8-20

1. 缓存DNS , 缓存解析结果,

存储用内存做的,重启服务器后记录消失(一般重启几率较小)

环境准备:

虚拟机A : 192.168.4.7/24 (真DNS服务器)

B : 192.168.4.207/24 (缓存DNS)

C : 192.168.4.10/24 (服务端)

路由器上可以做访问控制 , 控制上网 实际环境中,路由器和交换机的形状是方的

缓存DNS没有地址库文件,没有解析记录

A --------------交换机 --------------- 路由器 ---------------- DNS ------------------ Web

|

|

缓存DNS

1.构建缓存DNS 虚拟机B

[root@pa207 ~]# vim /etc/named.conf

options {

directory "/var/named";

forwarders { 192.168.4.7; }; #转发给真DNS服务器

};

[root@pa207 ~]# !sys

客户端测试 虚拟机C

[root@svr10 ~]# echo nameserver 192.168.4.207 > /etc/resolv.conf

[root@svr10 ~]# cat /etc/resolv.conf

[root@svr10 ~]# nslookup www.sina.com

Server: 192.168.4.207

Address: 192.168.4.207#53

Non-authoritative answer:

Name: www.sina.com

Address: 40.50.60.70 (原设置)

8-21

一.Split分离解析(视图解析)

-能够区分客户机的来源地址 , 为不同类别的客户机提供不同的解析结果(IP地址)

-为不同客户端解析同一个域名,得到的解析结果不同

-为客户端提供网络中最近的服务器(就近原则)

BIND的view视图

根据源地址集合将客户机分类

-不同客户机获得不同结果(待遇有差别)

-分类要合理,所有客户端都要找到自己的分类

-由上到下进行依次匹配,匹配即停止

· view "视图1" {

· match-clients { 客户端地址1; .. .. ; };         //匹配第1类客户机地址

· zone "目标域名" IN {                             //同一个DNS区域

· type master;

· file "地址库1";                             //第1份地址库

· }; };

1.

[root@svr7 ~]# vim /etc/named.conf

options {

directory "/var/named";

};

view "nsd" {

match-clients { 192.168.4.207; 192.168.7.0/24; }; #匹配客户端地址

zone "tedu.cn" IN {

type master;

file "tedu.cn.zone"; #写入解析结果为192.168.4.100

};

};

view "other" {

match-clients { any; }; #匹配客户端地址

zone "tedu.cn" IN {

type master;

file "tedu.cn.zone"; #写入解析结果为1.2.3.4

};

};

2.建立地址库文件tedu.cn.zone与tedu.cn.other 写入不同的解析结果

多区域的分类解析:

Tedu.cn.zone 192.168.4.100

sina.com.zone192.168.4.200

Tedu.cn.other 1.2.3.4

Sina.cn.other 10.20.30.40

acl地址列表(类似于变量)

为大批量的客户机地址建立列表

测试DNS解析命令 : dig

[root@svr7 named]# dig www.sina.com

dig

一.RAID磁盘阵列 廉价冗余磁盘阵列

-通过硬件/软件技术,将多个较小/低俗的磁盘整合成一个大磁盘

RAID 0 : 至少由两块磁盘组成 (较少使用)

同一个文档分散存在不同磁盘

并行写入以提高效率 (挂了一个另外一个只输出一半数据)

RAID 1 : 镜像模式 至少2块磁盘组成

多分拷贝提高可靠性,效率无提升 备份

RAID 5 : 高性价比模式 至少3块盘组成

可随机坏一块盘,两块以上服务器崩溃

若A盘挂了, C通过推理A1的数据,与B输出AB

RAID 6 : 高性价比/可靠模式 至少2块磁盘组成

相当于扩张的RAID5阵列,提供2份独立校验方案

至少需要两块磁盘的容量来存放校验数据

3块磁盘同时坏服务器就崩溃

RAID 10 = RAID 1+0

至少需要4块磁盘组成 可允许坏两块

并行存取提高效率 镜像写入提高可靠性

例如: 有A B C D 4块盘

0+1 : AB两块做RAID 0 为E CD 两块做RAID 0 为F

EF两部分做RAID 1

硬RAID : 有RAID控制卡管理阵列

-主板 --->阵列卡 ---> 磁盘 ---> 操作系统 --->数据

软RAID : 由操作系统来管理阵列

- 主板 ---> 磁盘 ---> 操作系统 ---> RAID软件 ---> 数据

二.进程管理

程序 : 静态代码 占用磁盘空间

进程 : 动态执行的代码 占用CPU与内存的资源 (提供功能) (由程序提供的)

父进程与子进程 : 树形结构 若父进程衰亡,子进程也关闭 还有僵尸进程 孤儿进程

*进程编号 : PID*

查看进程 ps

pstree -- Processes Tree

systemd : 所有进程的父进程 上帝进程 会接管孤儿进程

-a : 显示完整的命令行

-p : 列出对应PID编号

-用户 : 显示用户的进程

ps aus : 列出正在运行的所有进程

ps elf : 列出正在运行的所有进程

PPID : 父进程的PID号

PRI/NI : 进程优先级 , 数值越小优先级越高

top 交互式工具

-d 刷新秒数 -? 查看帮助(列出可用的按键指令)

-U 用户名 -P ,M 根据%CPU , %MEN降序排列

-T 根据进程消耗的TIME降序排列

-k 杀死指定的进程

-q 退出top程序

Load average : 1 5 15

total : 进程数

KiB Swap : 交换空间

检索进程

pgrep [选项]

-l : 输出进程名,而不仅仅是PID

-U : 检索指定用户的进程

-x : 精确匹配完整的进程名

控制进程

进程的后台调度

[root@svr7 ~]# sleep 1000 & #正在运行放入后台

[root@svr7 ~]# sleep 800

^Z #Ctrl+z 暂停放入后台

[root@svr7 ~]# bg 2 #激活后台编号为2的进程

[root@svr7 ~]# jobs #查看后台进程信息

[root@svr7 ~]# fg 2 #将后台编号为2的进程恢复到前提

^C #按ctrl+c 结束进程的运行

[root@svr7 ~]# jobs

[root@svr7 ~]# fg 1 #将后台编号为1的进程恢复到前提

sleep 1000

^C

[root@svr7 ~]# jobs

查杀进程

Kill killall

-9 : 强制杀 (少用)

pkill : 查找条件 (模糊查杀,包含相关内容也得关)

(既rm - rf 之后)

[root@svr7 ~]# jobs -l #查看详细进程

强制踢出一个用户( 杀死一个用户开启的所有的进程) (所有数据丢失)

[root@svr7 ~]# killall -9 -u lisi

1. 日志管理
2. 日志的功能

访问量汇总 用户访问量高低

ELK 日志分析平台

-记录系统.程序运行中发生的各种事件

-通过查看日志,了解及排除故障

-信息安全控制的”依据”

内核及系统日志:

/var/log/messages 专门记录内核消息 , 各种服务的公共消息

/var/log/dmesg 系统启动过程的各种消息

/var/log/cron cron计划任务相关的消息

/var/log/maillog 邮件收发

/var/log/secure 访问限制相关安全信息

用户日志:

查看文本日志消息

文本浏览,检索命令 :tail tailf grep

格式化过滤工具 : awk sed

日志分析平台 : elk

用户登录分析

who

pts : 图形命令行终端

w : 对字体大小有要求

last : 最近成功登录信息 last -2最近登录成功两条信息

lastb : 最近登录失败信息

[root@svr7 ~]# who

root pts/0 2019-08-21 16:09 (192.168.4.254)

Linux内核定义的事件紧急程度:

日志消息的优先级: 0~7 共8种优先级别

其数值越小,表示对应事件越紧急/重要

0 紧急

1 警告

2 严重

3 错误

4 提醒

提供由 System Journalctl 服务收集的日志

Systemd

RHEL 6 运行级别

0 : 关机

1 : 单用户模式(基本功能的实现,破密)

切换运行级别 : init 5 (RHEL6) =

RHEL 7 01246

