2020315Lrf

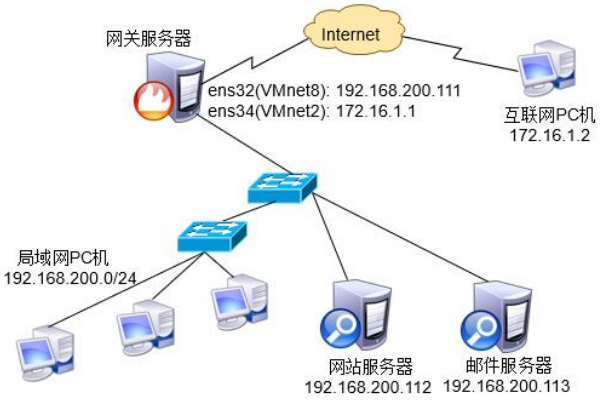
构建DNS域名服务器的分离解析

一、DNS分离解析域名服务器概述

使用同一服务器解析得到的IP地址不同

分析解析案例

一个村买卖西瓜，隔壁村来买西瓜　原理



1.前期环境准备

linux双网卡　！！！　win客户机

```
[root@ns2 ~]# ip a | grep ens
    ault qlen 1000
        inet 192.168.200.104/24 brd 192.168.200.255 scope global noprefixroute ens32
    ault qlen 1000
        inet 172.16.1.1/24 brd 172.16.1.255 scope global noprefixroute ens34
[root@ns2 ~]# iptables -F
[root@ns2 ~]# systemctl stop firewalld
[root@ns2 ~]# setenforce 0
```

```
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=static
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=ens34
DEVICE=ens34
ONBOOT=yes
IPADDR=172.16.1.1
NETMASK=255.255.255.0
```

添加网卡

```
cd /etc/sysconfig/network-script/
cp ifcfg-ens32  ifcfg-ens-34
vim /etc/sysconfig/network-script/ifcfg-ens34
```

2.修改主配置文件

```
[root@ns2 ~]# rpm -q bind
[root@ns2 ~]# vim /etc/named.conf
[root@ns2 ~]# cd /var/named
[root@ns2 named]# ls
data     named.ca     named.localhost   study.zheng.lan
dynamic  named.empty  named.loopback    study.zheng.wan
[root@ns2 named]# vim study.zheng.lan
[root@ns2 named]# cp -p study.zheng.lan study.zheng.wan
[root@ns2 named]# vim study.zheng.wan
[root@ns2 named]# ip a
[root@ns2 named]# systemctl restart named
```

3.测试

```
C:\Users\sofia>nslookup www.study.com
服务器:  UnKnown
Address:  192.168.200.104

名称:    www.study.com
Address:  192.168.200.105


C:\Users\sofia>nslookup mail.study.com
服务器:  UnKnown
Address:  192.168.200.104

名称:    mail.study.com
Address:  192.168.200.103
```

注意修改win7的虚拟机网络连接设置

```
C:\Users\sofia>nslookup www.study.com
服务器:  UnKnown
Address:  172.16.1.1

名称:    www.study.com
Address:  172.16.1.1


C:\Users\sofia>nslookup mail.study.com
服务器:  UnKnown
Address:  172.16.1.1

名称:    mail.study.com
Address:  172.16.1.1
```

二、DNS view的多种应用方式

1.match-clients 直接指定地址

```
[root@ns2 named]# vim /etc/named.conf
    options {
            directory       "/var/named";
    };

    view  "LAN"  {
       match-clients { 192.168.200.0/24; };
       zone "study.com" IN {
            type master;
            file "study.zheng.lan";
        };
    };

    view  "WAN"  {
       match-clients { any; };
       zone "study.com" IN {
            type master;
            file "study.zheng.wan";
        };
    };
```

2.基于acl访问控制列表

```
[root@ns2 named]# vim /etc/named.conf
    options {
            directory       "/var/named";
```

```
};
acl lan { 192.168.200.0/24; };
acl wan { any; };

view  "LAN"  {
   match-clients { lan; };
   zone "study.com" IN {
        type master;
        file "study.zheng.lan";
    };
};

view  "WAN"  {
   match-clients { wan; };
   zone "study.com" IN {
        type master;
        file "study.zheng.wan";
    };
};
[root@ns2 named]# systemctl restart named
```

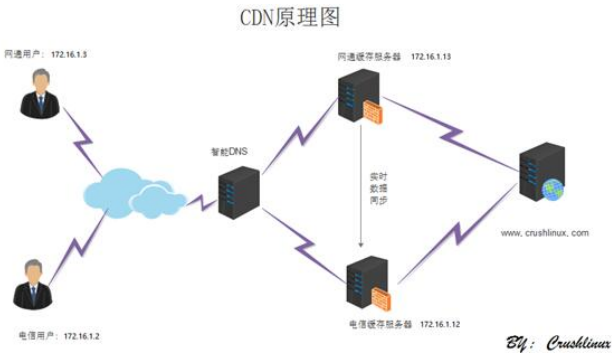
3.基于访问控制文件

```
[root@ns2 named]# vim /var/named/lan.txt
    acl lan {
    192.168.200.0/24;
    192.168.1.0/24;
    192.168.2.0/24;
    };

[root@ns2 named]# vim /var/named/wan.txt
        acl wan {
        any;
        };

[root@ns2 named]# vim /etc/named.conf
    options {
            directory       "/var/named";
    };

    include "/var/named/lan.txt";
    include "/var/named/wan.txt";

    view  "LAN"  {
       match-clients { lan; };
       zone "study.com" IN {
            type master;
            file "study.zheng.lan";
        };
    };

    view  "WAN"  {
```

```
        match-clients { wan; };
        zone "study.com" IN {
             type master;
             file "study.zheng.wan";
         };
    };
[root@ns2 named]# systemctl restart named
```

三、构建智能DNS域名解析服务器--CDN

网络划分　电信　网通　移动




```
[root@ns2 named]# vim /var/named/wangtong.txt
    acl wangtong {
    172.16.1.3;
    };
[root@ns2 named]# vim /var/named/dianxin.txt
    acl  dianxin  {
    172.16.1.2;
    };
[root@ns2 named]# vim /etc/named.conf
    options {
            directory       "/var/named";
    };

    include "/var/named/wangtong.txt";
    include "/var/named/dianxin.txt";

    view  "dianxin"  {
       match-clients { dianxin; };
       zone "study.com" IN {
            type master;
            file "study.zheng.dianxin";
        };
    };

    view  "wangtong"  {
       match-clients { wangtong; };
       zone "study.com" IN {
            type master;
            file "study.zheng.wangtong";
        };
```

```
};

[root@ns2 named]# cd /var/named
[root@ns2 named]# cp study.zheng.wan study.zheng.wangtong
[root@ns2 named]# cp study.zheng.wan study.zheng.dianxin
[root@ns2 named]# vim study.zheng.wangtong
$TTL    86400
@           IN      SOA         study.com.      admin.study.com.    (
                    20100314
                    15M
                    1D
                    3H
                    1W
)
@           IN      NS          ns.study.com.
            IN      MX 10       mail.study.com.
ns          IN      A           172.16.1.1
mail        IN      A           172.16.1.13
www         IN      A           172.16.1.13
[root@ns2 named]# vim study.zheng.dianxin
$TTL    86400
@           IN      SOA         study.com.      admin.study.com.    (
                    20100314
                    15M
                    1D
                    3H
                    1W
)
@           IN      NS          ns.study.com.
            IN      MX 10       mail.study.com.
ns          IN      A           172.16.1.1
mail        IN      A           172.16.1.12
www         IN      A           172.16.1.12
[root@ns2 named]# chgrp named study.zheng.*
[root@ns2 named]# systemctl restart named
```

四、获取不同运营商的IP地址范围

https://github.com/clangcn/everyday-update-cn-isp-ip

根据APNIC的最新IP地址列表及whois信息，每日0点（北京时间）生成的各主要运营商IP地址段。

中国电信 IP地址段：(http://ispip.clang.cn/chinatelecom.html)
中国联通（网通）IP地址段：(http://ispip.clang.cn/unicom_cnc.html)
中国移动 IP地址段：(http://ispip.clang.cn/cmcc.html)
中国铁通 IP地址段：(http://ispip.clang.cn/crtc.html)
中国教育网 IP地址段：(http://ispip.clang.cn/cernet.html)
中国其他ISP IP地址段：(http://ispip.clang.cn/othernet.html)

脚本

[root@localhost ~]# vim ispip.sh

```
#!/bin/bash

url="http://ispip.clang.cn/"
for  i  in  chinatelecom  unicom_cnc  cmcc  crtc  cernet  othernet
do
       wget  $url$i.html  -O  /tmp/$i.txt
       sed  -n  '/^[0-9]/  s/$/;/gp'  /tmp/$i.txt  |  sed  "1iacl  $i  {"  |  sed  '$a};'  >  /var/named/$i.txt
done
```

理解思路即可。无法实验
真正客户端的匹配方式

```
[root@ns2 named]# chmod +x ispip.sh
[root@ns2 named]# crontab -e
[root@ns2 named]# bash ispip.sh
[root@ns2 named]# ls
cernet.txt        crtc.txt  ispip.sh     named.localhost  unicom_cnc.txt
chinatelecom.txt  data      named.ca     named.loopback   vim
cmcc.txt          dynamic   named.empty  othernet.txt
[root@ns2 named]# cat /var/named/cernet.txt
[root@ns2 named]# vim /etc/named.conf
    options {

    include "/var/named/cernet.txt";
    include "/var/named/cernet.txt";
    include "/var/named/crtc.txt";
    include "/var/named/unicom_cnc.txt";
    include "/var/named/chinatelecom.txt";
    include "/var/named/cmcc.txt";
    include "/var/named/othernet.txt";

    view  "cernet"  {
       match-clients { cernet; };
       zone "study.com" IN {
    options {
            directory       "/var/named";
    };

    include "/var/named/cernet.txt";
    include "/var/named/crtc.txt";
    include "/var/named/unicom_cnc.txt";
    include "/var/named/chinatelecom.txt";
    include "/var/named/cmcc.txt";
    include "/var/named/othernet.txt";

    view  "cernet"  {
       match-clients { cernet; };
       zone "study.com" IN {
            type master;
            file "study.zheng.cernet";
        };
```

```
};

view  "crtc"  {
   match-clients { crtc; };
   zone "study.com" IN {
        type master;
        file "study.zheng.crtc";
    };
};

view  "unicom_cnc"  {
   match-clients { unicom_cnc; };
   zone "study.com" IN {
        type master;
        file "study.zheng.unicom_cnc";
    };
};


view  "chinatelecom"  {
   match-clients { chinatelecom; };
   zone "study.com" IN {
        type master;
        file "study.zheng.chinatelecom";
    };
};


view  "cmcc"  {
   match-clients { cmcc; };
   zone "study.com" IN {
        type master;
        file "study.zheng.cmcc";
    };
};


view  "othernet"  {
   match-clients { othernet; };
   zone "study.com" IN {
        type master;
        file "study.zheng.othernet";
    };
};
```

然后在指定IP就好啦　----参照构建智能DNS域名解析服务器的案例

RSYNC远程同步服务

一、RSYNC远程同步服务介绍

- 正确有效的备份方案结合Crontab计划任务、Shell脚本进行本地本地备份。
- 异地备份提高可靠性。（黑客攻击删除本地备份、磁盘故障）
- 在企业中广泛应用

1.rsync（Remote Sync,远程同步）开源异地文件备份工具

- 不同主机之间镜像同步整个目录树
- 增量备份
- 保持链接和权限功能
- 传输前压缩（带宽小、速度快）--异地备份、镜像服务

2.rsync在生产环境中的使用场景：

1.数据异地备份

异地灾备（容灾）https://www.jianshu.com/p/60d55f1bf493

一、异地容灾主要备份三种数据

1、DB数据

2、操作系统

3、日志信息

二、恢复时间不能超过30分钟

三、图中为DB的备份方式

DB总的有四份备份：

生产存储一份、移动硬盘一份、备份存储一份、灾备存储一份。

备份方式为：

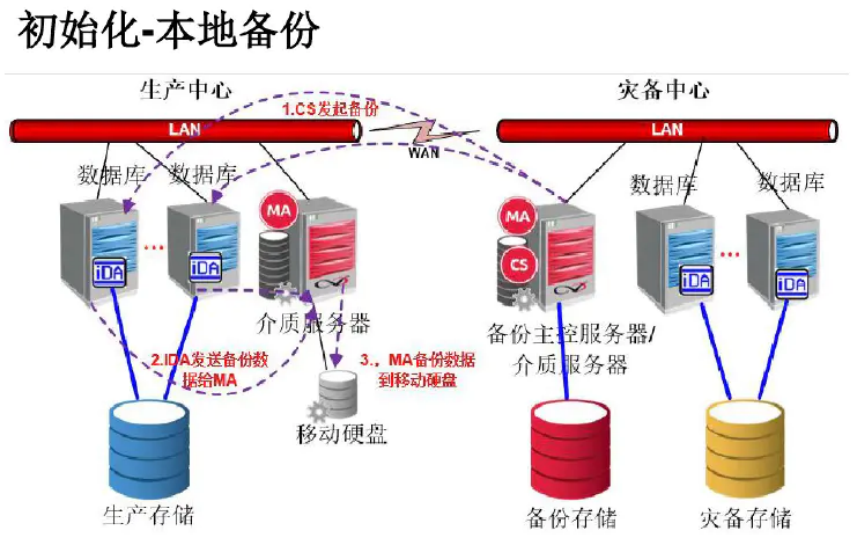
平时通过生产系统的介质服务器传输到移动硬盘，

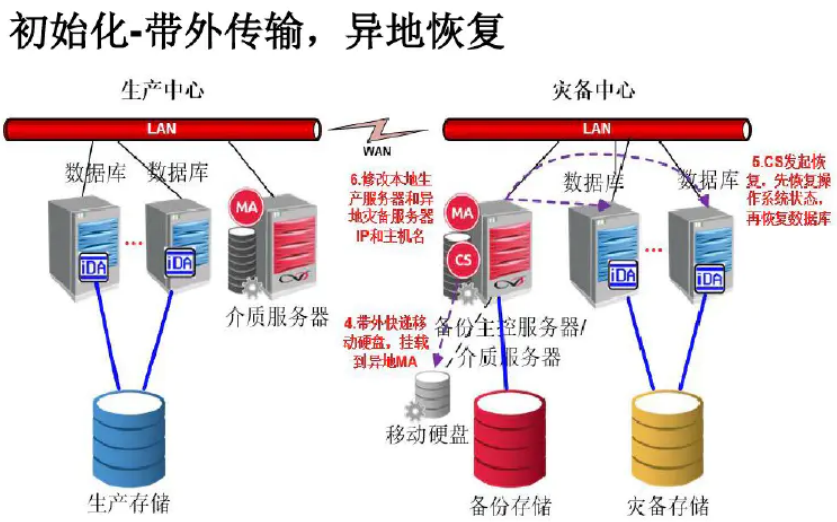
通过CS传输数据到灾备中心的介质服务器，

在通过介质服务器传输到备份存储、灾备存储。

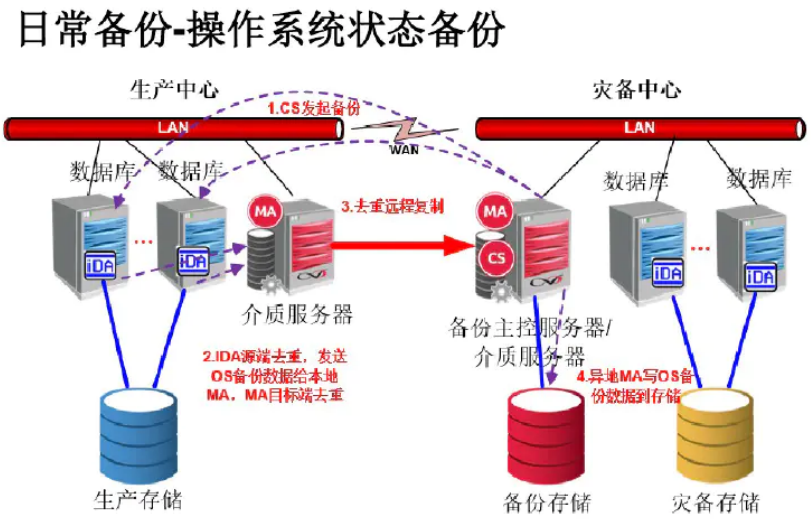
生产中心发生异常时的DB切换方式为：

将移动硬盘迅速转移挂载到灾备中心的介质服务器，然后再发起恢复

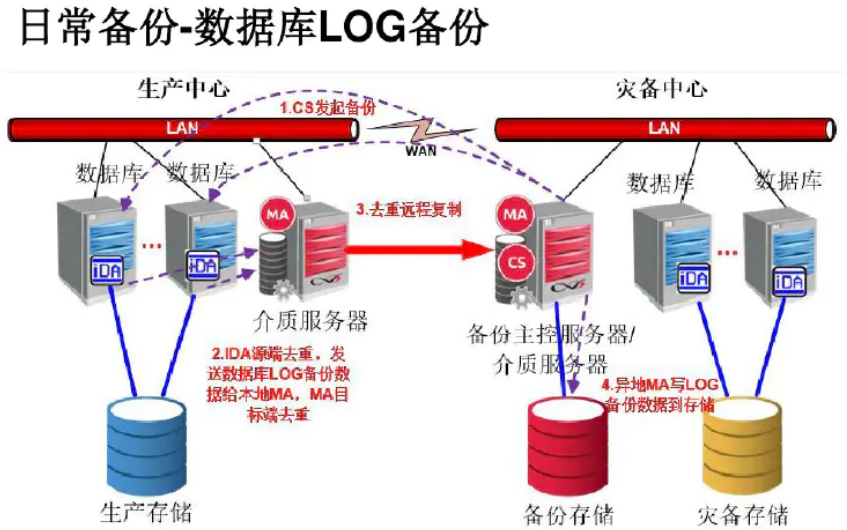




四、日常对OS进行每日备份，通过CS传输到灾备中心的介质服务器，再发送给备份存储和灾备存储，即OS的备份有三份:生产存储、备份存储、灾备存储

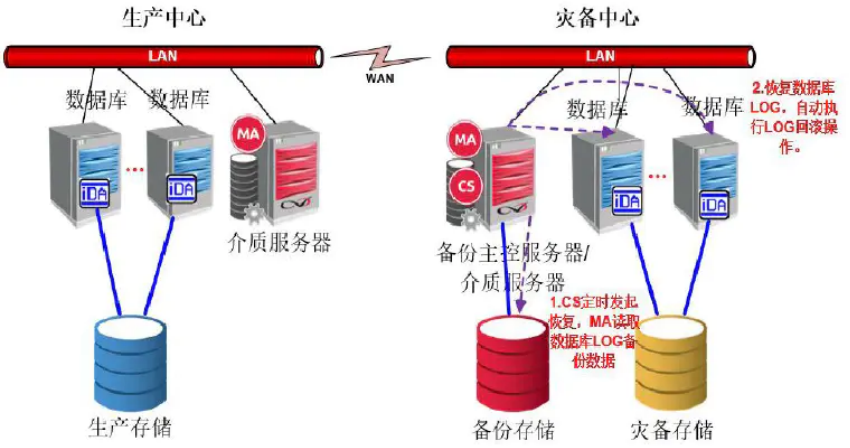


五、日志的备份和OS一样

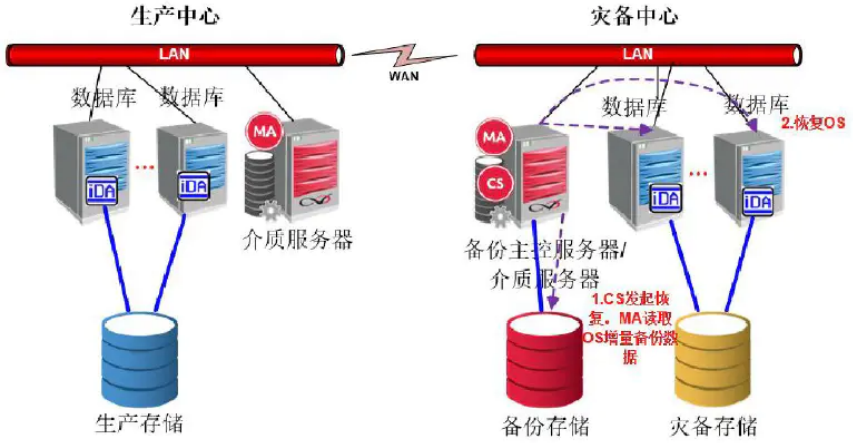


六、恢复切换步骤：日志恢复、OS恢复、修改IP和主机名、移动硬盘转移挂载

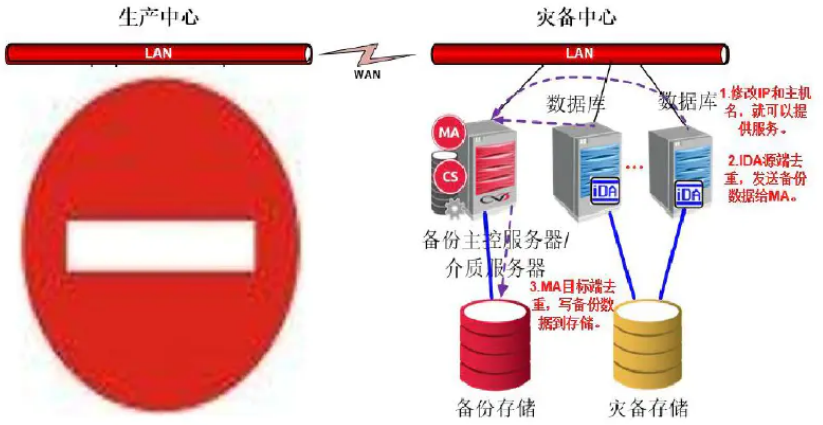
日常恢复-数据库LOG热恢复



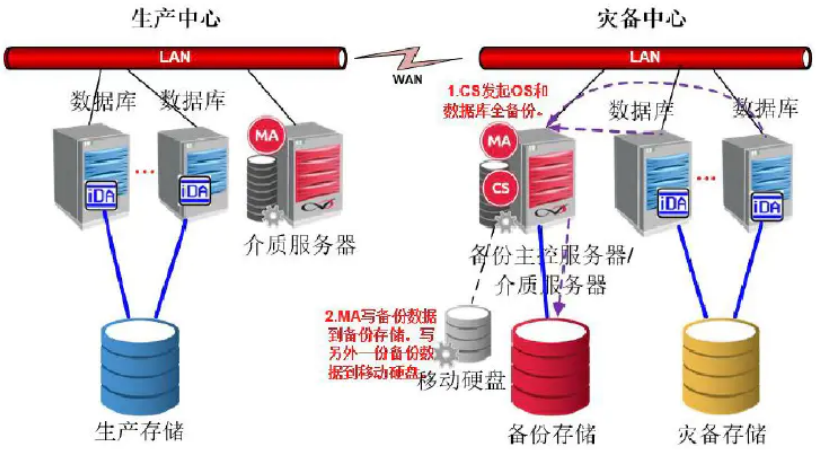
日常恢复-操作系统状态恢复



灾难切换

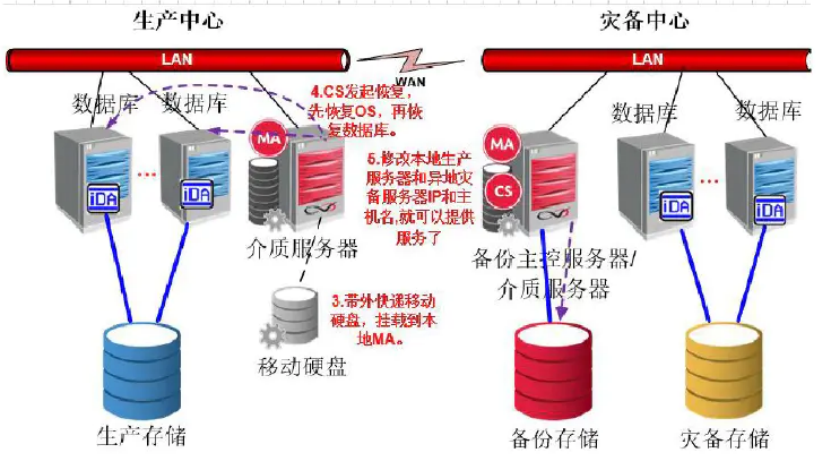


灾难回切-异地备份



七、本地恢复

灾难回切-本地恢复

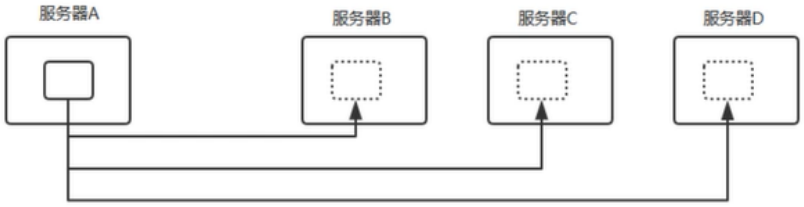


八、两地传输带宽的计算要考虑每日数据增量、每日传输量、传输可用率

灾备链路带宽估算

- 增量数据每台服务器每周大概为500M-2G，按上限每台服务器每周2G计算，6台服务器6*2=12GB，则每天增量数据是12/7=1.72 GB，10Mbps的专线链接带宽每小时可以传输的数据量最大是10Mbps*3600秒/8/1024=4.39 GB，考虑到网络丢包、重传等影响网络传输性能等因素，如果按带宽可利用率80%计算，10Mb的专线链接带宽每小时可以传输的数据量约为4.39 GB*0.8=3.51 GB。所以，传输完1.72GB的数据大约需要1.72GB/3.51GB=0.49小时。

2.WEB集群中节点数据的批量更新



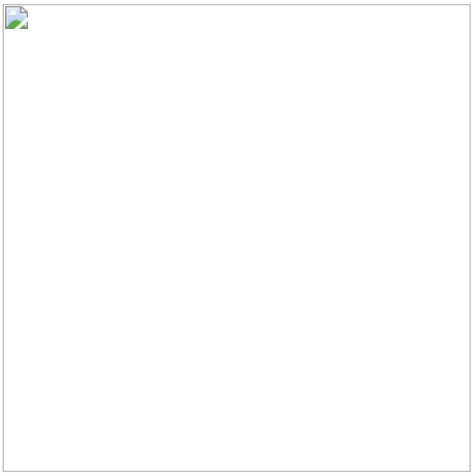
大型网站服务（淘宝）背后不止一台服务器

- 支撑用户数量多，访问量高；
- 故障导致服务中断的几率较小；
- 多台服务器的数据一样

管理员的操作：

先将第一台服务器的数据修改好，其他的通过备份工具同步（保证一模一样）就好

3.YUM服务器与互联网镜像站软件同步



centos的官方网站是最早发布软件包的，网络其他网站 服务器（阿里云，搜狐等从centOS官方网站下载），公司通过互联网服务器更新软件包，保证能够尽快安装最新版的软件包

只要涉及跨主机同步，数据同步，就要想到rsync

- 不用特意区分谁是发起端谁是同步源，谁都可以当，没有明显的C/S结构
- 权限：w、r

配置rsync同步源：

建立配置文件/etc/rsync.conf ---> 备份账号信息 ----> --deamon 运行

rsync两种源模式：

- SSH备份源
- rsync备份源

一、 配置rsync备份源 192.168.200.111

步骤：

1.建立/etc/rsync.conf 配置文件

2.为备份账户创建数据文件

3.启动rsync服务进程

1.建立/etc/rsync.conf 配置文件，以源目录/var/www/html，备份账号backuper作为示例

```
[root@master ~]# vim /etc/rsyncd.conf
uid  =  nobody                                                    //用户名
gid  =  nobody                                                    //组名
use  chroot  =  yes                                               //禁锢在源目录
address  =  192.168.200.103                                       //监听地址
port  =  873                                                      //监听端口
log  file  =  /var/log/rsyncd.log                                 //日志文件位置
pid  file  =  /var/run/rsyncd.pid                                 //存放进程ID文件位置
hosts allow = 192.168.200.0/24                                    //允许访问的客户端地址
[wwwroot]                                                         //共享模块名称
               path  =  /var/www/html                             //源目录的实际路径
               comment  =  Document  Root  os www.crushlinux.com  //描述信息
               read  only  =  yes                                 //是否为只读
               dont  compress  =  *.gz  *.bz2  *.tgz  *.zip  *.rar  *.z  //同步时不再压缩的文件类型
               auth  users  =  backuper                           //备份授权用户
               secrets  file  =  /etc/rsyncd_users.db             //存放账户信息的数据文件
```

基于安全考虑：

- rsync备份源仅允许只读 - 下行同步
- 同步采用匿名
- 将auth users 和 secrets file 两行去掉即可

2.为备份账户创建数据文件

- 冒号分割
- 密码明文存放
- 避免泄露调整权限

```
[root@master ~]# vim /etc/rsyncd_users.d
    backuper:pwd123
[root@master ~]# chmod 600 /etc/rsyncd_users.db
```

backuper应对/var/www/html 有读取权限 = other组具有读取权限即可

```
[root@master ~]# mkdir /var/www/html/ -p
[root@master ~]# ls -ld /var/www/html/
```

3.使rsync以守护进程的方式在后台运行

```
[root@master ~]# rsync --daemon
[root@master ~]# netstat -lnpt | grep rsync
    tcp        0      0 192.168.200.103:873     0.0.0.0:*               LISTEN      3728/rsync
```

4.关闭rsync服务 ----直接杀进程

```
[root@master ~]# kill $(cat /var/run/rsync.pid)
[root@master ~]# netstat -anpt | grep rsync
[root@master ~]# cat /var/run/rsyncd.pid
    3728
[root@master ~]# kill -9 3728
[root@master ~]# rsync --daemon
```

```
[root@master ~]# failed to create pid file /var/run/rsyncd.pid: File exists
[root@master ~]# rm -rf /var/run/rsyncd.pid
[root@master ~]# rsync --daemon
```

二、使用rsync备份工具　192.168.200.104

- 配好同步源服务器后，使用rsync工具执行远程同步
- 客户机（发起端）执行

本地备份“=cp”

```
[root@slave ~]# rsync /etc/fstab /opt/                          #文件
[root@slave ~]# rsync -rl /etc/fstab /boot/grub/ /opt      #目录
```

rsync命令格式及备选选项：

-a ： 归档模式，保留权限
-v ：详细(verbose)信息
-z ：压缩（compress）
-H ：保持硬链接属性
-A ： 保留ACL属性信息（setfacl 设置、getfacl查看）
--delete ：删除目标位置有但原始位置没有的文件

备份源表示：

- 用户名@主机地址：共享模块名
- rsync://用户名@主机地址/共享模块名

```
[root@slave ~]# rsync -avz backuper@192.168.200.103::wwwroot /root
    Password:
    receiving incremental file list
    ./
    1.txt

    sent 46 bytes  received 103 bytes  33.11 bytes/sec
    total size is 0  speedup is 0.00
[root@master ~]# cd /var/www/html/
[root@master html]# touch 2.txt
[root@slave ~]# rsync -avz rsync://backuper@192.168.200.103/wwwroot /root        #默认起增量功能
    Password:
    receiving incremental file list
    ./
    2.txt

    sent 46 bytes  received 127 bytes  38.44 bytes/sec
    total size is 0  speedup is 0.00
```

rsync备份操作案例

```
[root@slave ~]# mkdir /myweb
[root@slave ~]# rsync -avzH --delete backuper@192.168.200.103::wwwroot /myweb
    Password:
    receiving incremental file list
    ./
    1.txt
    2.txt
```

```
    sent 69 bytes  received 178 bytes  70.57 bytes/sec
    total size is 11  speedup is 0.04
```

编写rsync备份脚本

- 解决密码交互问题

  1.export RSYNC_PASSWORD

  取消密码交互设置

  unset RSYNC_PASSWORD

2. --password-file 指定存储密码文件

```
[root@slave ~]# vim /etc/server.pass
    pwd123
[root@slave ~]# chmod 600 /etc/server.pass
[root@slave ~]# rsync -avzH --delete --password-file=/etc/server.pass backuper@192.168.200.103::wwwroot
/mywebreceiving incremental file list

    sent 20 bytes  received 84 bytes  208.00 bytes/sec
    total size is 11  speedup is 0.11
```

- 计划任务

```
[root@slave ~]# vim /var/spool/cron/root
        30      22      *       *       *       rsync -avzH --delete --password-file=/etc/server.pass
backuper@192.168.20
[root@slave ~]# systemctl restart crond
```

二、配置rsync+inotify实时同步

rsync的问题：

- 更新不频繁
- 定时

ps： linux内核版本号

- inotify 的作用：监控文件系统的各种变化（文件存取、删除、移动、修改、属性）
- 实现触发式备份--实时同步
- 适合上行同步

1.调整inotify内核参数

监控目录、文件数量较多变化频繁时，加大数值

```
[root@slave ~]# vim /etc/sysctl.conf
[root@slave ~]# sysctl -p
    fs.inotify.max_queued_events = 16384
    fs.inotify.max_user_instances = 1024
    fs.inotify.max_user_watches = 1048576
```

2.安装inotify-tools

inotify-tools提供了inotifywait,inotifywatch工具，来监控、汇总改动情况

```
[root@slave ~]# rz

[root@slave ~]# tar -xf inotify-tools-3.14.tar.gz
[root@slave ~]# cd inotify-tools-3.14/
[root@slave inotify-tools-3.14]# ./configure && make && make install
```

以监控网站目录/var/www/html为例，执行inotifywait,命令后，在另外一个终端中改动/var/www/html目录下的内容。

```
[root@slave ~]# mkdir -p /var/www/html
[root@slave ~]# inotifywait -mrq -e modify,create,move,delete,attrib /var/www/html/
    -e：指定监控事件
    -m：持续监控
    -r：递归整个目录
    -q：简化输出信息
```

另一个终端：tty1 tty2

在/var/www/html/目录下添加文件、移动文件、跟踪屏幕输出结果

```
[root@slave ~]# inotifywait -mrq -e modify,create,move,delete,attrib /var/www/html/
    /var/www/html/ CREATE,ISDIR a
    /var/www/html/ CREATE,ISDIR b
    /var/www/html/ MOVED_FROM,ISDIR b
    /var/www/html/ MOVED_TO,ISDIR bb
    /var/www/html/ DELETE,ISDIR a
       目录                  事件                    文件
```

3.编写触发式同步脚本

环境准备：

```
    [root@master ~]# vim /etc/rsyncd.conf
    [root@master ~]# ps aux | grep rsync
        root       1740  0.0  0.0 114740   576 ?        Ss   17:10   0:00 rsync --daemon
        root       2954  0.0  0.0 112720   980 pts/0    R+   19:25   0:00 grep --color=auto rsyn
    [root@master ~]# kill -9 1740
    [root@master ~]# rm -rf /var/run/rsyncd.pid
    [root@master ~]# rsync --daemon
    [root@master ~]# netstat -lnpt | grep rsync
        tcp        0      0 192.168.200.103:873     0.0.0.0:*               LISTEN      2980/rsync
    [root@master ~]# chown nobody:nobody /var/www/html/
    [root@master ~]# ls -ld
        dr-xr-x---. 15 root root 4096 3月  15 19:24 .
    [root@master ~]# ls -ld /var/www/html/
        drwxr-xr-x 2 nobody nobody 32 3月  15 17:20 /var/www/html/
```

脚本：

```
[root@rsync-slave  ~]#  vim  /opt/inotify.sh      #注意：脚本名中不要包含rsync关键词
    #!/bin/bash
    INOTIFY_CMD="inotifywait  -mrq  -e  modify,create,move,attrib,delete  /var/www/html/"
    RSYNC_CMD="rsync  -azH  --delete  --password-
    file=/etc/server.pass  /var/www/html/  backuper@192.168.200.111::wwwroot"
    $INOTIFY_CMD  |  while  read  DIRECTORY  EVENT  FILE
    do
                    if  [  $(pgrep  rsync  |  wc  -l)  -le  0  ]
                    then
                                    $RSYNC_CMD
```

```
                fi
    done
[root@slave ~]# chmod +x /opt/inotify.sh
[root@slave ~]# echo "/bin/bash /opt/inotify.sh" >> /etc/rc.local        #每次开机自动执行脚本
```

脚本用来测试本机/var/www/html目录的变动，一旦有更新立刻触发rsync同步操作，上传至服务器192.168.200.103的/var/www/html/目录下

1.本机运行脚本

```
tty1 : [root@slave ~]# bash /opt/inotify.sh
```

2.本机执行创建、删除命令

```
tty2 : [root@slave ~]# cd /var/www/html/
    [root@slave html]# mkdir a
    [root@slave html]# mkdir b
    [root@slave html]# mv a aa
    [root@slave html]# mv b bb
    [root@slave html]# touch a.txt
```

3.查看服务器目录变化

```
[root@master ~]# ls /var/www/html/
```

三、Sersync+Rsync 实现数据文件实时同步

1.rsync+inotify-tools与rsync+sersync架构的区别

rsync+inotify-tools

- inotify只能记录变化，无法指向具体位置
- rsync同步时，遍历查找变更文件，触发全部文件同步，数据量很大时，整个目录同步非常耗时，效率很低。

rsync+sersync

- sersync记录变化的同时可以指向具体某个文件的名字
- rsync同步时，只同步增量文件目录，效率高

2.同步过程

1.在源数据服务器上开启sersync服务，sersync负责监控配置路径中的文件系统事件变化

2.调用rsync命令把更新的文件同步到目标服务器

3.需要在源数据服务器配置sersync，在同步目标服务器配置rsync server

3.操作：基于第一个实验

1.安装rsync

2.编辑/etc/rsyncd.conf配置文件

```
    [root@master ~]# vim /etc/rsyncd.conf
        uid = nobody
        gid = nobody
        use chroot = yes
        max connections = 100              #最大连接数
        timeout = 600                      #超时时间
        ignore errors                      #忽略错误
        list = false                       #不显示服务端资源列表
```

```
address = 192.168.200.103
port = 873
log file = /var/log/rsyncd.log
pid file = /var/run/rsyncd.pid
hosts allow =192.168.200.0/24
[wwwroot]
        path = /var/www/html
        comment = Document Root os www.crushlinux.com
        read only = no
        dont compress = *.gz *.bz2 *.tgz *.zip *.rar *.z
        auth users = backuper
        secrets file = /etc/rsyncd_users.db
```

3.创建用户数据文件

4.测试------ 一定要成功

```
[root@slave ~]# rsync -az --delete --password-file=/etc/server.pass /etc/hosts backuper@192.168.200.103::wwwroot
[root@master ~]# ls /var/www/html/
       aa  a.txt  bb  hosts
```

5.部署Sersync服务

下载sersync

```
[root@slave ~]# rz
[root@slave ~]# tar xf sersync2.5.4_64bit_binary_stable_final.tar.gz -C  /usr/local
[root@slave ~]# cd /usr/local
[root@slave local]# ls
bin  etc  games  GNU-Linux-x86  httpd  include  lib  lib64  libexec  sbin  share  src
[root@slave local]# mv GNU-Linux-x86 sersync          #修改文件名
[root@slave local]# cd sersync
[root@slave sersync]# ls
   confxml.xml  sersync2
[root@slave sersync]# cp confxml.xml confxml.xml.bak
[root@slave sersync]# vim confxml.xml
   24  <localpath watch="/var/www/html">
   25  <remote ip="192.168.200.103" name="wwwroot"/>
   31  <auth start="true" users="backuper" passwordfile="/etc/server.pass"/>
[root@slave sersync]# ./sersync2 -d -r -o /usr/local/sersync/confxml.xml
   option: -d      后台运行
   option: -r              递归目录
   option: -o      指定目录文件
```

6.测试

```
[root@slave sersync]# cd /var/www/html
[root@slave html]# ls
   aa  a.txt  bb  haha.txt  hehe.txt  hiehei.txt
[root@slave html]# mv haha.txt ahhahahha.txt
[root@slave html]# mv hehe.txt hehehehhehe.txt
[root@slave html]# mv hiehei.txt jajajjaja.txt

[root@master ~]# ls /var/www/html/
aa  ahhahahha.txt  a.txt  bb  hehehehhehe.txt  jajajjaja.txt
```

同步成功~

四、扩展一：配置SSH备份源

类似于rsync源的远程同步服务，源准备方式不同

1.SSH备份源

优点：远程连接安全，增强备份的保密性，容易实现。

下载：在下行同步中，备份源负责提供文档的原始位置，发起端应对文件具有读取权限

上传：在上行同步中，备份源负责提供文档的目标位置，发起端应对文件具有写入权限

SSH备份源表示方式：用户名@主机地址:目标路径

2.配置过程

a、确认备份源文件夹位置

b、准备备份操作用户

3.案例

- **192.168.200.103机器的网站目录/var/www/html作为备份源**
- **用户down做下行（下载）备份**
- **用户up做上行（上传）备份**

1.备份源192.168.200.103

```
[root@rsync-master ~]# yum -y install httpd rsync
[root@rsync-master ~]# useradd up
[root@rsync-master ~]# echo "123456" | passwd --stdin up
    更改用户 up 的密码 。
    passwd：所有的身份验证令牌已经成功更新。
[root@rsync-master ~]# useradd down
[root@rsync-master ~]# echo "123456" | passwd --stdin down
    更改用户 down 的密码 。
    passwd：所有的身份验证令牌已经成功更新。

[root@rsync-master ~]# vim /etc/ssh/sshd_config
    122 UseDNS no                                          //关闭UseDNS加速SSH登录
[root@rsync-master ~]# systemctl restart sshd
```

调整/var/www/html目录权限，使down用户有读取权限，up用户有写入权限,建议将目录的属主改为备份用户，另外需要为web服务的运行用户指定额外的权限。

```
[root@rsync-master ~]# chown -R up:up /var/www/html/
[root@rsync-master ~]# setfacl -R -m user:apache:rwx /var/www/html/
[root@rsync-master ~]# getfacl /var/www/html/
    getfacl: Removing leading '/' from absolute path names
    # file: var/www/html/
    # owner: up
    # group: up
    user::rwx
    user:apache:rwx
    group::r-x
    mask::rwx
    other::r-x
```

2.acl访问控制机制参数详解

setfacl ：设置acl权限

getfacl ：查看acl权限

-R： 递归

-m ：制定权限

-x ：个别删除

-b ：全部删除

注意：下面两行不需要执行，作为了解

```
setfacl -R -b /var/www/html                    //表示删除所有ACL属性
setfacl -R -x user:apache /var/www/html/              // 只删除某一项ACL属性
```

凡是以后在/var/www/html/upload/新建立的文档，apache都具有rwx权限

```
[root@rsync-master ~]# setfacl -m default:user:apache:rwx /var/www/html/
[root@rsync-master ~]# getfacl /var/www/html/ |grep default
getfacl: Removing leading '/' from absolute path namesdefault:user::rwx
default:user:apache:rwx
default:group::r-x
default:mask::rwx
default:other::r-x
```

3.访问SSH备份源，下载到本地/opt目录

```
[root@rsync-slave ~]# rsync -avz down@192.168.200.103:/var/www/html/ /opt/
down@192.168.200.103's password:                                    //输入down用户密码
```

4.下行同步ssh备份源

- **将服务器A的/var/www/html 文件夹与B本地/wwwroot文件夹同步**
- **（保持文件权限属性，软硬连接，ACL属性，删除/wwwroot中多余文件，传输过程进行加密）**

```
[root@rsync-slave ~]# mkdir -p /wwwroot
[root@rsync-slave ~]# rsync -avzH --delete down@192.168.200.103:/var/www/html/ /wwwroot
down@192.168.200.103's password:                                    //输入down用户密码
[root@rsync-slave ~]# ls /wwwroot/
aaa  bb
```

对于同一项远程同步任务，再次执行时，自动做增量更新，同名的文件将不再重复复制

备份源192.168.200.103

```
[root@rsync-master ~]# cd /var/www/html/
[root@rsync-master html]# for i in {1..10}; do touch $i.txt; done
```

发起源192.168.200.104：

```
[root@rsync-slave ~]# rsync -avzH --delete down@192.168.200.103:/var/www/html/ /wwwroot
down@192.168.200.103's password:                                    //输入down用户密码
[root@rsync-slave ~]# ls /wwwroot
10.txt  1.txt  2.txt  3.txt  4.txt  5.txt  6.txt  7.txt  8.txt  9.txt  aaa  bb
```

5.上行同步ssh备份源

将客户机中的anaconda-ks.cfg文件上传到备份源服务器的/var/www/html目录下，由于用户是up并非root用户，因此 -g -o 等选项无法使用.

```
[root@rsync-slave ~]# cd /root
```

```
[root@rsync-slave ~]# rsync -rlvz --delete anaconda- ks.cfg up@192.168.200.103:/var/www/html
up@192.168.200.103's password:                                    //输入up用户密码
sending incremental file list
anaconda-ks.cfg

sent 1,122 bytes  received 35 bytes  210.36 bytes/sec
total size is 1,847  speedup is 1.60

[root@rsync-master ~]# ls /var/www/html/

10.txt  1.txt  2.txt  3.txt  4.txt  5.txt  6.txt  7.txt  8.txt  9.txt  aaa  abc.txt  anaconda-
ks.cfg  bb
```

6.ssh备份源的无交互验证

- 由于脚本根据crond时间来执行，用户没办法按时根据提示输入密码
- 192.168.200.104主机上创建密钥对，将公钥文件发给192.168.200.103服务器中的备份用户，实现无交互登录

1.创建密钥对

```
[root@rsync-slave ~]# ssh-keygen -t rsa
```

2.复制密钥对

```
[root@rsync-slave ~]# ssh-copy-id up@192.168.200.103
[root@rsync-slave ~]# ssh-copy-id down@192.168.200.103
```

3.连接测试

```
[root@rsync-slave ~]# ssh down@192.168.200.103
[down@rsync-master ~]$ exit
    登出
    Connection to

    192.168.200.103 closed.
[root@rsync-slave ~]# ssh up@192.168.200.103
[up@rsync-master ~]$ exit
    登出
    Connection to
    192.168.200.103 closed.

[root@rsync-slave ~]# rsync -avzH --delete down@192.168.200.103:/var/www/html/ /wwwroot
```