Linux下Rsync+sersync实现数据实时同步

**前言：**

一、为什么要用Rsync+sersync架构?

1、sersync是基于Inotify开发的，类似于Inotify-tools的工具

2、sersync可以记录下被监听目录中发生变化的（包括增加、删除、修改）具体某一个文件或某一个目录的名字，然后使用rsync同步的时候，只同步发生变化的这个文件或者这个目录。

二、Rsync+Inotify-tools与Rsync+sersync这两种架构有什么区别？

1、Rsync+Inotify-tools

（1）：Inotify-tools只能记录下被监听的目录发生了变化（包括增加、删除、修改），并没有把具体是哪个文件或者哪个目录发生了变化记录下来；

（2）：rsync在同步的时候，并不知道具体是哪个文件或者哪个目录发生了变化，每次都是对整个目录进行同步，当数据量很大时，整个目录同步非常耗时（rsync要对整个目录遍历查找对比文件），因此，效率很低。

2、Rsync+sersync

（1）：sersync可以记录下被监听目录中发生变化的（包括增加、删除、修改）具体某一个文件或某一个目录的名字；

（2）：rsync在同步的时候，只同步发生变化的这个文件或者这个目录（每次发生变化的数据相对整个同步目录数据来说是很小的，rsync在遍历查找比对文件时，速度很快），因此，效率很高。

小结：当同步的目录数据量不大时，建议使用Rsync+Inotify-tools；当数据量很大（几百G甚至1T以上）、文件很多时，建议使用Rsync+sersync。

**说明：**

操作系统：CentOS 5.X

源服务器：192.168.21.129

目标服务器：192.168.21.127，192.168.21.128

目的：把源服务器上/home/www.osyunwei.com目录实时同步到目标服务器的/home/www.osyunwei.com下

**具体操作：**

第一部分：分别在两台目标服务器192.168.21.127，192.168.21.128上操作

一、分别在两台在目标服务器安装Rsync服务端

1、关闭SELINUX

vi /etc/selinux/config #编辑防火墙配置文件

#SELINUX=enforcing #注释掉

#SELINUXTYPE=targeted #注释掉

SELINUX=disabled #增加

:wq! #保存，退出

setenforce 0 #立即生效

2、开启防火墙tcp 873端口（Rsync默认端口）

vi /etc/sysconfig/iptables #编辑防火墙配置文件

-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 873 -j ACCEPT

:wq! #保存退出

/etc/init.d/iptables restart #最后重启防火墙使配置生效

3、安装Rsync服务端软件

yum install rsync xinetd #安装

vi /etc/xinetd.d/rsync #编辑配置文件，设置开机启动rsync

disable = no #修改为no

:wq! #保存退出

/etc/init.d/xinetd start #启动（CentOS中是以xinetd来管理Rsync服务的）

4、创建rsyncd.conf配置文件

vi /etc/rsyncd.conf #创建配置文件，添加以下代码

log file = /var/log/rsyncd.log #日志文件位置，启动rsync后自动产生这个文件，无需提前创建

pidfile = /var/run/rsyncd.pid  #pid文件的存放位置

lock file = /var/run/rsync.lock  #支持max connections参数的锁文件

secrets file = /etc/rsync.pass  #用户认证配置文件，里面保存用户名称和密码，后面会创建这个文件

motd file = /etc/rsyncd.Motd  #rsync启动时欢迎信息页面文件位置（文件内容自定义）

[home\_www.osyunwei.com] #自定义名称

path = /home/www.osyunwei.com/ #rsync服务端数据目录路径

comment = home\_www.osyunwei.com #模块名称与[home\_www.osyunwei.com]自定义名称相同

uid = root #设置rsync运行权限为root

gid = root #设置rsync运行权限为root

port=873  #默认端口

use chroot = no #默认为true，修改为no，增加对目录文件软连接的备份

read only = no  #设置rsync服务端文件为读写权限

list = no #不显示rsync服务端资源列表

max connections = 200 #最大连接数

timeout = 600  #设置超时时间

auth users = home\_www.osyunwei.com\_user #执行数据同步的用户名，可以设置多个，用英文状态下逗号隔开

hosts allow = 192.168.21.129  #允许进行数据同步的客户端IP地址，可以设置多个，用英文状态下逗号隔开

hosts deny = 192.168.21.254 #禁止数据同步的客户端IP地址，可以设置多个，用英文状态下逗号隔开

:wq!  #保存,退出

5、创建用户认证文件

vi /etc/rsync.pass #配置文件，添加以下内容

home\_www.osyunwei.com\_user:123456  #格式，用户名:密码，可以设置多个，每行一个用户名:密码

:wq!  #保存退出

6、设置文件权限

chmod 600 /etc/rsyncd.conf  #设置文件所有者读取、写入权限

chmod 600 /etc/rsync.pass  #设置文件所有者读取、写入权限

7、启动rsync

/etc/init.d/xinetd start  #启动

service xinetd stop   #停止

service xinetd restart  #重新启动

第二部分：在源服务器192.168.21.129上操作

一、安装Rsync客户端

1、关闭SELINUX

vi /etc/selinux/config  #编辑防火墙配置文件

#SELINUX=enforcing  #注释掉

#SELINUXTYPE=targeted  #注释掉

SELINUX=disabled  #增加

:wq!  #保存退出

setenforce 0   #立即生效

2、开启防火墙tcp 873端口（Rsync默认端口，做为客户端的Rsync可以不用开启873端口）

vi /etc/sysconfig/iptables  #编辑防火墙配置文件

-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 873 -j ACCEPT

:wq! #保存退出

/etc/init.d/iptables restart #最后重启防火墙使配置生效

3、安装Rsync客户端端软件

whereis rsync   #查看系统是否已安装rsync,出现下面的提示，说明已经安装

rsync: /usr/bin/rsync /usr/share/man/man1/rsync.1.gz

yum install  xinetd  #只安装xinetd即可，CentOS中是以xinetd来管理rsync服务的

yum install rsync xinetd #如果默认没有rsync，运行此命令进行安装rsync和xinetd

vi /etc/xinetd.d/rsync #编辑配置文件，设置开机启动rsync

disable = no #修改为no

/etc/init.d/xinetd start #启动（CentOS中是以xinetd来管理rsync服务的）

4、创建认证密码文件

vi /etc/passwd.txt  #编辑文件，添加以下内容

123456 #密码

:wq! #保存退出

chmod 600 /etc/passwd.txt  #设置文件权限，只设置文件所有者具有读取、写入权限即可

5、测试源服务器192.168.21.129到两台目标服务器192.168.21.127，192.168.21.128之间的数据同步

mkdir /home/www.osyunwei.com/ceshi  #在源服务器上创建测试文件夹，然后在源服务器运行下面2行命令

rsync -avH --port=873 --progress --delete  /home/www.osyunwei.com/  home\_www.osyunwei.com\_user@192.168.21.127::home\_www.osyunwei.com --password-file=/etc/passwd.txt

rsync -avH --port=873 --progress --delete  /home/www.osyunwei.com/  home\_www.osyunwei.com\_user@192.168.21.128::home\_www.osyunwei.com --password-file=/etc/passwd.txt

运行完成后，分别在两台目标服务器192.168.21.127，192.168.21.128上查看，在/home/www.osyunwei.com目录下有ceshi文件夹，说明数据同步成功。

系统运维  [www.osyunwei.com](http://www.osyunwei.com/)  温馨提醒：qihang01原创内容 版权所有,转载请注明出处及原文链接

二、安装sersync工具，实时触发rsync进行同步

1、查看服务器内核是否支持inotify

ll /proc/sys/fs/inotify   #列出文件目录，出现下面的内容，说明服务器内核支持inotify

-rw-r--r-- 1 root root 0 Mar  7 02:17 max\_queued\_events

-rw-r--r-- 1 root root 0 Mar  7 02:17 max\_user\_instances

-rw-r--r-- 1 root root 0 Mar  7 02:17 max\_user\_watches

备注：Linux下支持inotify的内核最小为2.6.13，可以输入命令：uname -a查看内核

CentOS 5.X 内核为2.6.18，默认已经支持inotify

2、修改inotify默认参数（inotify默认内核参数值太小）

查看系统默认参数值：

sysctl -a | grep max\_queued\_events

结果是：fs.inotify.max\_queued\_events = 16384

sysctl -a | grep max\_user\_watches

结果是：fs.inotify.max\_user\_watches = 8192

sysctl -a | grep max\_user\_instances

结果是：fs.inotify.max\_user\_instances = 128

修改参数：

sysctl -w fs.inotify.max\_queued\_events="99999999"

sysctl -w fs.inotify.max\_user\_watches="99999999"

sysctl -w fs.inotify.max\_user\_instances="65535"

vi /etc/sysctl.conf #添加以下代码

fs.inotify.max\_queued\_events=99999999

fs.inotify.max\_user\_watches=99999999

fs.inotify.max\_user\_instances=65535

:wq! #保存退出

参数说明：

max\_queued\_events：

inotify队列最大长度，如果值太小，会出现"\*\* Event Queue Overflow \*\*"错误，导致监控文件不准确

max\_user\_watches：

要同步的文件包含多少目录，可以用：find /home/www.osyunwei.com -type d | wc -l 统计，必须保证max\_user\_watches值大于统计结果（这里/home/www.osyunwei.com为同步文件目录）

max\_user\_instances：

每个用户创建inotify实例最大值

3、安装sersync

sersync下载地址：https://sersync.googlecode.com/files/sersync2.5.4\_64bit\_binary\_stable\_final.tar.gz

上传sersync2.5.4\_64bit\_binary\_stable\_final.tar.gz到/usr/local/src目录下

cd /usr/local/src

tar zxvf sersync2.5.4\_64bit\_binary\_stable\_final.tar.gz  #解压

mv GNU-Linux-x86  /usr/local/sersync  #移动目录到/usr/local/sersync

4、配置sersync

cd  /usr/local/sersync #进入sersync安装目录

cp confxml.xml confxml.xml-bak  #备份原文件

vi confxml.xml  #编辑，修改下面的代码

<?xml version="1.0" encoding="ISO-8859-1"?>

<head version="2.5">

<host hostip="localhost" port="8008"></host>

<debug start="false"/>

<fileSystem xfs="false"/>

<filter start="false">

<exclude expression="(.\*)\.svn"></exclude>

<exclude expression="(.\*)\.gz"></exclude>

<exclude expression="^info/\*"></exclude>

<exclude expression="^static/\*"></exclude>

</filter>

<inotify>

<delete start="true"/>

<createFolder start="true"/>

<createFile start="false"/>

<closeWrite start="true"/>

<moveFrom start="true"/>

<moveTo start="true"/>

<attrib start="false"/>

<modify start="false"/>

</inotify>

<sersync>

<localpath watch="/home/www.osyunwei.com">

<remote ip="192.168.21.127" name="home\_www.osyunwei.com"/>

<remote ip="192.168.21.128" name="home\_www.osyunwei.com"/>

<!--<remote ip="192.168.8.40" name="tongbu"/>-->

</localpath>

<rsync>

<commonParams params="-artuz"/>

<auth start="true" users="home\_www.osyunwei.com\_user" passwordfile="/etc/passwd.txt"/>

<userDefinedPort start="false" port="874"/><!-- port=874 -->

<timeout start="false" time="100"/><!-- timeout=100 -->

<ssh start="false"/>

</rsync>

<failLog path="/tmp/rsync\_fail\_log.sh" timeToExecute="60"/><!--default every 60mins execute once-->

<crontab start="true" schedule="600"><!--600mins-->

<crontabfilter start="false">

<exclude expression="\*.php"></exclude>

<exclude expression="info/\*"></exclude>

</crontabfilter>

</crontab>

<plugin start="false" name="command"/>

</sersync>

<plugin name="command">

<param prefix="/bin/sh" suffix="" ignoreError="true"/>  <!--prefix /opt/tongbu/mmm.sh suffix-->

<filter start="false">

<include expression="(.\*)\.php"/>

<include expression="(.\*)\.sh"/>

</filter>

</plugin>

<plugin name="socket">

<localpath watch="/opt/tongbu">

<deshost ip="192.168.138.20" port="8009"/>

</localpath>

</plugin>

<plugin name="refreshCDN">

<localpath watch="/data0/htdocs/cms.xoyo.com/site/">

<cdninfo domainname="ccms.chinacache.com" port="80" username="xxxx" passwd="xxxx"/>

<sendurl base="http://pic.xoyo.com/cms"/>

<regexurl regex="false" match="cms.xoyo.com/site([/a-zA-Z0-9]\*).xoyo.com/images"/>

</localpath>

</plugin>

</head>

:wq!  #保存退出

参数说明：

localpath watch="/home/www.osyunwei.com"：#源服务器同步目录

192.168.21.127，192.168.21.128：#目标服务器IP地址

name="home\_www.osyunwei.com"： #目标服务器rsync同步目录模块名称

users="home\_www.osyunwei.com\_user"： #目标服务器rsync同步用户名

passwordfile="/etc/passwd.txt"： #目标服务器rsync同步用户的密码在源服务器的存放路径

remote ip="192.168.21.127":  #目标服务器ip，每行一个

remote ip="192.168.21.128":  #目标服务器ip，每行一个

failLog path="/tmp/rsync\_fail\_log.sh"  #脚本运行失败日志记录

start="true"  #设置为true，每隔600分钟执行一次全盘同步

5、设置sersync监控开机自动执行

vi /etc/rc.d/rc.local  #编辑，在最后添加一行

/usr/local/sersync/sersync2 -d -r -o  /usr/local/sersync/confxml.xml  ＃设置开机自动运行脚本

:wq!  #保存退出

6、添加脚本监控sersync是否正常运行

vi  /home/crontab/check\_sersync.sh  #编辑，添加以下代码

#!/bin/sh

sersync="/usr/local/sersync/sersync2"

confxml="/usr/local/sersync/confxml.xml"

status=$(ps aux |grep 'sersync2'|grep -v 'grep'|wc -l)

if [ $status -eq 0 ];

then

$sersync -d -r -o $confxml &

else

exit 0;

fi

:wq!  #保存退出

chmod +x /usr/local/sersync/check\_sersync.sh #添加脚本执行权限

vi /etc/crontab #编辑，在最后添加下面一行

\*/5 \* \* \* \* root /home/crontab/check\_sersync.sh > /dev/null 2>&1  #每隔5分钟执行一次脚本

service crond reload  #重新加载服务

6、测试sersync实时触发rsync同步脚本是否正常运行

在源服务器192.168.21.129上创建文件inotify\_rsync\_ceshi

mkdir /home/www.osyunwei.com/inotify\_rsync\_ceshi

重新启动源服务器：192.168.21.129

等系统启动之后，查看两台目标服务器192.168.21.127，192.168.21.128的/home/www.osyunwei.com下是否有inotify\_rsync\_ceshi文件夹

然后再在源服务器192.168.21.129创建文件夹inotify\_rsync\_ceshi\_new

mkdir /home/www.osyunwei.com/inotify\_rsync\_ceshi\_new

继续查看两台目标服务器192.168.21.127，192.168.21.128的/home/www.osyunwei.com下是否有inotify\_rsync\_ceshi\_new文件夹

如果以上测试都通过，说明inotify实时触发rsync同步脚本运行正常。

**至此，Linux下Rsync+sersync实现数据实时同步完成。**

**扩展阅读：**

sersync开发者网站：http://blog.johntechinfo.com/sersyncguild

rsync参数

-v, --verbose 详细模式输出

-q, --quiet 精简输出模式

-c, --checksum 打开校验开关，强制对文件传输进行校验

-a, --archive 归档模式，表示以递归方式传输文件，并保持所有文件属性，等于-rlptgoD

-r, --recursive 对子目录以递归模式处理

-R, --relative 使用相对路径信息

-b, --backup 创建备份，也就是对于目的已经存在有同样的文件名时，将老的文件重新命名为~filename。可以使用--suffix选项来指定不同的备份文件前缀。

--backup-dir 将备份文件(如~filename)存放在在目录下。

-suffix=SUFFIX 定义备份文件前缀

-u, --update 仅仅进行更新，也就是跳过所有已经存在于DST，并且文件时间晚于要备份的文件。(不覆盖更新的文件)

-l, --links 保留软链结

-L, --copy-links 想对待常规文件一样处理软链结

--copy-unsafe-links 仅仅拷贝指向SRC路径目录树以外的链结

--safe-links 忽略指向SRC路径目录树以外的链结

-H, --hard-links 保留硬链结

-p, --perms 保持文件权限

-o, --owner 保持文件属主信息

-g, --group 保持文件属组信息

-D, --devices 保持设备文件信息

-t, --times 保持文件时间信息

-S, --sparse 对稀疏文件进行特殊处理以节省DST的空间

-n, --dry-run现实哪些文件将被传输

-W, --whole-file 拷贝文件，不进行增量检测

-x, --one-file-system 不要跨越文件系统边界

-B, --block-size=SIZE 检验算法使用的块尺寸，默认是700字节

-e, --rsh=COMMAND 指定使用rsh、ssh方式进行数据同步

--rsync-path=PATH 指定远程服务器上的rsync命令所在路径信息

-C, --cvs-exclude 使用和CVS一样的方法自动忽略文件，用来排除那些不希望传输的文件

--existing 仅仅更新那些已经存在于DST的文件，而不备份那些新创建的文件

--delete 删除那些DST中SRC没有的文件

--delete-excluded 同样删除接收端那些被该选项指定排除的文件

--delete-after 传输结束以后再删除

--ignore-errors 及时出现IO错误也进行删除

--max-delete=NUM 最多删除NUM个文件

--partial 保留那些因故没有完全传输的文件，以是加快随后的再次传输

--force 强制删除目录，即使不为空

--numeric-ids 不将数字的用户和组ID匹配为用户名和组名

--timeout=TIME IP超时时间，单位为秒

-I, --ignore-times 不跳过那些有同样的时间和长度的文件

--size-only 当决定是否要备份文件时，仅仅察看文件大小而不考虑文件时间

--modify-window=NUM 决定文件是否时间相同时使用的时间戳窗口，默认为0

-T --temp-dir=DIR 在DIR中创建临时文件

--compare-dest=DIR 同样比较DIR中的文件来决定是否需要备份

-P 等同于 --partial

--progress 显示备份过程

-z, --compress 对备份的文件在传输时进行压缩处理

--exclude=PATTERN 指定排除不需要传输的文件模式

--include=PATTERN 指定不排除而需要传输的文件模式

--exclude-from=FILE 排除FILE中指定模式的文件

--include-from=FILE 不排除FILE指定模式匹配的文件

--version 打印版本信息

--address 绑定到特定的地址

--config=FILE 指定其他的配置文件，不使用默认的rsyncd.conf文件

--port=PORT 指定其他的rsync服务端口

--blocking-io 对远程shell使用阻塞IO

-stats 给出某些文件的传输状态

--progress 在传输时现实传输过程

--log-format=formAT 指定日志文件格式

--password-file=FILE 从FILE中得到密码

--bwlimit=KBPS 限制I/O带宽，KBytes per second

-h, --help 显示帮助信息