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

利用真机clone-vm7产生产两台虚拟机，利用root密码123456

1.搭建虚拟机A服务端

要求：配置永久主机名svr7.tedu.cn

配置eth0永久 IP地址：192.168.4.7/24

使用真机Web服务器作为Yum源

2.搭建虚拟机B客户端

要求：配置永久主机名pc207.tedu.cn

配置eth0永久 IP地址：192.168.4.207/24

使用真机Web服务器作为Yum源

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

DNS服务器

– 系统服务: named

– 默认端口: 53

• 主配置文件:/etc/named.conf #设置本机负责解析的域名

• 地址库文件:/var/named/ #主机名与IP地址的对应关系

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

搭建基本的DNS服务器

虚拟机A：

1.安装软件包

[root@svr7 ~]# yum -y install bind-chroot bind

bind-9.9.4-29.el7.x86\_64 //域名服务包

bind-chroot-9.9.4-29.el7.x86\_64 //提供虚拟根支持,牢笼政策

2.修改主配置文件/etc/named.conf

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

options {

directory "/var/named"; #指定地址库文件的位置

};

zone "tedu.cn" IN { #指定本地负责解析的域名

type master; #指定本机为主DNS服务器

file "tedu.cn.zone"; #指定地址库文件名称

};

3.建立地址库文件 /var/named/tedu.cn.zone

# 注意文件权限,named用户有读取

# cp -p 保持源文件属性不变进行复制

# 本文件域名没有点结尾，默认补全本文件负责的域名

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

[root@svr7 named]# cp -p named.localhost tedu.cn.zone

[root@svr7 named]# ls -l tedu.cn.zone

[root@svr7 named]# vim tedu.cn.zone

tedu.cn. NS svr7 #证明DNS服务器主机名

svr7 A 192.168.4.7 #解析DNS服务器主机名对应IP

www A 1.2.3.4

ftp A 1.1.1.1

4.重起named服务

[root@svr7 ~]# systemctl restart named

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

客户端虚拟机B：

测试解析结果

# echo nameserver 192.168.4.7 > /etc/resolv.conf

# cat /etc/resolv.conf

# nslookup www.tedu.cn

# nslookup ftp.tedu.cn

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

检测虚拟机B是否可以，通过虚拟机A的DNS，进行解析www.tedu.cn

1.检测虚拟机B /etc/resolv.conf

2.检测虚拟机A named服务是否开启,将named服务设置为开机自启动

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

DNS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)

– 当不同的客户端解析同一域名，得到的结果不同

– 为客户端提供网络最近的资源

• 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

view "联通" {

match-clients { 来源地址1; }

zone "12306.cn" IN {

...... 地址库1;

}; };

view "铁通" {

match-clients { 来源地址2; }

zone "12306.cn" IN {

...... 地址库2;

}; };

注意：

1.所有的客户端到要匹配分类成功

2.由上及下，匹配及停止

3.所有的zone都要在view中

4.view中所有的zone数量必须保持一致

虚拟机A

1.修改配置文件/etc/named.conf

view "nsd" { #分类名称

match-clients { 192.168.4.207; }; #匹配客户及地址

zone "tedu.cn" IN {

type master;

file "tedu.cn.zone"; #指定读取的地址库文件

};

};

view "other" {

match-clients { any; };

zone "tedu.cn" IN {

type master;

file "tedu.cn.other";

};

};

2.建立两个地址库文件tedu.cn.zone与tedu.cn.other写入不同的解析结果

• 环境及需求

– 权威DNS:svr7.tedu.cn 192.168.4.7

– 负责区域:tedu.cn qq.com

– A记录分离解析 —— 以 www.tedu.cn 为例 以 www.qq.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

options {

directory "/var/named";

};

view "nsd" {

match-clients { 192.168.4.207; };

zone "tedu.cn" IN {

type master;

file "tedu.cn.zone";

};

zone "qq.com" IN {

type master;

file "qq.com.zone";

};

};

view "other" {

match-clients { any; };

zone "tedu.cn" IN {

type master;

file "tedu.cn.other";

};

zone "qq.com" IN {

type master;

file "qq.com.other";

};

};

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

acl地址列表,方便匹配客户端地址

acl "liantong" {

IP地址1; IP地址2; .. ..

网段1; 网段2; .. ..

.. ..

};

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

RAID阵列概述

• 廉价冗余磁盘阵列

– Redundant Arrays of Inexpensive Disks

– 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2块磁盘

• 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2块磁盘

• 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3块磁盘

• 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4块磁盘

• 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4块磁盘

• 硬RAID:由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

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

进程管理

程序：静态的代码，占用磁盘的空间

进程：动态运行的代码，占用CPU与内存

PID：标识进程

父进程与子进程

systemd:所有进程的父进程 PID为1

查看进程树

• pstree — Processes Tree

– 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 lisi

未发现进程。

[root@svr7 /]# pstree lisi

bash───vim

[root@svr7 /]# pstree -p lisi

bash(3356)───vim(3392)

[root@svr7 /]# pstree -ap lisi

bash,3356

└─vim,3392 a.txt

[root@svr7 /]# pstree

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

• ps aux 操作

– 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

[root@svr7 /]# ps aux | wc -l #统计正在运行进程的个数

• ps -elf 操作

– 列出正在运行的所有进程

PPID:父进程的PID号

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

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

[root@svr7 /]# ps aux

[root@svr7 /]# ps -elf

[root@svr7 /]# ps aux | grep lisi

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

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

按大写P可以CPU进行排序

按大写M可以内存进行排序

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

– -t:检索指定终端的进程

– -x:精确匹配完整的进程名

[root@svr7 /]# pgrep -lU lisi

[root@svr7 /]# pgrep -l crond

[root@svr7 /]# pgrep -l log

[root@svr7 /]# pstree -ap lisi

[root@svr7 /]# pgrep -lU lisi -t pts/2

图形下的终端窗口：依次命令为 pts/0 pts/1 pts/2

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

查看进程： pstree(结构显示清晰)

ps aux (输出进程内容详细)

ps -elf(输出进程的PPID)

top(动态排名，有CPU负载信息，可以进行排序)

pgrep(输出简洁，方便运用Shell脚本)

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

进程的前后台调度

进程放入后台：

1. 命令行末尾添加“&”符号

2.Ctrl + z 组合键暂停并转入后台

[root@svr7 /]# sleep 900 & #将进程正在运行放入后台

[root@svr7 /]# jobs #查看后台进程信息

[root@svr7 /]# sleep 800

^Z #按Ctrl+z暂停放入后台

[2]+ 已停止 sleep 800

[root@svr7 /]# jobs #查看后台进程信息

[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 900

^C #按Ctrl+c终止正在运行的进程

[root@svr7 /]# jobs

[root@svr7 /]# fg 2

sleep 800

^C

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 800 &

[root@svr7 /]# sleep 900 &

[root@svr7 /]# sleep 1000 &

[root@svr7 /]# sleep 1100 &

[root@svr7 /]# jobs -l

[root@svr7 /]# kill 5518

[root@svr7 /]# jobs -l

[root@svr7 /]# kill -9 5519

[root@svr7 /]# jobs -l

[root@svr7 /]# killall sleep

[root@svr7 /]# jobs -l

杀死一个用户开启的所有进程（踢出一个用户）

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

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

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”

• 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

tailf实时跟踪日志消息

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

[root@svr7 /]# tailf /opt/123.txt

• users、who、w 命令

– 查看已登录的用户信息,详细度不同

• last、lastb 命令

– 查看最近登录成功/失败的用户信息

[root@svr7 /]# users

[root@svr7 /]# who

[root@svr7 /]# w

[root@svr7 /]# last -2 #最近登陆成功2条记录

[root@svr7 /]# lastb -2 #最近登陆失败2条记录

• 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]

[root@svr7 /]# systemctl restart httpd

[root@svr7 /]# journalctl | grep httpd

[root@svr7 /]# journalctl -u httpd -p 6

– journalctl -n 消息条数

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

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

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

补充：

/etc/hosts:提供域名解析，只为本机提供解析服务

虚拟机A：

[root@svr7 /]# tail -1 /etc/hosts #追加写入

192.168.4.7 www.abc.com www.haha.com

[root@svr7 /]# ping www.abc.com

解析优先级 ：

浏览器输入域名------>本机/etc/hosts--->/etc/resolv.conf--->DNS

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

补充：

PATH：环境变量

PATH：提供命令搜寻路径

(执行命令时，会到PATH变量值路径中寻找改程序，如果找到就执行，如果找不到输出未找到)

[root@svr7 /]# vim /opt/hello.sh

#!/bin/bash

echo hello world

[root@svr7 /]# chmod +x /opt/hello.sh

[root@svr7 /]# /opt/hello.sh

hello world

[root@svr7 /]# hello.sh

bash: hello.sh: 未找到命令...

[root@svr7 /]# echo $PATH

[root@svr7 /]# cp /opt/hello.sh /usr/bin/

[root@svr7 /]# ls /usr/bin/hello.sh

/usr/bin/hello.sh

[root@svr7 /]# hello.sh

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