

sersync+rsync 原理及部署

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）具体某一个文件或者某一个目录的名字，然后使用 rsync 同步的时候，只同步发生变化的文件或者目录

二、rsync+inotify-tools 与 rsync+sersync 架构的区别？

- 1、rsync+inotify-tools
 - a、inotify 只能记录下被监听的目录发生了变化（增，删，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；
 - b、rsync 在同步的时候，并不知道具体是哪个文件或目录发生了变化，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
 - a、sersync 可以记录被监听目录中发生变化的（增，删，改）具体某个文件或目录的名字；
 - b、rsync 在同步时，只同步发生变化的文件或目录（每次发生变化的数据相对整个同步目录数据来说很小，rsync 在遍历查找对比文件时，速度很快），因此效率很高。

总结：

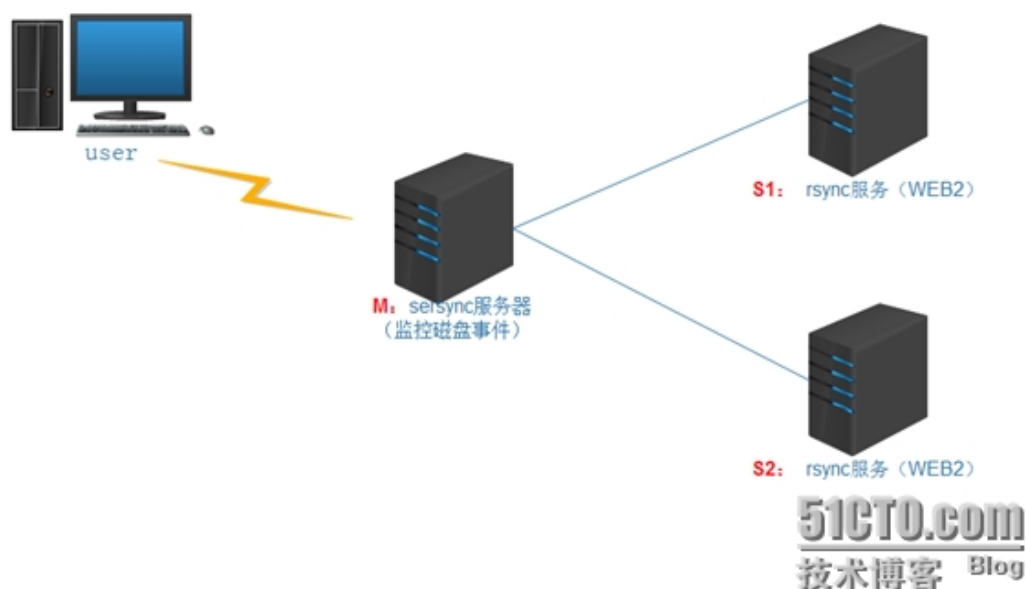
当同步的目录数据量不大时，建议使用 rsync+inotify

当同步的目录数据量很大时（几百 G 甚至 1T 以上）文件很多时，建议使用 rsync+sersync

二、sersync 安装配置

2.1 sersync 同步逻辑图

当前版本的 sersync 依赖于 rsync 进行数据同步；



原理步骤：

1. 在同步服务器（Master）上开启 sersync 服务，sersync 负载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（S1 和 S2）；
3. 需要在主服务器配置 sersync，在同步目标服务器配置 rsync server（注意：是 rsync 服务）

同步原理：

1. 用户实时的往 sersync 服务器（M）上写入更新文件数据；
2. 此时需要在同步主服务器（M）上配置 sersync 服务；
3. 在 S1 和 S2 上开启 rsync 守护进程服务，以同步拉取来自 sersync 服务器（M）上的数据；通过 rsync 的守护进程服务后可以发现，实际上 sersync 就是监控本地的数据写入或更新事件；然后，在调用 rsync 客户端的命令，将写入或更新事件对应的文件通过 rsync 推送到目标服务器（S1 和 S2），如此简单；

2.2 安装环境准备

1.系统资源列表

角色	服务器配置	操作系统版本	IP	机器名
sersync 服务 (M)	VM	CentOS6.6	172.16.1.28	sersync
rsync 服务 (S1)	VM	CentOS6.6	172.16.1.25	WEB1
rsync 服务 (S2)	VM	CentOS6.6	172.16.1.26	WEB2

2.检查系统环境

```
1[root@web ~]# cat/etc/redhat-release
```

```
2CentOS release 6.6 (Final)
```

```
3[root@web ~]# uname -r
```

```
42.6.32-504.el6.x86_64
```

```
5[root@web ~]# uname -m
```

```
6x86_64
```

2.3 配置同步服务器

1.slave 上部署 rsync 服务

2. 升级 rsync 到 3.0 版本

```
1[root@web1 ~]# rsync --version|head -2
```

```
2rsync version 3.0.6 protocol version 30
```

```
3Copyright (C) 1996-2009 byAndrew Tridgell, Wayne Davison, and others.
```

3.部署 rsync 服务

确定 S1 和 S2 服务器版本是最新的，这看下多台目标服务器（S1，S2）上配置如下：

```
1 [root@web1 ~]# vim /etc/rsyncd.conf
```

```
2 #Rsync server
```

```
3 uid = root
```

```
4 gid = root
```

```
5 use chroot = no # 安全相关
```

```
6 max connections = 2000 # 并发连接数
```

```

7 timeout = 600                # 超时时间 (秒)
8 pid file = /var/run/rsyncd.pid    # 指定 rsync 的 pid 目录
9 lock file = /var/run/rsync.lock   # 指定 rsync 的锁文件【重要】
10 log file = /var/log/rsyncd.log    # 指定 rsync 的日志目录
11 ignore errors
12 read only = false
13 list = false
14 hosts allow = 172.16.1.0/24
15 hosts deny = 0.0.0.0/32
16 auth users = rsync_backup
17 secrets file = /etc/rsync.password
18 #####
19 [www]                        # 模块
20 comment = www
21 path = /data/www/
22 #####
23 [bbs]
24 comment = bbs
25 path = /data/bbs/
26 #####
27 [blog]
28 comment = blog
29 path = /data/blog/
30 #rsync_config_____end

```

特别提示：S1，S2 的机器 同时部署上述服务；

上面 rsync 服务的配置文件，表面允许 sersync 主服务器（ip：172.16.1.28）访问 rsync 同步模块名为[www][bbs][blog] 将同步过来的文件分别放入对应的 path 指定的目录/data/{www,bbs,blog}下面；如果有多台目标服务器，则每一台都需要进行类似的 rsync 服务配置，上面的 uid、gid 要换成您服务器的相应的同步用户；注意，rsync 服务账户（本文用的是 root）要有对被同步目录（/data/）的写入更新权限；

4.创建 rsync 同步密码文件，并设置权限为 600

```

1[root@web1 ~]# echo "rsync_backup:liubl">/etc/rsync.password
2[root@web1 ~]# chmod 600 /etc/rsync.password
3[root@web1 ~]# ll /etc/rsync.password
4-rw-----. 1 root root 19Jun  3 18:19 /etc/rsync.password
5[root@web1 ~]# cat /etc/rsync.password
6rsync_backup:liubl

```

5.启动守护进程，并写入开机自启动

```

1 [root@web1 ~]# rsync --daemon
2 [root@web1 ~]# lsof -i:873
3 COMMAND PID USER FD  TYPE  DEVICE SIZE/OFF NODE NAME
4 rsync   1070 root  4u  IPv4 17190189   0t0  TCP *:rsync (LISTEN)
5 [root@web1 ~]#
6 [root@web1 ~]# netstat -nulp | grep rsync

```

```
7 tcp    0    0 0.0.0.0:873      0.0.0.0:*        LISTEN   1070/rsync
```

8 设置开机自启动【写入到/etc/rc.local 里面】

```
9 [root@web1 ~]# vim /etc/rc.local
```

```
10# rsync server progress
```

```
11/usr/bin/rsync --daemon
```

6. 创建相关待同步的目录

```
1mkdir -p /data/{www,bbs,blog}
```

```
2tree /data
```

3提示：此步骤在 S1，S2 都要执行，否则 rsync 服务会因为没有 PATH 路径而无法启动

2.4Master 上配置 rsync 客户端

在 master 上配置 rsync 客户端相关权限认证：

```
1[root@web ~]# echo "liubl">/etc/rsync.password
```

```
2[root@web ~]# chmod 600 /etc/rsync.password
```

```
3[root@web ~]# ll/etc/rsync.password
```

```
4-rw-----. 1 root root 19Jun  5 05:57 /etc/rsync.password
```

```
5[root@web ~]# cat /etc/rsync.password
```

```
6liubl
```

2.4.1master 上手工测试 rsync 同步情况

特别提示：此步非常关键，如果测试不成功，后面的 sersync 配好了也不会同步数据；

1) 分别创建待同步数据

```
1 [root@web ~]# mkdir -p /data/{www,bbs,blog}
```

```
2 [root@web ~]# touch /data/www/www.log /data/bbs/bbs.log/data/blog/blog.log
```

```
3 [root@web ~]# tree /data/
```

```
4 /data/
```

```
5 |— bbs
```

```
6 |   └─ bbs.log
```

```
7 |— blog
```

```
8 |   └─ blog.log
```

```
9 └─ www
```

```
10   └─ www.log
```

```
11
```

```
12 directories, 3 files
```

2) 执行同步命令

```
# rsync-avzP /data/www/ rsync_backup@172.16.1.25::www/--password-  
file=/etc/rsync.password  
sending incremental file list
```

```
sent 38 bytes received 8 bytes 92.00 bytes/sec
```

```
total size is 0 speedup is 0.00
```

```
rsync-avzP /data/www/ rsync_backup@172.16.1.25::www/--password-  
file=/etc/rsync.password
```

```
rsync-avzP /data/www/ rsync_backup@172.16.1.26::www/--password-  
file=/etc/rsync.password
```

```
rsync-avzP /data/bbs/ rsync_backup@172.16.1.26::bbs/--password-file=/etc/rsync.password
```

```
rsync-avzP /data/bbs/ rsync_backup@172.16.1.25::bbs/--password-file=/etc/rsync.password
rsync-avzP /data/blog/ rsync_backup@172.16.1.25::blog/--password-file=/etc/rsync.password
rsync-avzP /data/blog/ rsync_backup@172.16.1.26::blog/--password-file=/etc/rsync.password
```

提示：

在后面进行部署 sersync 之前，sersync 主服务器上必须要确保手工可以把文件推送到 S1，S2 上，这样后续 sersync 才能调用这些命令来自动推送

在推送前关闭 iptables

3) 推送的命令是在 master 端（也就是 sersync 服务器上）操作的，同步后查看 S1，S2

```
1 [root@web1 ~]# tree /data/
```

```
2 /data/
```

```
3 |— bbs
```

```
4 |   |— bbs.log
```

```
5 |— blog
```

```
6 |   |— blog.log
```

```
7 |— www
```

```
8 |   |— www.log
```

```
9
```

```
10 directories, 3 files
```

2.5 Master 上开始部署 sersync 服务

1、下载 sersync

在 google code 下载 sersync 的可执行文件版本，里面有配置文件与可执行文件，这用

```
1 mkdir -p /appliance/tools
```

```
2 cd /appliance/tools
```

```
3 wget https://sersync.googlecode.com/files/sersync2.5.4_64bit_binary_stable_final.tar.gz
```

4 【有时下载失败，所有要本地留存才行】

```
5 [root@web ~]# tar fxzsersync2.5.4_64bit_binary_stable_final.tar.gz -C /usr/local/
```

```
6 [root@web ~]# cd /usr/local/
```

```
7 [root@cache local]# mv GNU-Linux-x86 sersync
```

```
8 [root@cache local]# tree sersync/
```

```
9 sersync/
```

```
10 |— confxml.xml # 配置文件
```

```
11 |— sersync2 # 二进制文件【启动 sersync 使用】
```

```
12
```

```
13 directories, 2 files
```

2、配置 sersync

```
1 [root@cache local]# cp sersync/confxml.xml sersync/confxml.xml.$(date +%F)
```

```
2 [root@cache local]# ll sersync/confxml.xml
```

```
3 -rwxr-xr-x. 1 root root 2214 Oct 26 2011 sersync/confxml.xml
```

```
4 [root@cache local]# ll sersync/confxml.xml*
```

```
5 -rwxr-xr-x. 1 root root 2214 Oct 26 2011 sersync/confxml.xml
```

```
6 -rwxr-xr-x. 1 root root 2214 Jun 5 06:38 sersync/confxml.xml.2015-06-05
```

更改优化 sersync 配置文件：

a) 修改 24--28 行

```

    <localpathwatch="/opt/tongbu"> # 定义本地要同步的目录
    <remote ip="127.0.0.1" name="tongbu1"/>
    <!--<remoteip="192.168.8.39" name="tongbu"/>--> # 同步到哪台机器上 tongbu 模
块 rsync 端模块名字
    <!--<remoteip="192.168.8.40" name="tongbu"/>--> # 同步到哪台机器上 tongbu
模块
  </localpath>

```

修改后的内容为：

```

    <localpathwatch="/data/www">
    <remoteip="172.16.1.25" name="www"/>
    <remoteip="172.16.1.26" name="www"/>
  </localpath>
  <!--##### -->
  <localpathwatch="/data/bbs">
    <remoteip="172.16.1.25" name="bbs"/>
    <remoteip="172.16.1.26" name="bbs"/>
  </localpath>
  <!--##### -->
  <localpathwatch="/data/blog">
    <remote ip="172.16.1.25" name="blog"/>
    <remoteip="172.16.1.26" name="blog"/>
  </localpath>
  <!--##### -->

```

提示：此步 watch="/data/blog"就是定义服务端待同步的目录，和目标服务器的模块 name="blog"

b) 修改 31--34 行，认证部分【rsync 密码认证】

```

1  <rsync>
2    <commonParamsparams="-artuz"/>
3    <auth start="false" users="root" passwordfile="/etc/rsync.pas"/>
4    <userDefinedPortstart="false" port="874"/><!-- port=874 -->
5    <timeoutstart="false" time="100"/><!-- timeout=100 -->
6    <sshstart="false"/>
7  </rsync>

```

修改后的内容如下：

```

<rsync>
  <commonParamsparams="-artuz"/>
  <auth start="true" users="rsync_backup" passwordfile="/etc/rsync.password"/>
  <userDefinedPortstart="false" port="874"/><!-- port=874 -->
  <timeout start="true" time="100"/><!-- timeout=100 -->
  <sshstart="false"/>
</rsync>

```

***修改内容为 rsync 的密码文件以及 同步所使用的账号类似：

```

rsync -avzP /data/www/rsync_backup@172.16.1.25::www/ --password-
file=/etc/rsync.password

```

c) 修改 37 行

```
<failLogpath="/tmp/rsync_fail_log.sh"timeToExecute="60"/><!--default every 60mins execute once-->
```

修改后如下：

```
<failLog path="/usr/local/sersync/logs/rsync_fail_log.sh"timeToExecute="60"/><!--default every 60mins execute once-->
```

2# 当同步失败后，日志记录到/usr/local/sersync/logs/rsync_fail_log.sh 文件中，并且每 60 分钟对失败的 log 进行重新同步

修改后的完整配置文件为：

```
1 [root@cache local]# cat sersync/confxml.xml
2 <?xmlversion="1.0" encoding="ISO-8859-1"?>
3 <headversion="2.5">
4   <host hostip="localhost"port="8008"></host>
5   <debug start="false"/>
6   <fileSystem xfs="false"/>
7   <filter start="false">
8     <excludeexpression="(.)\.svn"></exclude>
9     <excludeexpression="(.)\.gz"></exclude>
1    <excludeexpression="^info/*"></exclude>
0    <excludeexpression="^static/*"></exclude>
1  </filter>
1  <inotify>
1    <delete start="true"/>
2    <createFolderstart="true"/>
1    <createFilestart="false"/>
3    <closeWritestart="true"/>
1    <moveFromstart="true"/>
4    <moveTo start="true"/>
1    <attrib start="false"/>
5    <modify start="false"/>
1  </inotify>
6
1  <sersync>
7    <localpathwatch="/data/www">
1      <remoteip="172.16.1.25" name="www"/>
8      <remoteip="172.16.1.26" name="www"/>
1    </localpath>
9    <!--##### -->
2    <localpathwatch="/data/bbs">
0      <remoteip="172.16.1.25" name="bbs"/>
2      <remoteip="172.16.1.26" name="bbs"/>
1    </localpath>
2    <!--##### -->
2    <localpathwatch="/data/blog">
```

```

    <remoteip="172.16.1.25" name="blog"/>
    <remoteip="172.16.1.26" name="blog"/>
</localpath>
<!-- ##### -->
<rsync>
    <commonParamsparams="-artuz"/>
    <auth start="true"users="rsync_backup"passwordfile="/etc/rsync.password"/>
    <userDefinedPortstart="false" port="874"/><!-- port=874 -->
    <timeout start="true"time="100"/><!-- timeout=100 -->
    <sshstart="false"/>
</rsync>
    <failLogpath="/usr/local/sersync/logs/rsync_fail_log.sh"timeToExecute="60"/><!--default e
very 60mins execute once-->
    <crontab start="false"scheduled="600"><!--600mins-->
        <crontabfilterstart="false">
            <excludeexpression="*.php"></exclude>
            <excludeexpression="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 su
ffix-->
    <filter start="false">
        <include expression="(.*).php"/>
        <includeexpression="(.*).sh"/>
    </filter>
</plugin>

<plugin name="socket">
    <localpathwatch="/opt/tongbu">
        <deshostip="192.168.138.20" port="8009"/>
    </localpath>
</plugin>
<plugin name="refreshCDN">
    <localpathwatch="/data0/htdocs/cms.xoyo.com/site/">
        <cdninfodomainname="ccms.chinacache.com" port="80"username="xxx" passwd="xxx
x"/>
        <sendurlbase="http://pic.xoyo.com/cms"/>
        <regexurlregex="false"match="cms.xoyo.com/site([a-zA-Z0-9]*).xoyo.com/images"/>
    </localpath>
</plugin>

```


</head>

3、开启 sersync 守护进程同步数据

启动命令

```
1[root@web ~]# /usr/local/sersync/sersync2 -d -r -o /usr/local/sersync/confxml.xml
```

2配置 sersync 环境变量

```
3[root@web ~]# echo "PATH=$PATH:/usr/local/sersync/">>/etc/profile
```

```
4[root@web ~]# source /etc/profile
```

```
5[root@web ~]# sersync2
```

启动命令后返回结果如下为正常：

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 remoteservers before the sersync work

option: -o config xml name : /usr/local/sersync/confxml.xml

daemon thread num: 10

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 12 = 1(primary thread) + 1(fail retry thread) + 10(daemon sub threads)

Max threads numbers is: 32 = 12(Thread pool nums) + 20(Sub threads)

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

chmod: cannot access /usr/local/sersync/logs/rsync_fail_log.sh: No such file or directory

rsync the directory recursively to the remote servers once

working please wait...

execute command: cd /data/www && rsync -artuz -R --delete ./ --timeout=100 rsync_backup@172.16.1.25::www --password-file=/etc/rsync.password >/dev/null 2>&1

run the sersync:

watch path is: /data/www

补充：多实例情况

1、配置多个 confxml.xml 文件（比如：www、bbs、blog....等等）

2、根据不同的需求同步对应的实例文件

```
1 /usr/local/sersync/sersync2 -d -o /usr/local/sersync/www_confxml.xml
```

```
2 /usr/local/sersync/sersync2 -d -o /usr/local/sersync/bbs_confxml.xml
```

```
3 /usr/local/sersync/sersync2 -d -o /usr/local/sersync/blog_confxml.xml
```

```

4 # 多实例初始化同步命令 :
5 /usr/local/sersync/sersync2 -r-d -o /usr/local/sersync/www_confxml.xml
6 /usr/local/sersync/sersync2 -r-d -o /usr/local/sersync/bbs_confxml.xml
7 /usr/local/sersync/sersync2 -r-d -o /usr/local/sersync/blog_confxml.xml
8
9 /bin/cp /etc/rc.local/etc/rc.local_$(data +%F)
10cat>>/etc/rc.local<<EOF
11#sync data to 25 26
12/usr/local/sersync/sersync2 -d -o /usr/local/sersync/www_confxml.xml
13/usr/local/sersync/sersync2 -d -o /usr/local/sersync/bbs_confxml.xml
14/usr/local/sersync/sersync2 -d -o /usr/local/sersync/blog_confxml.xml
15EOF

```

压测：写入 10K 个文件批量同步测试结果：

```

# for n in `seq 10000`;do echodddd>www/$n.txt;done
# ps -ef |grep rsync
root  17283  1 0 Jun05 ?    00:00:02 /usr/local/sersync/sersync2 -d-r -o /usr/local/sersync/c
onfxml.xml
root  19363  1 0 Jun05 ?    00:00:01 /usr/local/sersync/bin -d -
o/usr/local/sersync/confxml.xml
root  19394  1 0 Jun05 ?    00:00:01 /usr/local/sersync/bin -r -d-o /usr/local/sersync/confx
ml.xml
root  19414  1 0 Jun05 ?    00:00:01 /usr/local/sersync/bin -r -d-o /usr/local/sersync/confx
ml.xml
root  29484 17283 0 01:33 ?    00:00:00 sh -c cd /data/www &&rsync -artuz -R --
timeout=100"/.395.txt" rsync_backup@172.16.1.25::www--password-file=/etc/rsync.password
>/dev/null 2>&1
root  29487 29484 0 01:33 ?    00:00:00 rsync -artuz -R --timeout=100/.395.txt rsync_backu
p@172.16.1.25::www --password-file=/etc/rsync.password
root  29490 17283 0 01:33 ?    00:00:00 sh -c cd /data/www &&rsync -artuz -R --
timeout=100"/.396.txt" rsync_backup@172.16.1.25::www--password-file=/etc/rsync.password
>/dev/null 2>&1

```

提示：我们发现本地已经写完了 10000 个，但是同步的线程，依然在同步；甚至才同步了 1000 多个

2.6 检查节点是否同步成功

WEB1 同步查看对比	WEB2 同步查看对比
[root@web1 data]# du /data/www/ 26944 /data/www/	[root@web2 data]# du /data/www/ 26880 /data/www/
[root@web1 data]# du /data/www/ 26964 /data/www/	[root@web2 data]# du /data/www/ 26908 /data/www/
[root@web1 data]# du /data/www/ 27024 /data/www/	[root@web2 data]# du /data/www/ 26940 /data/www/

[root@web1 data]# du /data/www/ 27036 /data/www/	[root@web2 data]# du /data/www/ 26960 /data/www/
每秒同步 20--30 个文件	每秒同步 20-30 个文件

三、命令参数说明

Sersync 参数	说明
./sersync -r	-r 参数作用是：开启实时监控的之前对主服务器目录与远程目标机器的目录进行一次整体同步；如果需要将 sersync 运行前，主服务器目录下已经存在的所有文件或目录全部同步到远端，则要以 -r 参数运行 sersync，将本地与远程整体同步一次； 提别说明：如果设置了过滤器，即在 xml 文件中，filter 为 true，则暂时不能使用-r 参数进行整体同步；
./sersync -o xx.xml	不指定 -o 参数：sersync 使用 sersync 可执行文件目录下的默认配置文件 confxml.xml 指定 -o 参数：可以指定多个不同的配置文件，从而实现 sersync 多进程多实例的数据同步
./sersync -n num	-n 参数为：指定默认的线程池的线程总数； 例如：./sersync -n 5 则指定线程总数为 5，如果不指定，默认启动线程池数量是 10，如果 cpu 使用过高，可以通过该参数调低，如果机器配置较高，可以调高默认的线程总数，提升同步效率；
./sersync -d	-d 参数为：后台服务，通常情况下使用 -r 参数对本地到远端整体同步一遍后，在后台运行此参数启动守护进程实时同步；在第一次整体同步时，-d 和 -r 参数经常会联合使用；
./sersync -m pluginName	-m 参数：不进行同步，只运行插件 ./sersync -m pluginName 例如：./sersync -m command，则在监控到事件后，不对远程目标服务器进行同步，而是直接运行 command 插件
组合命令使用说明：	
-n 8 -o liubl.xml -r -d	多个参数可以配合使用，例如：./sersync -n 16 -o config.xml -r -d 表示设置线程池工作线程为 16 个，指定 liubl.xml 作为配置文件，在实时监控前做一次整体同步，以守护进程方式在后台运行；
./sersync --help	很遗憾，它没有查看帮助（需要的话 2 条路，要么看源代码，要么自测求验证）

四、sersync 服务配置文件参数详解

4.1 初始的配置文件

sersync 可选功能是通过 xml 配置文件来实现的，基本配置文件如下：

```
1 [root@cache sersync]# cat confxml.xml.2015-06-05
2 1 <?xml version="1.0" encoding="ISO-8859-1"?>
3 2 <head version="2.5">
4 3 <host hostip="localhost" port="8008"></host>
```

```

4  <debug start="false"/>
5  <fileSystem xfs="false"/>
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>
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>
22
23 <sersync>
24     <localpath watch="/opt/tongbu">
25         <remoteip="127.0.0.1" name="tongbu1"/>
26         <!--<remoteip="192.168.8.39" name="tongbu"/>-->
27         <!--<remoteip="192.168.8.40" name="tongbu"/>-->
28     </localpath>
29     <rsync>
30         <commonParamsparams="-artuz"/>
31         <auth start="false"users="root" passwordfile="/etc/rsync.pas"/>
32         <userDefinedPortstart="false" port="874"/><!-- port=874 -->
33         <timeoutstart="false" time="100"/><!-- timeout=100 -->
34         <sshstart="false"/>
35     </rsync>
36     <failLog path="/tmp/rsync_fail_log.sh"timeToExecute="60"/><!--default every 60min
s execute once-->
37     <crontab start="false"scheduled="600"><!--600mins-->
38         <crontabfilterstart="false">
39             <excludeexpression="*.php"></exclude>
40             <excludeexpression="info/*"></exclude>
41         </crontabfilter>
42     </crontab>
43     <plugin start="false" name="command"/>
44 </sersync>
45
46 <plugin name="command">

```

```

47   <param prefix="/bin/sh" suffix="" ignoreError="true"/> <!--prefix /opt/tongbu/mmm.
sh suffix-->
48   <filter start="false">
49       <includeexpression="(.)\.php"/>
50       <includeexpression="(.)\.sh"/>
51   </filter>
52 </plugin>
53
54 <plugin name="socket">
55   <localpath watch="/opt/tongbu">
56       <deshostip="192.168.138.20" port="8009"/>
57   </localpath>
58 </plugin>
59 <plugin name="refreshCDN">
60   <localpath watch="/data0/htdocs/cms.xoyo.com/site/">
61       <cdninfodomainname="ccms.chinacache.com" port="80"username="xxxx" passw
d="xxxx"/>
62       <sendurlbase="http://pic.xoyo.com/cms"/>
63       <regexurlregex="false" match="cms.xoyo.com/site([/a-zA-Z0-
9]*).xoyo.com/images"/>
64   </localpath>
65 </plugin>
66 </head>

```

4.2 xml 配置文件说明

说明：xml 配置文件的注释不用“#”，而是<!-- 中间是注释内容 -->

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

hostip 与 port 是针对插件的保留字段，对于同步功能没有任何作用，保留默认即可；

4.3 Debug 开启开关

```
4   <debug start="false"/>
```

设置为 true，表示开启 debug 模式，会在 sersync 正在运行的控制台打印 inotify 时间与 rsync 同步命令；

4.4 XFS 文件系统开关

```
5   <fileSystem xfs="false"/>
```

对于 xfs 文件系统的用户，需要将这个选项开启，才能使用 sersync 正常工作；

4.5 filter 文件过滤功能

说明：一般情况下，不给客户端添加过滤，如有必要才添加；

```

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>

```

对于大多数应用，可以尝试把 createFile（监控文件事件选项）设置为 false 来提高性能，减

少 rsync 通讯；

因为拷贝文件到监控目录会产生 create 事件与 close_write 事件，所以如果关闭 create 事件，只监控文件拷贝结束时的时间 close_write，同样可以实现文件完整同步；

注意：强将 creatFolder 保持为 true，如果将 createFolder 设为 false，则不会对产生的目录进行监控，该目录下的子文件与子目录也不会被监控；所以除非特殊需要，请开启；默认情况下对创建文件（目录）事件与删除文件（目录）事件都进行监控，如果项目中不需要删除远程目标服务器的文件（目录），则可以将 delete 参数设置为 false，则不对删除事件进行监控；

本文出自“[靠谱儿](http://liubao0312.blog.51cto.com/2213529/1677586)”博客，请务必保留此出处

<http://liubao0312.blog.51cto.com/2213529/1677586>