|  |
| --- |
| 文章标题  inotify+rsync  sersync |
| **作者：**  **归档：学习笔记** |
| **快捷键：**  Ctrl + 1 标题1  Ctrl + 2 标题2  Ctrl + 3 标题3  Ctrl + 4 实例  Ctrl + 5 程序代码  Ctrl + 6 正文 |
| **格式说明：**  蓝色字体：注释  黄色背景：重要  绿色背景：注意 |

****

**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！**

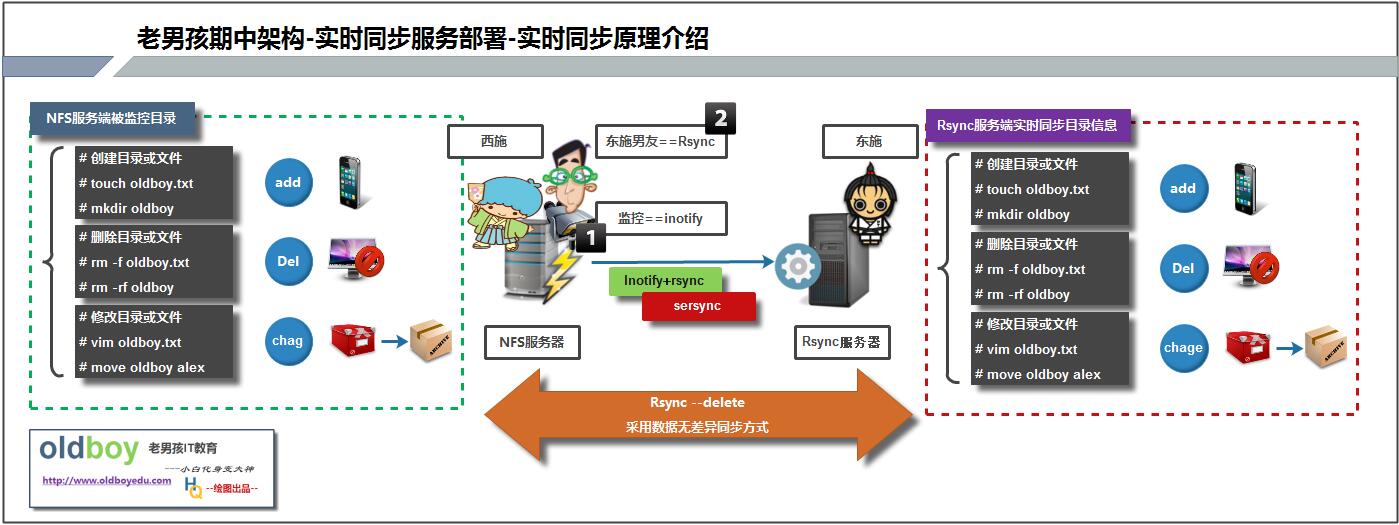
## 数据实时同步

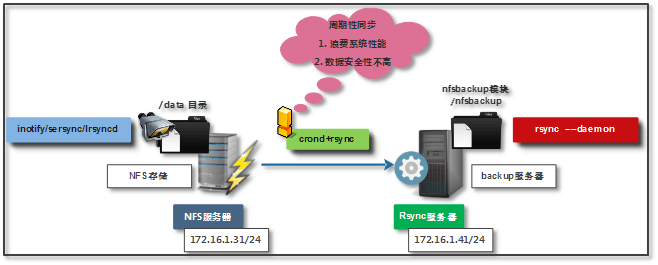
### 数据实时同步方法

什么是实时同步方法

01. 要利用监控服务，监控同步数据服务目录中信息的变化

02. 发现目录中数据产生变化，就利用rsync服务推送数据到备份服务器上





### 实现实时同步方法：

#### inotify+rsync方式实现实时数据同步：

## inotify：对同步数据目录信息的监控

## rsync：完成对数据信息的实时同步

## inotify+rsync：利用脚本进行结合

#### sersync方式实现实时数据同步

## inotify+rsync方式实现实时数据同步

### 第一个里程碑，部署rsync服务

服务端部署过程：

客户端部署过程：

### 第二个里程碑，部署inotify软件

yum install -y inotify-tools

说明：inotify-tools软件需要依赖epel源

inotify软件的参考资料链接：

https://github.com/rvoicilas/inotify-tools/wiki

常用yum仓库

01. http://mirrors.163.com/

02. http://mirrors.aliyun.com/

03. http://mirrors.sohu.com/

wget -O /etc/yum.repos.d/epel.repo http://mirrors.aliyun.com/repo/epel-6.repo ---下载centos6版本epel源文件

说明信息：

yum仓库搭建参考链接：<http://blog.oldboyedu.com/autodeploy-yum/>

#### inotify软件详解

[root@nfs01 ~]# rpm -ql inotify-tools

/usr/bin/inotifywait \*\*\*\*\*

在被监控的文件或目录上等待特定文件系统事件（open close delete等）发生，执行后处于阻塞状态，适合在shell脚本中使用

通过此命令，实现对目录或文件的监控

/usr/bin/inotifywatch

收集被监控的文件系统使用的统计数据，指文件系统事件发生的次数统计。

统计文件数据信息变化的数量

命令man手册说明

# man inotifywait

inotifywait - wait for changes to files using inotify

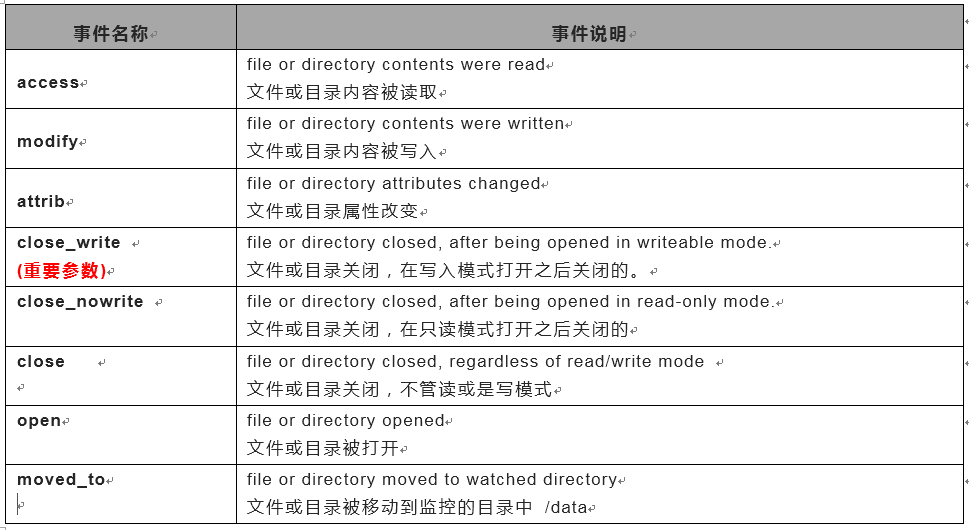
使用inotify进行监控，等待产生变化的文件信息

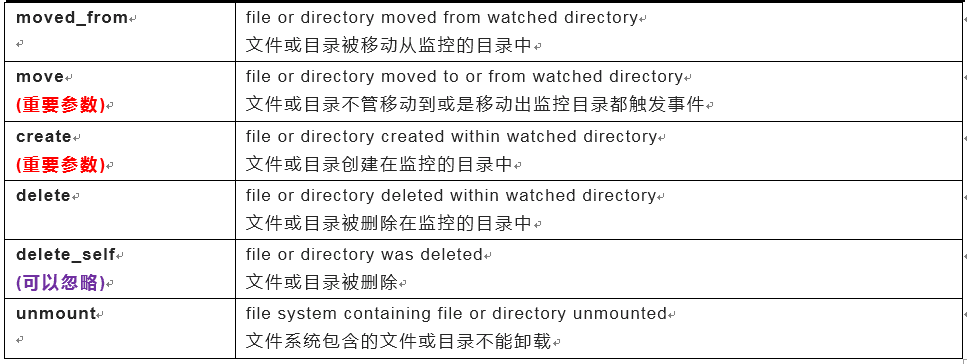
# man inotifywatch

inotifywatch - gather filesystem access statistics using inotify

使用inotify进行监控，收集文件系统访问统计信息

#### 事件说明





利用inotifywait命令监控指定事件，如何执行事件信息（-e）

close\_write 文件或目录关闭，在写入模式打开之后关闭的。

moved\_to 文件或目录被移动到监控的目录中

create 文件或目录创建在监控的目录中

delete 文件或目录被删除在监控的目录中

创建、修改、删除事件说明

创建一个文件逻辑：

inotifywait -m /data

Setting up watches.

Watches established.

/data/ CREATE oldboy.txt

/data/ OPEN oldboy.txt

/data/ ATTRIB oldboy.txt

/data/ CLOSE\_WRITE,CLOSE oldboy.txt

删除一个文件逻辑：

/data/ DELETE oldboy.txt

修改一个文件逻辑：

/data/ OPEN oldboy.txt

/data/ MODIFY oldboy.txt

/data/ CLOSE\_WRITE,CLOSE oldboy.txt

sed命令修改文件逻辑过程：

/data/ OPEN oldboy.txt

/data/ CREATE sedaRSL6J

/data/ OPEN sedaRSL6J

/data/ ACCESS oldboy.txt

/data/ MODIFY sedaRSL6J

/data/ ATTRIB sedaRSL6J

/data/ ATTRIB sedaRSL6J

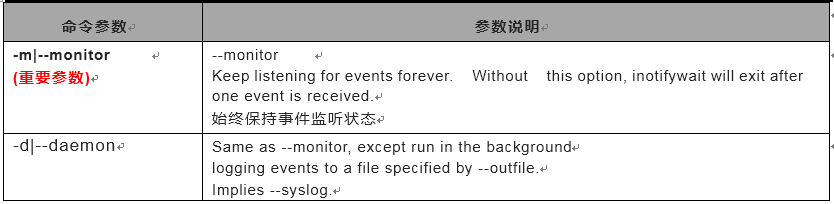
/data/ CLOSE\_NOWRITE,CLOSE oldboy.txt

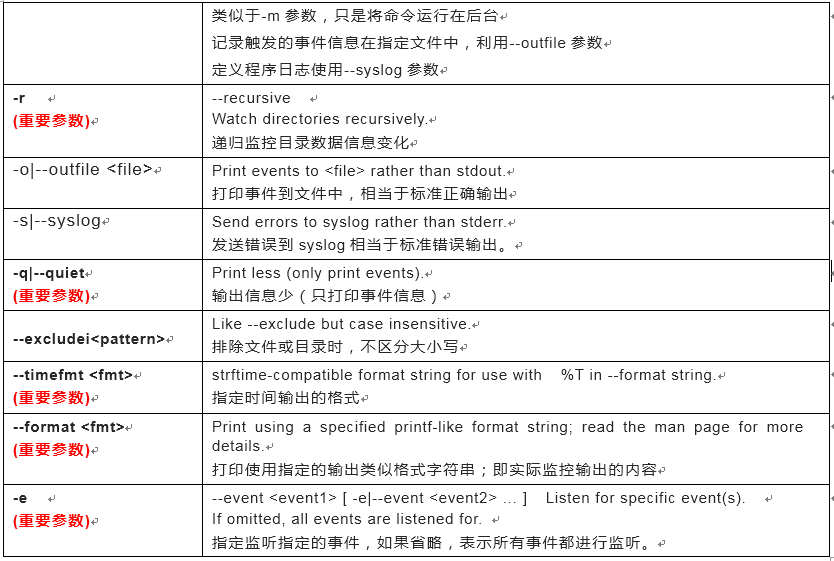
/data/ CLOSE\_WRITE,CLOSE sedaRSL6J

/data/ MOVED\_FROM sedaRSL6J

/data/ MOVED\_TO oldboy.txt

#### 事件监控参数说明





inotifywait命令相关参数说明：

-m|--monitor 始终保持事件监听状态

-r 递归监控目录数据信息变化

-q|--quiet 输出信息少（只打印事件信息） 2>/dev/null

--timefmt <fmt> 指定时间输出的格式

--format <fmt> 打印使用指定的输出类似格式字符串；即实际监控输出的内容

-e 指定监听指定的事件，如果省略，表示所有事件都进行监听。

#### 删除目录逻辑

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

02/09/17 03:21 /data/dir01/123.txt DELETE

02/09/17 03:21 /data/dir01/1234.txt DELETE

02/09/17 03:21 /data/dir01 DELETE,ISDIR

说明：先删除目录下面内容，最后删除目录本身

#### inotify监控命令用法

inotifywait -mrq /data ---简单用法

inotifywait -mrq --timefmt '%d/%m/%y %H:%M' --format '%T %w%f' -e create /backup --- 指定监控输出格式用法

[root@nfs01 ~]# ll /proc/sys/fs/inotify/

total 0

-rw-r--r-- 1 root root 0 Sep 2 01:57 max\_queued\_events

-rw-r--r-- 1 root root 0 Sep 2 01:57 max\_user\_instances

-rw-r--r-- 1 root root 0 Sep 2 01:57 max\_user\_watches

max\_queued\_events：设置inotify实例事件（event）队列可容纳的事件数量

max\_user\_watches： 设置inotifywait或inotifywatch命令可以监视的文件数量（单进程）

max\_user\_instances：设置每个用户可以运行的inotifywait或inotifywatch命令的进程数

以上参数文件默认数值信息：

[root@nfs01 inotify]# cat max\_queued\_events

16384

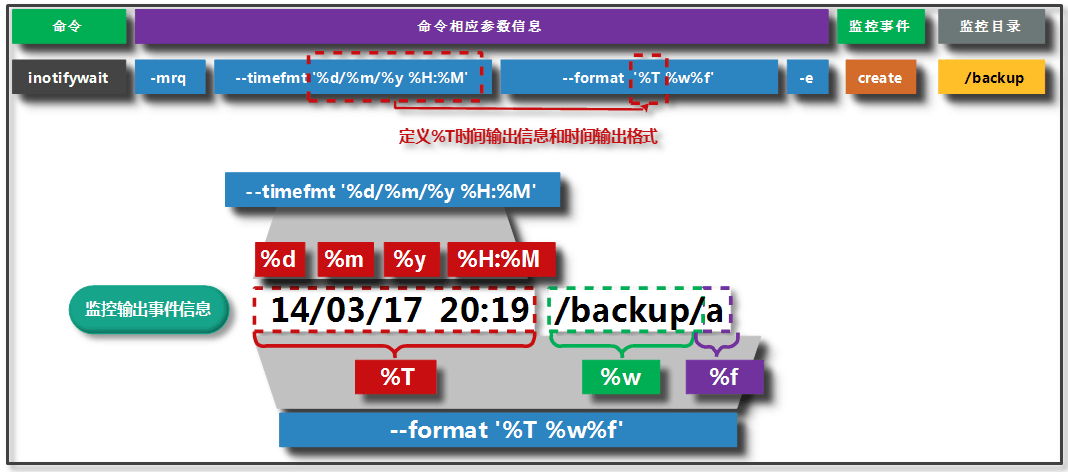
[root@nfs01 inotify]# cat max\_user\_watches

8192

[root@nfs01 inotify]# cat max\_user\_instances

128

说明：以上文件说明信息可以通过man proc获取得到



[root@nfs01 inotify]# inotifywait -mrq --timefmt '%d/%m/%y %H:%M' --format '%T %w%f 事件信息:%e' /data

02/09/17 03:02 /data/oldgirl02.txt 事件信息:CREATE

02/09/17 03:02 /data/oldgirl02.txt 事件信息:OPEN

02/09/17 03:02 /data/oldgirl02.txt 事件信息:ATTRIB

02/09/17 03:02 /data/oldgirl02.txt 事件信息:CLOSE\_WRITE,CLOSE

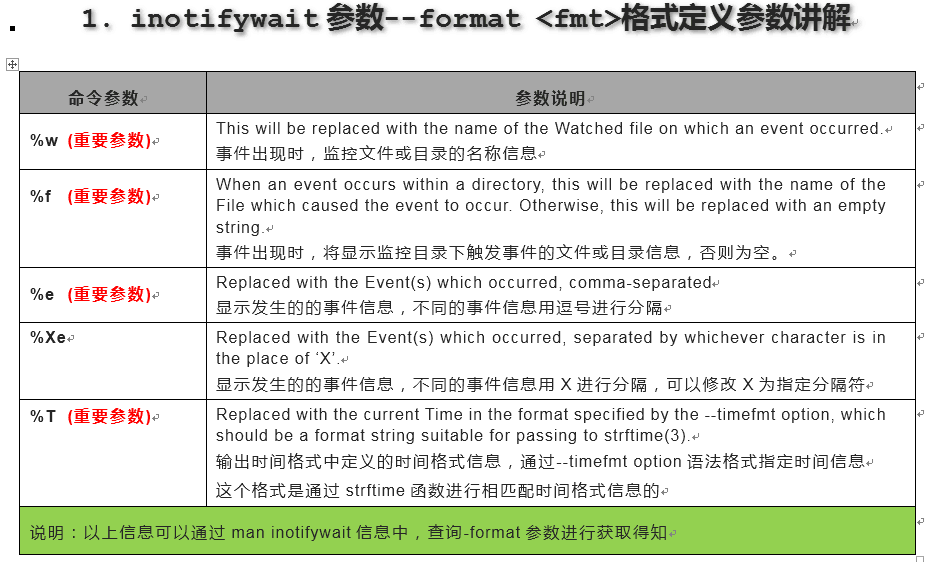
^C

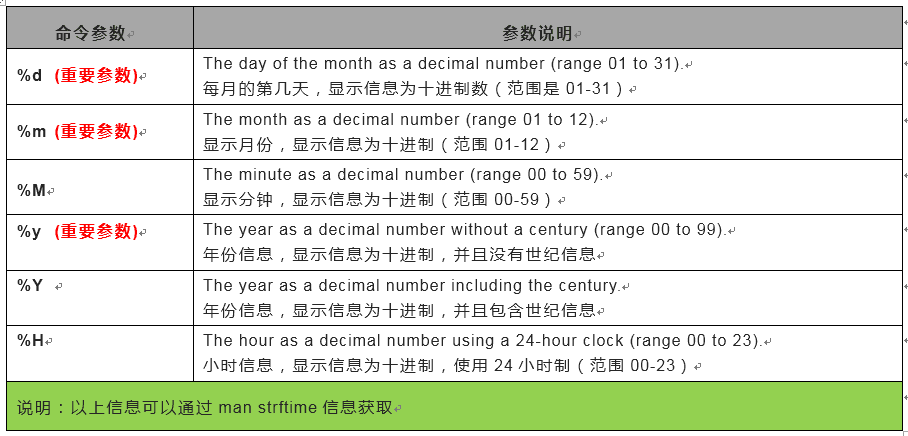
[root@nfs01 inotify]# inotifywait -mrq --timefmt '%d/%m/%y %H:%M' --format '%T %w%f' -e create /data

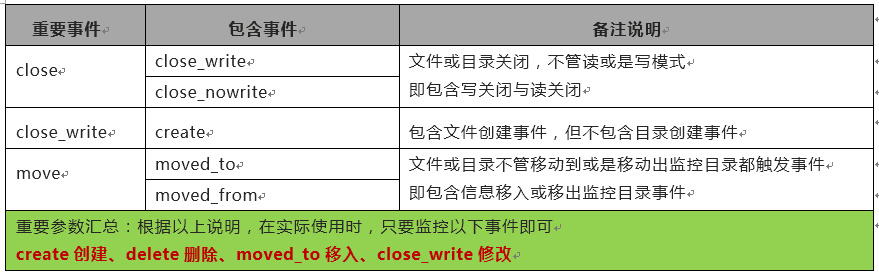
^C

[root@nfs01 inotify]# inotifywait -mrq --timefmt '%d/%m/%y %H:%M' --format '%T %w%f 事件信息:%e' -e create /data

02/09/17 03:03 /data/oldgirl03.txt 事件信息:CREATE







### 第三个里程碑，编写inotify+rsync结合脚本

#### inotify使用方法

inotifywait -mrq /data --format '%w%f' -e create,delete,close\_write,moved\_to

#### rsync使用方法

rsync -avz $file\_info rsync\_backup@172.16.1.41::backup --password-file=/etc/rsync.password

#### 脚本

#!/bin/bash

###########

inotifywait -mrq /data --format '%w%f' -e create,delete,close\_write,moved\_to|\

while read line

do

rsync -az --delete /data/ rsync\_backup@172.16.1.41::backup --password-file=/etc/rsync.password

done

inotifywait -mrq /data --format '%w%f' -e create,delete,close\_write,moved\_to|\

while read line

do

echo $line

done

让脚本程序在后台运行的方法：

1. sh inotify.sh &

2. nohup sh inotify.sh &

另一种脚本

[root@web scripts]# cat inotify1.sh

#!/bin/sh

cmd="/usr/local/inotify-tools/bin/inotifywait"

$cmd -mrq --format '%w%f' -e create,close\_write,delete /backup|\

while read line

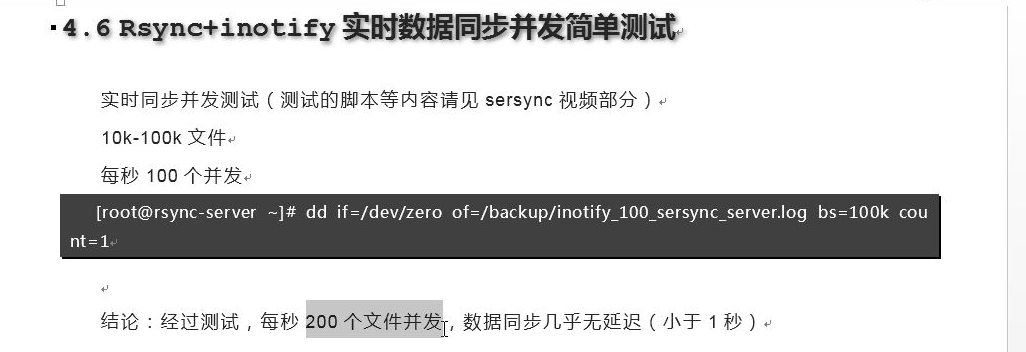
do

[ ! -e "$line" ] && cd /backup &&\

rsync -az --delete ./ rsync\_backup@10.0.0.8::backup --password-file=/etc/rsync.password && continue

rsync -az --delete $line rsync\_backup@10.0.0.8::backup --password-file=/etc/rsync.password

done

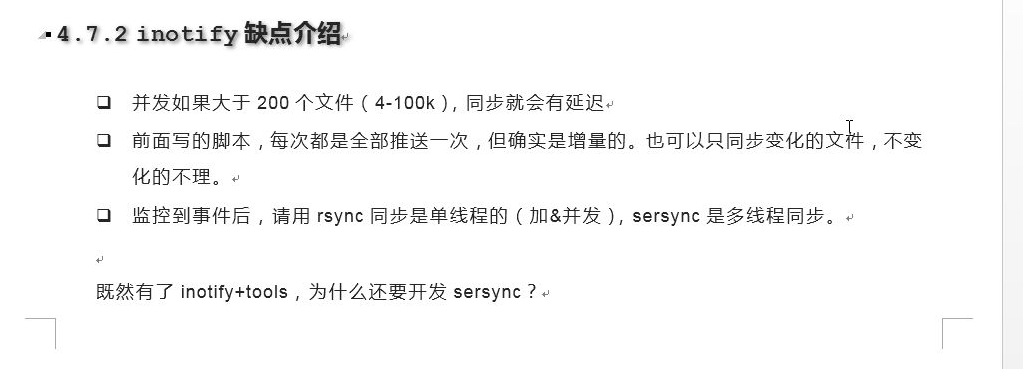


#### inotify 优化

[root@inotify\_server inotify]# echo “50000000” >/proc/sys/fs/inotify/max\_user\_watches

[root@inotify\_server inotify]# echo “50000000” >/proc/sys/fs/inotify/max\_queued\_events

说明：以上配置重启可能会失效，需要放置在rc.local文件中



#### 快捷命令：

ctrl+z --- 暂停运行的进程

jobs --- 检查放入后台运行的进程信息

bg/fg --- bg表示后台运行进程 fg表示前台运行进程

说明：杀手三人组只能杀死运行中的进程，不能杀死暂停的进程

pkill -f "inotify.sh" --- -f 参数表示可以把进程中的任何字符串都变为进程名，进行杀死

## sersync实时同步

### 第一个里程碑：安装部署软件

sersync软件下载地址

https://github.com/wsgzao/sersync

#### 上传软件到服务器上

直接拉过去

#### 解压软件包

unzip sersync\_installdir\_64bit.zip

#### 将软件包移动到指定目录中

mv sersync/ /usr/local/

### 第二个里程碑：编写软件配置文件（难点）

#### 对配置文件进行备份

cp confxml.xml{,.bak}

#### 编写配置文件

/user/local/ sersync/ conf/ confxml.xml

6 <filter start="false">

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

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

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

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

11 </filter>

说明：表示排除同步数据

12 <inotify>

13 <delete start="true"/>

14 <createFolder start="true"/>

15 <createFile start="false"/>

16 <closeWrite start="true"/>

17 <moveFrom start="true"/>

18 <moveTo start="true"/>

19 <attrib start="false"/>

20 <modify start="false"/>

21 </inotify>

说明：指定监控的事件信息

需要修改的内容

24 <localpath watch="/data">

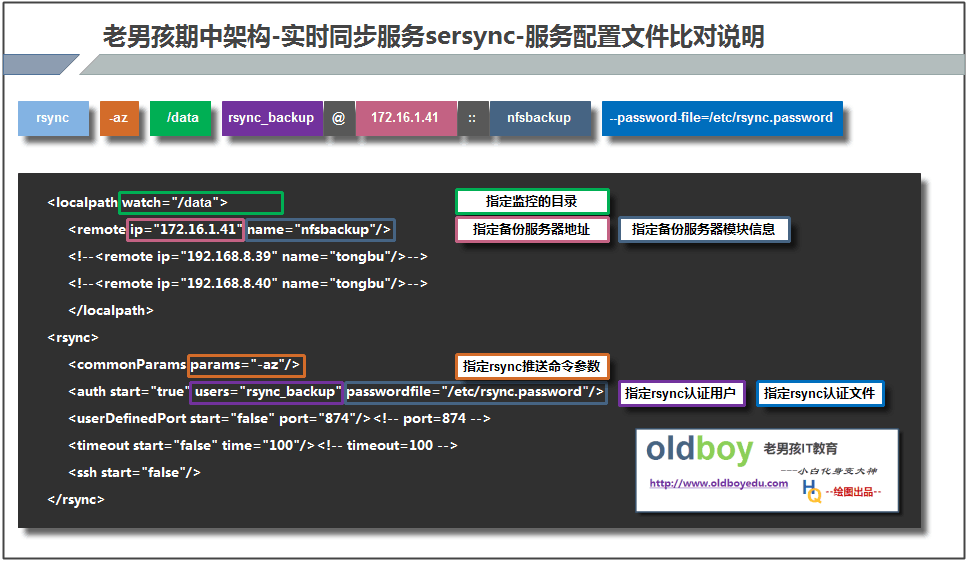
25 <remote ip="172.16.1.41" name="backup"/>

26 <!--<remote ip="192.168.8.39" name="tongbu"/>-->

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

28 </localpath>

说明：定义监控的目录信息，指定推送的备份服务器IP地址与模块命令



29 <rsync>

30 <commonParams params="-az"/>

31 <auth start="true" users="rsync\_backup" passwordfile="/etc/rsync.password"/>

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

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

34 <ssh start="false"/>

35 </rsync>

说明：定义rsync数据推送时的参数信息

### 第三个里程碑：启动sersync软件服务

#### 给sersync 加上执行权限

[root@nfs01 sersync]# cd /usr/local/sersync/bin/

[root@nfs01 bin]# ll

total 1768

-rwxr-xr-x 1 root root 1810128 Oct 26 2011 sersync

[root@nfs01 bin]# chmod +x sersync

#### 启动

[root@nfs01 bin]# ./sersync -h （绝对路径为 /usr/local/sersync/bin/）

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

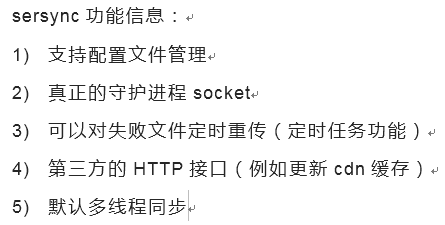
不加-m参数，则默认执行同步程序

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

#### 启动时添加参数（根据需要添加参数）

在/usr/local/sersync/bin/下执行

[root@nfs01 bin]# ./sersync -do /usr/local/sersync/conf/confxml.xml



### inotify-tools参考资料：

https://github.com/rvoicilas/inotify-tools/wiki

http://www.ibm.com/developerworks/cn/linux/l-inotifynew/

inotify--Linux 2.6内核中的文件系统变化通知机制

## NFS弃用方案

