



链滴

rsync+lsyncd 实现文件实时同步 (自己实践)

作者: [jinggo](#)

原文链接: <https://ld246.com/article/1533785970900>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

参考地址：[rsync+lsyncd实现文件实时同步](#)

参考地址：[lsyncd实时同步搭建指南——取代rsync+inotify](#)

一、环境

lsyncd 10.211.55.22
rsync 10.211.55.21

二、配置rsync服务器

配置rsync以xineted方式运行

```
[root@rsync ~]# yum install rsync -y
[root@rsync ~]# yum install xinetd -y

#修改/etc/xinetd.d/rsync
[root@rsync ~]# vim /etc/xinetd.d/rsync
service rsync
{
    disable      = no      ##将yes改成no
    socket_type  = stream
    wait         = no
    user         = root
    server       = /usr/bin/rsync
    server_args  = --daemon
    log_on_failure += USERID
}
```

```
#启动xinetd服务
[root@rsync ~]# service xinetd start
Starting xinetd: [ OK ]
```

#rsync默认的监听端口是873，查看873号端口是否启动

```
[root@rsync ~]# netstat -tunlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State      PID/Program name
tcp      0      0 0.0.0.0:22              0.0.0.0:*          LISTEN      1247/sshd
tcp      0      0 127.0.0.1:25             0.0.0.0:*          LISTEN      1324/master
tcp      0      0 ::22                   :::*                  LISTEN      1247/sshd
tcp      0      0 ::1:25                 :::*                  LISTEN      1324/master
tcp      0      0 ::873                 :::*                  LISTEN      1561/xinetd
```

创建rsync服务目录和配置文件

```
#创建rsync服务目录
[root@rsync ~]# mkdir /etc/rsyncd
# 创建配置文件
[root@rsync ~]# touch /etc/rsyncd/rsyncd.conf
# 创建密码文件
[root@rsync ~]# touch /etc/rsyncd/rsyncd.secrets
#权限修改
```

```
[root@rsync ~]# chown root:root /etc/rsyncd/rsyncd.secrets
[root@rsync ~]# chmod 600 /etc/rsyncd/rsyncd.secrets      #这里的权限设置必须是600
```

创建用户和密码

```
[root@rsync ~]# echo "rsync:test" >>/etc/rsyncd/rsyncd.secrets
```

创建rsync配置文件

```
# GLOBAL OPTIONS
uid = root
gid = root

use chroot = yes      #这个参数要设置成yes，如果同步的是软连接文件，同步过来后会多一个前
, 导致软连接不能正常使用
read only = no        #我们需要实时同步lsyncd服务器上的资源，这个需要有写权限，或者在模块中
予写权限

#limit access to private LANs
hosts allow=10.211.55.21/255.255.0.0
hosts deny=*
max connections = 5

pid file = /var/run/rsyncd.pid

secrets file = /etc/rsyncd/rsyncd.secrets
#lock file = /var/run/rsync.lock

motd file = /etc/rsyncd/rsyncd.motd

#This will give you a separate log file
log file = /var/log/rsync.log

#This will log every file transferred - up to 85,000+ per user, per sync
transfer logging = yes

log format = %t %a %m %f %b
syslog facility = local3
timeout = 300

# MODULE OPTIONS
[test]
path = /home/syncfile
list=yes
ignore errors
auth users = rsync      #客户端连接过来使用的用户是rsync
comment = welcome to rsync server
```

编辑xinetc的rsync配置文件，添加配置文件路径

```
#添加rsync的配置文件路径
[root@rsync ~]# vim /etc/xinetd.d/rsync
service rsync
{
    disable = no
```

```

socket_type    = stream
wait          = no
user          = root
server        = /usr/bin/rsync
server_args   = --daemon --config=/etc/rsyncd/rsyncd.conf #添加配置文件路径
log_on_failure += USERID
}

#重启xinetd服务
[root@rsync ~]# service xinetd restart
Stopping xinetd:                                     [ OK ]
Starting xinetd:                                     [ OK ]
[root@rsync ~]# netstat -anpt |grep 873
tcp      0      0 ::873          ::*          LISTEN     1586/xinetd

#创建数据目录
[root@rsync ~]# mkdir -p /home/syncfile

```

三、配置lsyncd服务器

```

#安装rsync,lsyncd
[root@lsyncd ~]# rpm -ivh http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
[root@lsyncd ~]# sed -i 's@#b@b@g' /etc/yum.repos.d/epel.repo
[root@lsyncd ~]# sed -i 's@mirrorlist@#mirrorlist@g' /etc/yum.repos.d/epel.repo
[root@lsyncd ~]# yum install rsync lsyncd -y

```

配置lsyncd服务配置文件适用：500+万文件，变动不大

注意：这里配置的方案有很多，我是参考上面两个链接的方法：

#lsyncd.conf配置选项说明：

settings

这里是全局设置，`--`开头表示注释，下面是几个常用选项说明：

- **logfile** 定义日志文件
- **statusFile** 定义状态文件
- **nodaemon=true** 表示不启用守护模式，默认
- **statusInterval** 将lsyncd的状态写入上面的statusFile的间隔，默认10秒
- **inotifyMode** 指定inotify监控的事件，默认是**CloseWrite**，还可以是**Modify**或**CloseWrite or Modify**
- **maxProcesses** 同步进程的最大个数。假如同时有20个文件需要同步，而**maxProcesses = 8**，则能看到有8个rsync进程
- **maxDelays** 累计到多少所监控的事件激活一次同步，即使后面的**delay**延迟时间还未到

sync

这里是定义同步参数，可以继续使用**maxDelays**来重写**settings**的全局变量。一般第一个参数指定**lsyncd**以什么模式运行：**rsync**、**rsyncssh**、**direct**三种模式：

- **default.rsync**：本地目录间同步，使用rsync，也可以达到使用ssh形式的远程rsync效果，或daemon方式连接远程rsyncd进程；

default.direct：本地目录间同步，使用cp、rm等命令完成差异文件备份；

default.rsyncssh：同步到远程主机目录，rsync的ssh模式，需要使用key来认证

- **source** 同步的源目录，使用绝对路径。

- **target** 定义目的地址.对应不同的模式有几种写法：

/tmp/dest：本地目录同步，可用于**direct**和**rsync**模式

172.29.88.223:/tmp/dest：同步到远程服务器目录，可用于**rsync**和**rsyncssh**模式，拼接的命令类似于`/usr/bin/rsync -ltsd --delete --include-from=- --exclude=*` SOURCE TARGET，剩下的就是rsync的内容了，比如指定username，免密码同步

172.29.88.223::module：同步到远程服务器目录，用于**rsync**模式

三种模式的示例会在后面给出。

- **init** 这是一个优化选项，当**init = false**，只同步进程启动以后发生改动事件的文件，原有的目录即有差异也不会同步。默认是**true**

- **delay** 累计事件，等待rsync同步延时时间，默认15秒（最大累计到1000个不可合并的事件）。也是15s内监控目录下发生的改动，会累积到一次rsync同步，避免过于频繁的同步。（可合并的意思是15s内两次修改了同一文件，最后只同步最新的文件）

excludeFrom

排除选项，后面指定排除的列表文件，如

excludeFrom = "/etc/lsyncd.exclude"

如果是简单的排除，可以使用

exclude = LIST

这里的排除规则写法与原生rsync有点不同，更为简单：

- 监控路径里的任何部分匹配到一个文本，都会被排除，例如 `/bin/foo/bar`可以匹配规则**foo**
- 如果规则以斜线 `/`开头，则从头开始要匹配全部
- 如果规则以 `/`结尾，则要匹配监控路径的末尾
- `?`匹配任何字符，但不包括`/`
- `*`匹配0或多个字符，但不包括`/`
- `**`匹配0或多个字符，可以是`/`
- **delete** 为了保持target与source完全同步，Lsyncd默认会**delete = true**来允许同步删除。它除了**false**，还有**startup**、**running**值，请参考 [Lsyncd 2.1.x || Layer 4 Config || Default Behavior](#)。

rsync

（提示一下，**delete**和**exclude**本来都是**rsync**的选项，上面是配置在**sync**中的，我想这样做的原因为了减少rsync的开销）

- **bwlimit** 限速，单位kb/s，与rsync相同（这么重要的选项在文档里竟然没有标出）
- **compress** 压缩传输默认为**true**。在带宽与cpu负载之间权衡，本地目录同步可以考虑把它设为**false**
- **perms** 默认保留文件权限。

- 其它rsync的选项

其它还有rsyncssh模式独有的配置项，如host、targetdir、rsync_path、password_file，见后文示。
。 rsyncOps={"-avz","--delete"}这样的写法在2.1.*版本已经不支持。

lsyncd.conf可以有多个sync，各自的source，各自的target，各自的模式，互不影响。

lsyncd.conf其它模式示例：

```
[root@rsync ~]# cat /etc/rsyncd/rsyncd.conf

# GLOBAL OPTIONS
uid = root
gid = root

use chroot = yes      #这个参数要设置成yes，如果同步的是软连接文件，同步过来后会多一个前
, 导致软连接不能正常使用

read only = no        #我们需要实时同步lsyncd服务器上的资源，这个需要有写权限，或者在模块中
予写权限

#limit access to private LANs
hosts allow=10.211.55.21/255.255.0.0
hosts deny=*
max connections = 5

pid file = /var/run/rsyncd.pid

secrets file = /etc/rsyncd/rsyncd.secrets
#lock file = /var/run/rsync.lock

motd file = /etc/rsyncd/rsyncd.motd

#This will give you a separate log file
log file = /var/log/rsync.log

#This will log every file transferred - up to 85,000+ per user, per sync
transfer logging = yes

log format = %t %a %m %f %b
syslog facility = local3
timeout = 300

# MODULE OPTIONS
[test]
path = /home/syncfile
list=yes
ignore errors
auth users = rsync      #客户端连接过来使用的用户是rsync
comment = welcome to rsync server
```

```

#添加rsync的配置文件路径
[root@rsync ~]# vim /etc/xinetd.d/rsync
service rsync
{
    disable = no
    socket_type  = stream
    wait         = no
    user         = root
    server       = /usr/bin/rsync
    server_args  = --daemon --config=/etc/rsyncd/rsyncd.conf #添加配置文件路径
    log_on_failure += USERID
}

#重启xinetd服务
[root@rsync ~]# service xinetd restart
Stopping xinetd: [ OK ]
Starting xinetd: [ OK ]
[root@rsync ~]# netstat -anpt |grep 873
tcp      0      0 ::873          ::*          LISTEN     1586/xinetd

#创建数据目录
[root@rsync ~]# mkdir -p /data/test

#安装rsync,lsyncd
[root@lsyncd ~]# rpm -ivh http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
[root@lsyncd ~]# sed -i 's@#b@b@g' /etc/yum.repos.d/epel.repo
[root@lsyncd ~]# sed -i 's@mirrorlist@#mirrorlist@g' /etc/yum.repos.d/epel.repo
[root@lsyncd ~]# yum install rsync lsyncd -y

settings {
    logfile = "/usr/local/rsyncd-2.1.5/var/rsyncd.log",
    statusFile = "/usr/local/rsyncd-2.1.5/var/rsyncd.status",
    inotifyMode = "CloseWrite",
    maxProcesses = 8,
}

-- I. 本地目录同步, direct: cp/rm/mv。 适用: 500+万文件, 变动不大
sync {
    default.direct,
    source   = "/tmp/src",
    target   = "/tmp/dest",
    delay = 1
    maxProcesses = 1
}

-- II. 本地目录同步, rsync模式: rsync
sync {
    default.rsync,
    source   = "/tmp/src",
    target   = "/tmp/dest1",
    excludeFrom = "/etc/rsyncd.d/rsync_exclude.lst",
    rsync   = {

```

```

        binary = "/usr/bin/rsync",
        archive = true,
        compress = true,
        bwlimit  = 2000
    }
}

-- III. 远程目录同步, rsync模式 + rsyncd daemon
sync {
    default.rsync,
    source   = "/tmp/src",
    target   = "syncuser@172.29.88.223::module1",
    delete="running",
    exclude = { ".*", ".tmp" },
    delay   = 30,
    init    = false,
    rsync   = {
        binary = "/usr/bin/rsync",
        archive = true,
        compress = true,
        verbose  = true,
        password_file = "/etc/rsyncd.d/rsync.pwd",
        _extra   = {"--bwlimit=200"}
    }
}

-- IV. 远程目录同步, rsync模式 + ssh shell
sync {
    default.rsync,
    source   = "/tmp/src",
    target   = "172.29.88.223:/tmp/dest",
    -- target  = "root@172.29.88.223:/remote/dest",
    -- 上面target, 注意如果是普通用户, 必须拥有写权限
    maxDelays = 5,
    delay   = 30,
    -- init = true,
    rsync   = {
        binary = "/usr/bin/rsync",
        archive = true,
        compress = true,
        bwlimit  = 2000
        -- rsh = "/usr/bin/ssh -p 22 -o StrictHostKeyChecking=no"
        -- 如果要指定其它端口, 请用上面的rsh
    }
}

-- V. 远程目录同步, rsync模式 + rsyncssh, 效果与上面相同
sync {
    default.rsyncssh,
    source   = "/tmp/src2",
    host     = "172.29.88.223",
    targetdir = "/remote/dir",
    excludeFrom = "/etc/rsyncd.d/rsync_exclude.lst",
    -- maxDelays = 5,
}

```

```
delay = 0,
-- init = false,
rsync  = {
    binary = "/usr/bin/rsync",
    archive = true,
    compress = true,
    verbose  = true,
    _extra = {"--bwlimit=2000"},
},
ssh   = {
    port  = 1234
}
}
```