enable-proxy --enable-proxy-fcgi
# make
# make install
```

优化执行路径

```
# ln -s /usr/local/httpd/bin/* /usr/bin/
```

添加httpd系统服务

```
[root@www ~]# cp /usr/local/httpd/bin/apachectl /etc/init.d/httpd
[root@www ~]# vi /etc/init.d/httpd
#!/bin/bash
# chkconfig: 35 85 15
# description: Startup script for the Apache HTTP Server
……
[root@www ~]# chkconfig --add httpd
[root@www ~]# chkconfig --list httpd
```

编译安装 Mariadb数据库

安装CentOS扩展源

```
# yum -y install epel-release
```

安装依赖包

yum -y install ncurses-devel readline-devel zlib-devel openssl-devel libxml2-devel libxml2 cracklib-devel libevent libevent-devel pam-devel opencv opencv-devel jemalloc jemalloc-devel libaio libaio-devel

添加mysql用户

useradd -r -M -s /sbin/nologin mysql

解压Mariadb源码包

tar zxf mariadb-10.1.7-linux-glibc_214-x86_64.tar.gz -C /usr/local

切换至解压目录

cd /usr/local

为Mariadb数据库做软链接

ln -s mariadb-10.1.7-linux-x86_64/ mysql

为Mariadb准备配置文件

cp /usr/local/mysql/support-files/my-large.cnf /etc/my.cnf

修改Mariadb配置文件

vim /etc/my.cnf:在[mysqld]添加或修改如下内容

datadir=/usr/local/mysql/data

修改Mariadb安装路径的归属

chown -R mysql:mysql /usr/local/mysql

初始化数据库

```
# /usr/local/mysql/scripts/mysql_install_db \
    --basedir=/usr/local/mysql --datadir=/usr/local/mysql/data \
    --user=mysql
```

为Mariadb准备启动脚本

#cp /usr/local/mysql/support-files/mysql.server /etc/init.d/mysqld

为启动脚本添加执行权限

#chmod +x /etc/init.d/mysqld

优化Mariadb服务路径

```
# ln -s /usr/local/mysql/bin/* /usr/bin/
# ln -s /usr/local/mysql/include/* /usr/include/
# ln -s /usr/local/mysql/lib/* /usr/lib
```

把mysqld添加为系统服务并启动

```
# chkconfig mysqld on
# chkconfig --add mysqld
```

启动mysqld服务

service mysqld start

编译安装php-7.0.7:

安装扩展工具libmcrypt //加密库文件

tar xvzf libmcrypt-2.5.8.tar.gz -C /usr/src/

cd /usr/src/libmcrypt-2.5.8

./configure && make -j 4 && make install

ln -s /usr/local/lib/libmcrypt.* /usr/lib/

安装扩展工具mhash: //加密算法文件

tar xvzf mhash-0.9.9.9.tar.gz -C /usr/src/

cd /usr/src/mhash-0.9.9.9

./configure && make -j 4 && make install

ln -s /usr/local/lib/libmhash* /usr/lib/

安装mcrypt: //加密文件

cd /usr/src/mcrypt-2.6.8/

export LD_LIBRARY_PATH=/usr/local/lib:$LD_LIBRARY_PATH

./configure && make -j 4 && make install

安装PHP7.0

tar xf php-7.0.7.tar.gz

cd php-7.0.7.tar.gz

./configure --prefix=/usr/local/php7 --with-openssl --with-mysqli=/usr/local/mysql/bin/mysql_config --enable-mbstring --with-freetype-dir --with-jpeg-dir --with-png-dir --with-zlib --with-libxml-dir=/usr --enable-xml --enable-sockets --with-apxs2=/usr/local/httpd/bin/apxs --with-mcrypt --with-config-file-path=/usr/local/php7 --with-config-file-scan-dir=/usr/local/php7/php.d --with-bz2 --enable-maintainer-zts --enable-mysqlnd --with-pdo-mysql=mysqlnd --with-mysqli=mysqlnd --with-gd --with-curl --with-gd

#make && make install

PHP-7.0.7源码包中两个主配文件模板:

php.ini-development:开发测试环境

php.ini-production:生产环境

#cp php.ini-production /usr/local/php/php.ini //准备主配文件

添加apache对php支持: 修改httpd主配文件

vim /usr/local/httpd/conf/httpd.conf:修改或添加如下内容:

LoadModule php5_module modules/libphp5.so(默认已经添加,提
供apache和php连接的模块)
DirectoryIndex index.php (添加apache支持的php文档)
AddType application/x-httpd-php .php
#service httpd restart //重启httpd服务

LAMP:Linux Apache MySQL/mariadb PHP/Perl/Python

日志功能:
1.系统登录方面的信息
2.解决系统错误方面的问题
3.解决网络服务方面的问题
4.解决应用服务方面的问题

Linux日志通常放在/var/log目录中

系统登录文件:
/var/log/boot.log:开机的时候系统内核侦测硬件与启动硬件,存放内核支持的启动功能
/var/log/cron:任务计划相关的信息,如执行错误,/etc/crontab有没有语法错误等
/var/log/dmesg:侦测系统开机时内核侦测过程中产生的信息
/var/log/lastlog:系统中所有账户最后登录的相关信息,用lastlog查看该文件信息
/var/log/maillog或/var/log/mail/*:记录邮件的往来信息,主要记录邮件服务器信息
/var/log/messages:记录操作系统中发生的信息,包括错误信息等
/var/log/secure:只要涉及到需要用户名和密码软件,登录时都记录在该文件,如ssh等
/var/log/wtmp,/var/log/faillog:记录正常登录系统的账号

Linux中关于日志的服务和程序:
rsyslog.service:主要是登录系统和网络等服务的信息
logrotate:对系统或登录日志进行轮替
systemd-journald.service:日志接收者,由systemd提供管理的

Linux日志格式:

时间日期　主机名　服务字段　日志详细解释

系统日志服务:rsyslog.service,主配文件/etc/rsyslog.conf

1） #### MODULES #### --->启动/sbin/rsyslogd要加载的模块
格式： $ModLoad module-name #注释
$ModLoad 是关键字（/sbin/rsyslogd程序中定义的变量名）
module-name：出现在/lib64/rsyslog/ 中的模块名称，记住不带.so
2） #### GLOBAL DIRECTIVES #### //定义日志格式
3) #### RULES ####
格式： facility.priority　　target
　　facility:表示产生日志的设备
　　priority:表示产生日志的级别
　　target:表示日志存放的地方

facility:表示产生日志的设备　用# man 3 syslog查看系统的facility:
　LOG_AUTH:认证授权相关机制,如login,ssh,su等
　LOG_AUTHPRIV:和AUTH类似,服务相关的认证信息和PAM模块的认证信息,比如ftp
　LOG_CRON:任务计划相关的信息,比如cron和at
　LOG_DAEMON:系统服务产生的信息,比如systemd
　LOG_FTP:ftp服务器产生的日志信息
　LOG_KERN:内核产生的信息,大部分都是硬件侦测及核心功能的启用信息
　LOG_LOCAL0-7:保留给本地用户使用的一些登录文件信息,通常与终端有关
　LOG_LPR:打印相关信息
　LOG_MAIL:邮件收发相关的信息
　LOG_NEWS:与新闻组服务器有关的信息
　LOG_SYSLOG:有syslog相关协议产生的信息,/sbin/syslogd程序本身产生的信息
　LOG_USER:使用者产生的信息
　LOG_UUCP:Unix to Unix Copy Protocol:用户Unix系统间数据交换

priority:表示产生日志的级别
　7:debug:用户调试时产生日志信息
　6:info:基本的信息说明
　5:notice:正常信息,但是应该被注意到
　4:warn:警告信息,可能会产生问题

3:err:错误信息,如配置的设置造成服务无法启动等信息
2:crit:critical比error还要验证的信息
1:alect:已经很严重的信息
0:emery(panic):指的是系统已经几乎死机的状态

target:表示日志存放的地方:
1.文件的绝对路径,如/var/log
2.打印机或其他设备:比如/dev/lp0或/dev/console等
3.使用者的名字:
4.发送给远程主机
5.*代表发送给当前主机登录的所有用户

facility和priority连接符号:
.xxx:如news.crit,代表比.后面级别xxx还要严重的级别都被记录,包括当前定义的级别
.=xxx:表示等于xxx级别的信息,只记录xxx级别
.!=xxx:表示不包含xxx级别

CentOS5:syslog
CentOS6和CentOS7:rsyslog,rsyslod.service为了兼容systemd控制

把192.168.10.20上的日志转发到192.168.10.10:
修改192.168.10.10主机上的/etc/rsyslog.conf,启用TCP或UDP传输
修改192.168.10.20主机上的/etc/rsyslog.conf,在rules部分添加:
. @@192.168.10.10 //表示用TCP方式传输,使用514端口
. @192.168.10.10 //表示用UDP方式传输,使用514端口

日志轮替:/etc/logrotate.conf
/etc/logrotate.conf文件解释:
weekly:代表每周轮替一次
rotate 4:保存四份轮替文件
create:轮替以后创建新的文件来存储日志
dateext:旧的日志文件加上日志进行保存
#compress:被更改的日志文件是否被压缩
include /etc/logrotate.d:RPM包删除的日志轮替信息保存位置

logrotate:手动实现日志轮替

-v:显示轮替信息

-f:强制轮替

systemd-journald.service:

以前rsyslogd必须要开机以后才能运行,启动rsyslogd服务之前,系统内核也会产生日志信息,所有rsyslogd产生的启动信息,要在内核日志信息之后才被送给rsyslogd程序执行

CentOS7:systemd-journald.service是由systemd程序管理,systemd是第一个进行.主动调用systemd-journald.service服务,systemd-journald.service日志保存在内存中,以文件的形式存在,放在/run/log/中

journalctl:查看systemd-journald.service产生的信息:

journalctl:默认会显示所有的log内容

-n num:显示最近的num行信息

-r:反向输出

-p --priority=RANGE:按照日志等级进行排序

-f:动态显示日志内容

--since DATE --until DATE:按照开始和结束时间查看日志内容

-k:显示内核相关的日志信息

_COMM=bash:显示与bash相关的信息

logger:把日志存储到某个日志文件

语法:# logger [-p 服务名称.级别] [信息]