虚拟机A:构建DNS服务器,实现www.sina.com的解析结果为10.20.30.40

1.修改主配置文件

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

options {

directory "/var/named";

recursion no;

};

zone "tedu.cn" IN {

type master;

file "tedu.cn.zone";

};

zone "qq.com" IN {

type master;

file "qq.com.zone";

};

zone "sina.com" IN {

type master;

file "sina.com.zone";

};

2.建立地址库文件

[root@svr7 named]# cd /var/named/

[root@svr7 named]# cp -p qq.com.zone sina.com.zone

[root@svr7 named]# vim /var/named/sina.com.zone

sina.com. NS svr7

svr7 A 192.168.4.7

www A 10.20.30.40

[root@svr7 named]# systemclt restart named

####################################################

Split分离解析(视图解析)

什么是分离解析

• 当收到客户机的DNS查询请求的时候

– 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

– 不同客户端解析同一个域名,解析结果不同

– 作用:为客户端就近提供服务器

BIND的view视图

• 根据源地址集合将客户机分类

由上到下匹配及停止,所有的客户端都要找到自己的分类

所有的zone都必须在view中

view "nsd" {

match-clients { 192.168.1.1; }; #匹配客户机来源地址

zone "12306.cn" IN {

...... 地址库文件12306.zone;

}; };

view "abc" {

match-clients { 192.168.2.1; };

zone "12306.cn" IN {

...... 地址库文件12306.abc;

}; };

view "other" {

match-clients { any; };

zone "12306.cn" IN {

...... 地址库文件12306.other;

}; };

###################################################

案例需求及要点

• 环境及需求

– 权威DNS:svr7.tedu.cn 192.168.4.7

– 负责区域:tedu.cn

– A记录分离解析 —— 以 www.tedu.cn 为例

客户机来自 解析结果

192.168.4.207-----> 192.168.4.100

其他地址 -----> 1.2.3.4

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

options {

directory "/var/named";

};

view "nsd" { #分类名称

match-clients { 192.168.4.207; }; #匹配客户机来源地址

zone "tedu.cn" IN {

type master;

file "tedu.cn.zone";---->解析结果写192.168.4.100

};

};

view "other" {

match-clients { any; };

zone "tedu.cn" IN {

type master;

file "tedu.cn.other";---->解析结果写1.2.3.4

};

};

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

#############################################################

多个域名的分离解析,每一个view中zone个数保持一致

– A记录分离解析 —— 以 www.tedu.cn 为例

– A记录分离解析 —— 以 www.sina.com 为例

客户机来自 解析结果

192.168.4.207-----> 192.168.4.100

其他地址 -----> 1.2.3.4

view "nsd" {

match-clients { 192.168.4.207; };

zone "tedu.cn" IN {

type master;

file "tedu.cn.zone"; -------> 解析结果为192.168.4.100

};

zone "sina.com" IN {

type master;

file "sina.com.zone"; -------> 解析结果为192.168.4.100

};

};

view "other" {

match-clients { any; };

zone "tedu.cn" IN {

type master;

file "tedu.cn.other"; -------> 解析结果为1.2.3.4

};

zone "sina.com" IN {

type master;

file "sina.com.other"; -------> 解析结果为1.2.3.4

};

};

#####################################################

扩展:

– A记录分离解析 —— 以 www.tedu.cn 为例

– A记录分离解析 —— 以 www.sina.com 为例

客户机来自 解析结果

192.168.4.207---www.tedu.cn--> 192.168.4.100

192.168.4.7---www.sina.com--> 192.168.4.200

其他地址 --www.tedu.cn---> 1.2.3.4

其他地址 --www.sina.com---> 1.2.3.4

view "nsd" {

match-clients { 192.168.4.207; };

zone "tedu.cn" IN {

type master;

file "tedu.cn.zone";

};

zone "sina.com" IN {

type master;

file "sina.com.other";

};

};

view "abc" {

match-clients { 192.168.4.7; };

zone "tedu.cn" IN {

type master;

file "tedu.cn.other";

};

zone "sina.com" IN {

type master;

file "sina.com.zone";

};

};

view "other" {

match-clients { any; };

zone "tedu.cn" IN {

type master;

file "tedu.cn.other";

};

zone "sina.com" IN {

type master;

file "sina.com.other";

};

};

#####################################################

acl地址列表,类似于变量作用

acl "test" { 192.168.4.207; 192.168.1.1; 192.168.2.1; 192.168.3.1; 192.168.7.0/24; };

view "nsd" {

match-clients { test; };

zone "tedu.cn" IN {

type master;

file "tedu.cn.zone";

};

zone "sina.com" IN {

type master;

file "sina.com.other";

};

};

view "abc" {

match-clients { test; };

zone "tedu.cn" IN {

type master;

file "tedu.cn.zone";

};

zone "sina.com" IN {

type master;

file "sina.com.other";

};

};

####################################################

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/低速的磁盘整合成一个大磁盘

– 阵列的价值:提升I/O效率、硬件级别的数据冗余

– 不同RAID级别的功能、特性各不相同

• RAID 0,条带模式

– 同一个文档分散存放在不同磁盘

– 并行写入以提高效率

– 至少需要两块磁盘组成

• RAID 1,镜像模式

– 一个文档复制成多份,分别写入不同磁盘

– 多份拷贝提高可靠性,效率无提升

– 至少需要两块磁盘组成

• RAID5,高性价比模式

– 相当于RAID0和RAID1的折中方案

– 需要至少一块磁盘的容量来存放校验数据

– 至少需要三块磁盘组成

• RAID6,高性价比/可靠模式

– 相当于扩展的RAID5阵列,提供2份独立校验方案

– 需要至少两块磁盘的容量来存放校验数据

– 至少需要四块磁盘组成

• RAID 0+1/RAID 1+0

– 整合RAID 0、RAID 1的优势

– 并行存取提高效率、镜像写入提高可靠性

– 至少需要四块磁盘组成

######################################################

进程管理

程序: 静态的代码 占用磁盘空间

进程: 动态执行的代码 占用CPU 内存

父进程/子进程 树型结构 僵尸进程 孤儿进程

进程唯一标识: PID 进程的编号

systemd:上帝进程,所有进程的父进程

查看进程树

• pstree — Processes Tree 15:15上课

– 格式:pstree [选项] [PID或用户名]

• 常用命令选项

– -a:显示完整的命令行

– -p:列出对应PID编号

[root@svr7 /]# pstree

[root@svr7 /]# useradd lisi

[root@svr7 /]# pstree lisi

bash───vim

[root@svr7 /]# pstree -a lisi

bash

└─vim a.txt

[root@svr7 /]# pstree -ap lisi

• ps aux 操作,信息全面

– 列出正在运行的所有进程

用户 进程ID %CPU %内存 虚拟内存 固定内存 终端 状态 起始时间 CPU时间 程序指令

• ps -elf 操作,有进程的父进程的PID值

– 列出正在运行的所有进程

PPID:父进程的PID号

统计正在运行的进程有多少?

[root@svr7 /]# ps aux

[root@svr7 /]# ps aux | wc -l

[root@svr7 /]# ps -elf | wc -l

进程动态排名

• top 交互式工具

– 格式:top [-d 刷新秒数] [-U 用户名]

[root@svr7 /]# top -d 1

按P(大写)可以进行CPU的排序

按M(大写)可以进行内存的排序

load average: 0.00, 0.01, 0.05 #CPU负载量 1分 5分钟 15 分钟

检索进程

• pgrep — Process Grep

– 用途:pgrep [选项]... 查询条件

• 常用命令选项

– -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

[root@svr7 /]# pgrep -l a #检索所有进程名带a的

[root@svr7 /]# pgrep -lU lisi #检索lisi的所有进程

[root@svr7 /]# pstree -ap lisi #检索lisi的所有进程

进程的前后台调度

• 后台启动,正在运行放入后台

– 在命令行末尾添加“&”符号,不占用当前终端

• Ctrl + z 组合键

– 挂起当前进程(暂停并转入后台)

• jobs 命令

– 查看后台任务列表

• fg 命令

– 将后台任务恢复到前台运行

• bg 命令

– 激活后台被挂起的任务

[root@svr7 /]# sleep 9000 & #正在运行放入后台

[root@svr7 /]# jobs #查看后台运行的进程

[root@svr7 /]# sleep 8000

^Z #输入Ctrl+z 暂停放入后台

[2]+ 已停止 sleep 8000

[root@svr7 /]# jobs

[root@svr7 /]# bg 2 #将后台编号为2的进程继续运行

[root@svr7 /]# jobs

[root@svr7 /]# fg 1 #将后台编号为1的进程恢复到前台

sleep 9000

^C #输入Ctrl+C 结束

[root@svr7 /]# jobs

[root@svr7 /]# fg 2 #将后台编号为2的进程恢复到前台

sleep 8000

^C #输入Ctrl+C 结束

[root@svr7 /]# jobs

###################################################

杀死进程

• 干掉进程的不同方法

– Ctrl+c 组合键,中断当前命令程序

– kill [-9] PID... 、kill [-9] %后台任务编号

– killall [-9] 进程名...

– pkill 查找条件

[root@svr7 /]# sleep 1000 &

[root@svr7 /]# jobs -l

[root@svr7 /]# kill 4013 #按照PID进行杀死

[root@svr7 /]# sleep 1000 &

[root@svr7 /]# jobs -l

[root@svr7 /]# kill -9 4015 #按照PID强制杀死

[root@svr7 /]# jobs -l

[root@svr7 /]# sleep 1000 &

[root@svr7 /]# sleep 1000 &

[root@svr7 /]# sleep 1000 &

[root@svr7 /]# jobs

[root@svr7 /]# killall sleep #杀死所有sleep进程

杀死一个用户开启的所有进程(强制踢出一个用户)

[root@svr7 /]# killall -9 -u 用户名

#################################################

日志管理

日志的功能

• 系统和程序的“日记本”

– 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

• 由系统服务rsyslog统一记录/管理

– 日志消息采用文本格式

– 主要记录事件发生的时间、主机、进程、内容

• 常见的日志文件

/var/log/messages 记录内核消息、各种服务的公共消息

/var/log/dmesg 记录系统启动过程的各种消息

/var/log/cron 记录与cron计划任务相关的消息

/var/log/maillog 记录邮件收发相关的消息

/var/log/secure 记录与访问限制相关的安全消息

日志分析

• 通用分析工具

– tail、tailf、less、grep等文本浏览/检索命令

– awk、sed等格式化过滤工具

tailf:实时跟踪日志消息

[root@svr7 /]# echo 123 > /opt/1.txt

[root@svr7 /]# cat /opt/1.txt

123

[root@svr7 /]# tailf /opt/1.txt

123

################################################

用户登录分析

• users、who、w 命令

– 查看已登录的用户信息,详细度不同

pts:图形命令行终端

• last、lastb 命令

– 查看最近登录成功/失败的用户信息

[root@svr7 /]# last -2 #最近登录的2条成功信息

[root@svr7 /]# lastb -2 #最近登录的2条失败信息

[root@svr7 /]# users #查看已登录的用户信息

[root@svr7 /]# who #查看已登录的用户信息

[root@svr7 /]# w #查看已登录的用户信息

###################################################

日志消息的优先级

• Linux内核定义的事件紧急程度

– 分为 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 EMERG（紧急） 会导致主机系统不可用的情况

1 ALERT（警告） 必须马上采取措施解决的问题

2 CRIT（严重） 比较严重的情况

3 ERR（错误） 运行出现错误

4 WARNING（提醒） 可能会影响系统功能的事件

5 NOTICE（注意） 不会影响系统但值得注意

6 INFO（信息） 一般信息

7 DEBUG（调试） 程序或系统调试信息等

使用journalctl工具

• 提取由 systemd-journal 服务搜集的日志

– 主要包括内核/系统日志、服务日志

• 常见用法

– journalctl | grep 关键词

– journalctl -u 服务名 [-p 优先级]

– journalctl -n 消息条数

– journalctl --since="yyyy-mm-dd HH:MM:SS" --

until="yyyy-mm-dd HH:MM:SS"

##################################################

systemctl控制

• Linux系统和服务管理器

– systemd是内核引导之后加载的第一个初始化进程(PID=1)

– 负责掌控整个Linux的运行/服务资源组合

systemd

• 一个更高效的系统&服务管理器

– 开机服务并行启动,各系统服务间的精确依赖

– 配置目录:/etc/systemd/system/

– 服务目录:/lib/systemd/system/

– 主要管理工具:systemctl

对于服务的管理

systemctl restart 服务名 #重起服务

systemctl start 服务名 #开启服务

systemctl stop 服务名 #停止服务

systemctl status 服务名 #查看服务当前的状态

systemctl enable 服务名 #设置服务开机自启动

systemctl disable 服务名 #设置服务不开机自启动

systemctl is-enabled 服务名 #查看是否为开机自启

RHEL6 运行级别 不同级别,开启的服务不同

0：关机

1：单用户模式（基本功能的实现，破解Linux密码）

2：多用户字符界面（不支持网络）

3：多用户字符界面（支持网络）服务器默认的运行级别

4：未定义

5：图形界面

6：重起

切换运行级别：init 5

RHEL7 运行模式

字符模式：multi-user.target

图形模式：graphical.target

[student@room9pc01 ~]$ rht-vmctl reset classroom

classroom [OK]

域 classroom 已开始

[student@room9pc01 ~]$ rht-vmctl reset server

server [OK]

域 server 已开始

[root@svr7 /]# ls -l /lib/systemd/system/

当前直接切换到字符模式

]# systemctl isolate multi-user.target = init 3

当前直接切换到图形模式

]# systemctl isolate graphical.target = init 5

查看每次开机默认进入模式

[root@svr7 /]# systemctl get-default

multi-user.target

设置永久策略，每次开机自动进入graphical.target

# systemctl set-default graphical.target

# reboot

#################################################