**老男孩linux运维实战培训**

**老男孩教育教学核心思想6重：重目标、重思路、重方法、重实践、重习惯、重总结**

**学无止境，老男孩教育成就你人生的起点！**

**版权声明：**

本文作者为《老男孩linux运维实战培训》学生—**你的名字**

本文的所有内容均来自老男孩培训**命令总结**，未经本人及老男孩培训许可，禁止私自转发及使用。

QQ: **你的**XXXX

E-mail: **你的**XXXX

**联系方式:**

|  |  |  |  |
| --- | --- | --- | --- |
| 网站运维交流群： | | | |
| **Linux QQ交流群 385168604** | | **架构师 QQ交流群 390642196** | |
| **Python QQ交流群 29215534** | | **大数据 QQ交流群 421358633** | |
| =================================================================================== | | | |
| **老男孩linux实训联系方式** |  | |  |
| **咨询QQ** | **41117397（丹丹）** | | **70271111（歪歪）** |
|  | **80042789（飞雪）** | | **390320151（小雨）** |
|  | **41117483（冰冰）** | |  |
| **电话：** | **158-1059-0206（丹丹）** | | **189-1171-8229（歪歪）** |
|  | **135-5261-2571（飞雪）** | | **186-0046-2391（小雨）** |
| **网站:** | <http://www.etiantian.org> | | <http://www.oldboyedu.com> |
| **博客:** | http://oldboy.blog.51cto.com | | <http://blog.oldboyedu.com> |

**快捷方式说明:**

**ctrl + 1 一级标题**

**ctrl + 2 二级标题**

**ctrl + 3 三级标题**

**ctrl + 5 程序代码**

**ctrl + 6 正文**

**格式约定：**

蓝色字体：内容注释

目 录

[Inotify+rsync课程总结 1](#_Toc510378042)

[第1章 Inotify+rsync 实时同步实战 1](#_Toc510378043)

[1.1.1 检查系统是否支持，具备以下条件即可 2](#_Toc510378044)

[1.1.2 安装inotify-tools 2](#_Toc510378045)

[1.1.3 配置特定yum源 2](#_Toc510378046)

[1.1.4 Inotifywait命令参数 3](#_Toc510378047)

[1.1.5 演示： 3](#_Toc510378048)

[1.1.6 设置定时任务设置实时同步 4](#_Toc510378049)

[1.1.7 工具实时复制企业级调优 6](#_Toc510378050)

[1.1.8 inotify 优化（/proc/sys/fs/inotify/） 6](#_Toc510378051)

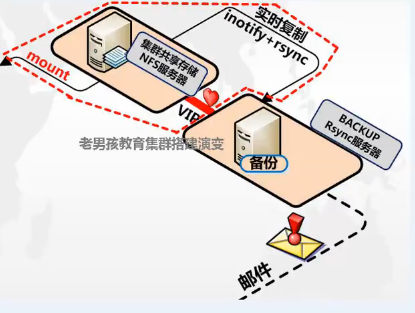
[1.1.9 Sersync功能： 7](#_Toc510378052)

[1.1.10 Sersync实时复制实践 7](#_Toc510378053)

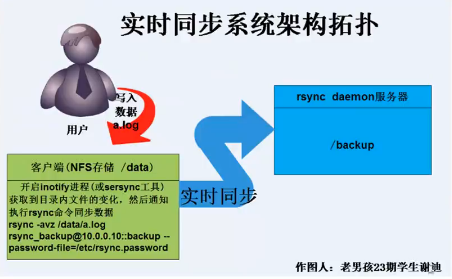
[1.1.11 启动sersync 8](#_Toc510378054)

Inotify+rsync课程总结

# Inotify+rsync 实时同步实战



是一种强大的、细粒度的、异步文件系统时间监控机制，linux内核从2.6.13起，加入了Inotify支持，通过Inotif可以监控文件系统中添加、删除、修改、移动各种事件，利用这个内核接口，第三方软件就可以监控文件系统下文件的各种变化情况，而inotify-tools正是实施这样监控的软件。



### 检查系统是否支持，具备以下条件即可

[root@nfs01 /]# uname -r

2.6.32-573.el6.x86\_64

[root@nfs01 /]# ls -l /proc/sys/fs/inotify/

总用量 0

-rw-r--r-- 1 root root 0 4月 1 10:26 max\_queued\_events

-rw-r--r-- 1 root root 0 4月 1 10:26 max\_user\_instances

-rw-r--r-- 1 root root 0 4月 1 10:26 max\_user\_watche

### 安装inotify-tools

[root@nfs01 /]# rpm -qa inotify-tools

[root@nfs01 /]#

### 配置特定yum源

wget -O /etc/yum.repos.d/epel.repo <http://mirrors.aliyun.com/repo/epel-6.repo>

[root@nfs01 /]# yum install -y inotify-tools

[root@nfs01 /]# rpm -qa inotify-tools

inotify-tools-3.14-1.el6.x86\_64

[root@nfs01 /]# which inotifywait

/usr/bin/inotifywait

### Inotifywait命令参数

|  |  |  |
| --- | --- | --- |
| 参数 | 说明 | 含义 |
| -r | --recursive | 递归查询目录 |
| -q | --quiet | 打印少量信息，仅仅打印监控事件的信息 |
| -m | --monitor | 始终保持事假监听状态 |
| --excludei |  | 排除文件或目录时，不区分大小写 |
| --timefmt |  | 指定事件输出格式 |
| --format |  | 打印使用指定的输出类似格式字符 |
| -e | --event<event1>  Access  Modify  attrib  close  open  move\_to  move  create  delete  umount | 通过此参数可以指定监控的事件  文件或目录被读取  文件或目录的内容被修改  文件或目录属性被改变  文件或目录封闭，无论读/写模式  文件或目录被打开  文件或目录被移动至另外一个目录  文件或目录被移动另一个目录或从另一个目录移动至当前目录  文件或目录被创建在当前目录  文件或目录被删除  文件系统被卸载 |
|  |  |  |

### 演示：

创建、删除

[root@nfs01 /]# inotifywait -mrq --timefmt '%d/%m/%y %H:%M' --format '%T %w%f' -e create,delete /backup/

关闭写 close\_write

[root@nfs01 /]# inotifywait -mrq --timefmt '%d/%m/%y %H:%M' --format '%T %w%f' -e close\_write,create,delete /backup/ >>/inotify.log

编写脚本：

[root@nfs01 scripts]# vim inotify\_rsync.sh

#!/bin/bash

#The is inotify and rsync dir by shell

#zuozhe cj

#time 20180401

IP=172.16.1.41

#Path=/nfsbackup/

Path=/server/scripts/

/usr/bin/inotifywait -mrq --format '%w%f' -e close\_write,delete $Path \

|while read file

do

cd $Path &&\

rsync -az ./ --delete rsync\_backup@$IP::nfsbackup/ --password-file=/etc/rsync.password

done

优化版本：

#!/bin/bash

#The is inotify and rsync dir by shell

#zuozhe cj

#time 20180401

IP=172.16.1.41

#Path=/nfsbackup/

Path=/server/scripts/

/usr/bin/inotifywait -mrq --format '%w%f' -e close\_write,delete $Path \

|while read file

do

cd $Path

if [ -f $file ];then

rsync -az $file --delete rsync\_backup@$IP::nfsbackup/ --password-file=/etc/rsync.password

else

cd $Path &&\

rsync -az ./ --delete rsync\_backup@$IP::nfsbackup/ --password-file=/etc/rsync.password

fi

done

### 设置定时任务设置实时同步

[root@nfs01 scripts]# crontab -e

#The is inotify and rsync dir by shell 🡨每分钟进行同步

#\* \* \* \* \* /bin/sh /server/scripts/inotify\_rsync.sh

开机启动脚本实现stop 和start

[root@nfs01 scripts]# vim syncd

#!/bin/bash

#chkconfig: 2345 55 66

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

# this scripts is created by cj

#cj QQ:278028843

#site :

#time 20180401

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

. /etc/init.d/functions

if [ $# -ne 1 ];then

usage: $0 {start|stop}

exit 1

fi

case "$1" in

start)

/bin/bash /server/scripts/inotify\_rsync.sh &

echo $$ >/var/run/inotify.pid

if [ `ps -ef|grep inotify|wc -l` -gt 2 ];then

action "inotify server is started" /bin/true

else

action "inotify yserver is started" /bin/false

fi

;;

stop)

kill -9 `cat /var/run/inotify.pid` >/dev/null 2>&1

pkill inotifywait

sleep 2

if [ `ps -ef|grep inotify|grep -v grep|wc -l` -eq 0 ];then

action "inotify server is stopped" /bin/true

else

action "inotify server is stopped" /bin/false

fi

;;

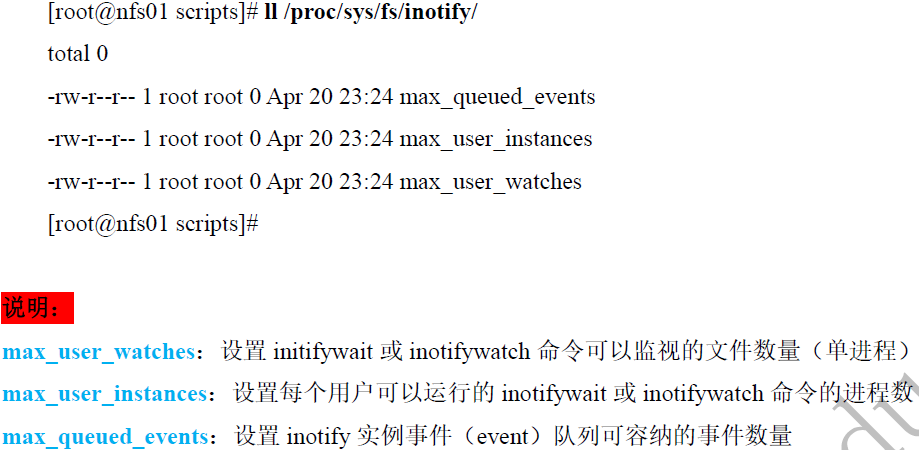
\*)

usage: $0 {start|stop}

exit 1

esac

### 工具实时复制企业级调优



### inotify 优化（/proc/sys/fs/inotify/）

1、max\_user\_watches 优化  
echo “50000000” >/proc/sys/fs/inotify/max\_user\_watches

2、max\_queued\_events优化  
echo “50000000” >/proc/sys/fs/inotify/max\_queued\_events   
注意：以上配置重启可能会失效，需要放置在rc.local文件中

#### Inotify优缺点

优点：

1. 监控文件系统事件变化，通过同步工具实现实时数据同步

缺点：

1. 并发如果大于200个文件（10-100k），同步时就会有延迟

2. 我们前面写的脚本，每次都是全部推送一次，但确实是增量的。也可以只是同步变化，不变化的不理

3. 监控事件后，条用rsync同步是单进程（加&并发），sersync多进程同步

### Sersync功能：

1. 支持配置文件管理
2. 真正的守护进程
3. 可以对失败文件定时重传
4. 第三方HTTP接口（例如更新cdn缓存）
5. 默认多线程rsync同步

***高并发数据实时同步方案小结：***

1、inotify+sersync/rsync 【文件级别】

2、drbd文件系统级别，基于block 【缺点：备节点数据不可用】

3、第三方软件同步：mysql同步，oracle，mongodb

4、程序双写，直接写两台服务器

5、产品业务逻辑解决（读写分离，备读不到，读 主）

6、NFS集群（1、4、5方案整合）（双写主存储，备存储用inotify（sersync）+rsync）,备没有找主解决延迟问题

### Inotify+Sersync实时复制实践



http://edu.51cto.com/course/5907.html?source=so

[root@nfs01 ~]# mkdir -p /application

[root@nfs01 ~]# cd /application/

[root@nfs01 application]# ll

总用量 3364

-rw-r--r-- 1 root root 3444563 4月 1 16:39 28-sersync.tar.gz

[root@nfs01 application]# tar -zxvf 28-sersync.tar.gz

[root@nfs01 application]# tree sersync/

sersync/

├── bin

│   └── sersync

├── conf

│   ├── confxml.xml

│   └── confxml.xml.ori

├── GNU-Linux-x86

├── logs

└── sersync2.5.4\_64bit\_binary\_stable\_final.tar.gz

4 directories, 4 files

【主配置文件已经配置好】

[root@nfs01 application]# chmod +x sersync/bin/sersync

[root@nfs01 application]# /application/sersync/bin/sersync -h

set the system param

execute：echo 50000000 > /proc/sys/fs/inotify/max\_user\_watches

execute：echo 327679 > /proc/sys/fs/inotify/max\_queued\_events

parse the command param

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

参数-d:启用守护进程模式

参数-r:在监控前，将监控目录与远程主机用rsync命令推送一遍

c参数-n: 指定开启守护线程的数量，默认为10个

参数-o:指定配置文件，默认使用confxml.xml文件

参数-m:单独启用其他模块，使用 -m refreshCDN 开启刷新CDN模块

参数-m:单独启用其他模块，使用 -m socket 开启socket模块

参数-m:单独启用其他模块，使用 -m http 开启http模块

不加-m参数，则默认执行同步程序

### 启动sersync

[root@nfs01 nfsbackup]# /application/sersync/bin/sersync -d -r -n 8 -o /application/sersync/conf/confxml.xml 🡨启动sersync

set the system param

execute：echo 50000000 > /proc/sys/fs/inotify/max\_user\_watches

execute：echo 327679 > /proc/sys/fs/inotify/max\_queued\_events

parse the command param

option: -d run as a daemon

option: -r rsync all the local files to the remote servers before the sersync work

option: -n thread num is： 8

option: -o config xml name： /application/sersync/conf/confxml.xml

parse xml config file

host ip : localhost host port: 8008

daemon start，sersync run behind the console

use rsync password-file :

user is rsync\_backup

passwordfile is /etc/rsync.password

config xml parse success

please set /etc/rsyncd.conf max connections=0 Manually

sersync working thread 10 = 1(primary thread) + 1(fail retry thread) + 8(daemon sub threads)

Max threads numbers is: 18 = 10(Thread pool nums) + 8(Sub threads)

please according your cpu ，use -n param to adjust the cpu rate

------------------------------------------

rsync the directory recursivly to the remote servers once

working please wait...

execute command: cd /nfsbackup && rsync -avz -R --delete ./ --timeout=100 rsync\_backup@172.16.1.41::nfsbackup --password-file=/etc/rsync.password >/dev/null 2>&1

run the sersync:

watch path is: /nfsbackup

配置开机自启动：

[root@nfs01 /]# tail -2 /etc/rc.local

/application/sersync/bin/sersync -d -r -n 8 -o /application/sersync/conf/confxml.xml

[root@nfs01 /]#