

# rsync

远程数据同步工具

## 补充说明

**rsync**命令是一个远程数据同步工具，可通过LAN/WAN快速同步多台主机间的文件。rsync使用所谓的“rsync算法”来使本地和远程两个主机之间的文件达到同步，这个算法只传送两个文件的不同部分，而不是每次都整份传送，因此速度相当快。rsync是一个功能非常强大的工具，其命令也有很多功能特色选项，我们下面就对它的选项一一进行分析说明。

## 语法

```
rsync [OPTION]... SRC DEST
rsync [OPTION]... SRC [USER@]host:DEST
rsync [OPTION]... [USER@]HOST:SRC DEST
rsync [OPTION]... [USER@]HOST::SRC DEST
rsync [OPTION]... SRC [USER@]HOST::DEST
rsync [OPTION]... rsync://[USER@]HOST[:PORT]/SRC [DEST]
```

对应于以上六种命令格式，rsync有六种不同的工作模式：

1. 拷贝本地文件。当SRC和DES路径信息都不包含有单个冒号":"分隔符时就启动这种工作模式。  
如：`rsync -a /data /backup`
2. 使用一个远程shell程序(如rsh、ssh)来实现将本地机器的内容拷贝到远程机器。当DST路径地址包含单个冒号":"分隔符时启动该模式。如：`rsync -avz *.c foo:src`
3. 使用一个远程shell程序(如rsh、ssh)来实现将远程机器的内容拷贝到本地机器。当SRC地址路径包含单个冒号":"分隔符时启动该模式。如：`rsync -avz foo:src/bar /data`
4. 从远程rsync服务器中拷贝文件到本地机。当SRC路径信息包含":"分隔符时启动该模式。如：`rsync -av root@192.168.78.192::www /databack`
5. 从本地机器拷贝文件到远程rsync服务器中。当DST路径信息包含":"分隔符时启动该模式。  
如：`rsync -av /databack root@192.168.78.192::www`
6. 列远程机的文件列表。这类似于rsync传输，不过只要在命令中省略掉本地机信息即可。如：`rsync -v rsync://192.168.78.192/www`

## 选项

```
-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 保留软链接。
-L, --copy-links 想对待常规文件一样处理软链接。
```

```
--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或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KB/s
-h, --help 显示帮助信息。
```

## 实例

### SSH方式

首先在服务端启动ssh服务：

```
service sshd start
启动 sshd [确定]
```

### 使用rsync进行同步

接下来就可以在客户端使用rsync命令来备份服务端上的数据了。SSH方式是通过系统用户来进行备份的，如下：

```
rsync -vzrtpg --progress -e ssh --delete work@172.16.78.192:/www/*
/databack/experiment/rsync
work@172.16.78.192's password:
receiving file list ...
5 files to consider
test/
a
0 100% 0.00kB/s 527:35:41 (1, 20.0% of 5)
b
67 100% 65.43kB/s 0:00:00 (2, 40.0% of 5)
c
0 100% 0.00kB/s 527:35:41 (3, 60.0% of 5)
dd
100663296 100% 42.22MB/s 0:00:02 (4, 80.0% of 5)
sent 96 bytes received 98190 bytes 11563.06 bytes/sec
total size is 100663363 speedup is 1024.19
```

上面的信息描述了整个的备份过程，以及总共备份数据的大小。

### 后台服务方式

启动rsync服务，编辑/etc/xinetd.d/rsync文件，将其中的disable=yes改为disable=no，并重启xinetd服务，如下：

```
vi /etc/xinetd.d/rsync

#default: off
# description: The rsync server is a good addition to an ftp server, as it \
# allows crc checksumming etc.
service rsync {
  disable = no
  socket_type = stream
  wait = no
  user = root
  server = /usr/bin/rsync
  server_args = --daemon
  log_on_failure += USERID
```

```
}
```

```
/etc/init.d/xinetd restart  
停止 xinetd [确定]  
启动 xinetd [确定]
```

创建配置文件，默认安装好rsync程序后，并不会自动创建rsync的主配置文件，需要手工来创建，其主配置文件为“/etc/rsyncd.conf”创建该文件并插入如下内容：

```
vi /etc/rsyncd.conf  
  
uid=root  
gid=root  
max connections=4  
log file=/var/log/rsyncd.log  
pid file=/var/run/rsyncd.pid  
lock file=/var/run/rsyncd.lock  
secrets file=/etc/rsyncd.passwd  
hosts deny=172.16.78.0/22  
  
[www]  
comment= backup web  
path=/www  
read only = no  
exclude=test  
auth users=work
```

创建密码文件，采用这种方式不能使用系统用户对客户端进行认证，所以需要创建一个密码文件，其格式为“username:password”用户名可以和密码可以随便定义，最好不要和系统帐户一致，同时要把创建的密码文件权限设置为600，这在前面的模块参数做了详细介绍。

```
echo "work:abc123" > /etc/rsyncd.passwd  
chmod 600 /etc/rsyncd.passwd
```

备份，完成以上工作，现在就可以对数据进行备份了，如下：

```
rsync -avz --progress --delete work@172.16.78.192::www  
/databack/experiment/rsync  
  
Password:  
receiving file list ...  
6 files to consider  
./ files...  
a  
0 100% 0.00kB/s 528:20:41 (1, 50.0% of 6)  
b  
67 100% 65.43kB/s 0:00:00 (2, 66.7% of 6)  
c  
0 100% 0.00kB/s 528:20:41 (3, 83.3% of 6)  
dd  
100663296 100% 37.49MB/s 0:00:02 (4, 100.0% of 6)
```

```
sent 172 bytes received 98276 bytes 17899.64 bytes/sec
total size is 150995011 speedup is 1533.75
```

恢复，当服务器的数据出现问题时，那么这时就需要通过客户端的数据对服务端进行恢复，但前提是服务端允许客户端有写入权限，否则也不能在客户端直接对服务端进行恢复，使用rsync对数据进行恢复的方法如下：

```
rsync -avz --progress /databack/experiment/rsync/ work@172.16.78.192::www

Password:
building file list ...
6 files to consider
./
a
b
67 100% 0.00kB/s 0:00:00 (2, 66.7% of 6)
c
sent 258 bytes received 76 bytes 95.43 bytes/sec
total size is 150995011 speedup is 452080.87
```

### 将源目录同步到目标目录

```
$ rsync -r source destination
```

上面命令中，`-r`表示递归，即包含子目录。注意，`-r`是必须的，否则rsync运行不会成功。source目录表示源目录，destination表示目标目录。

### 多个文件或目录同步

```
$ rsync -r source1 source2 destination
```

上面命令中，source1和source2都会被同步到destination目录。

### 同步元信息

`-a`参数可以替代`-r`，除了可以递归同步以外，还可以同步元信息（比如修改时间、权限等）。由于rsync默认使用文件大小和修改时间决定文件是否需要更新，所以`-a`比`-r`更有用。下面的用法才是常见的写法。

```
$ rsync -a source destination
```

目标目录destination如果不存在，rsync会自动创建。执行上面的命令后，源目录source被完整地复制到了目标目录destination下面，即形成了destination/source的目录结构。

如果只想同步源目录source里面的内容到目标目录destination，则需要在源目录后面加上斜杠。

```
$ rsync -a source/ destination
```

上面命令执行后，source目录里面的内容，就都被复制到了destination目录里面，并不会在destination下面创建一个source子目录。

### 模拟执行的结果

如果不确定 rsync 执行后会产生什么结果，可以先用 `-n` 或 `--dry-run` 参数模拟执行的结果。

```
$ rsync -anv source/ destination
```

上面命令中，`-n` 参数模拟命令执行的结果，并不真的执行命令。`-v` 参数则是将结果输出到终端，这样就可以看到哪些内容会被同步。

### 目标目录成为源目录的镜像副本

默认情况下，rsync 只确保源目录的所有内容（明确排除的文件除外）都复制到目标目录。它不会使两个目录保持相同，并且不会删除文件。如果要使得目标目录成为源目录的镜像副本，则必须使用 `--delete` 参数，这将删除只存在于目标目录、不存在于源目录的文件。

```
$ rsync -av --delete source/ destination
```

上面命令中，`--delete` 参数会使得 destination 成为 source 的一个镜像。

### 排除文件

有时，我们希望同步时排除某些文件或目录，这时可以用 `--exclude` 参数指定排除模式。

```
$ rsync -av --exclude='*.txt' source/ destination  
# 或者  
$ rsync -av --exclude '*.txt' source/ destination
```

上面命令排除了所有 TXT 文件。

注意，rsync 会同步以“点”开头的隐藏文件，如果要排除隐藏文件，可以这样写 `--exclude=".*"`

如果要排除某个目录里面的所有文件，但不希望排除目录本身，可以写成下面这样。

```
$ rsync -av --exclude 'dir1/*' source/ destination
```

多个排除模式，可以用多个 `--exclude` 参数。

```
$ rsync -av --exclude 'file1.txt' --exclude 'dir1/*' source/ destination
```

多个排除模式也可以利用 Bash 的大括号的扩展功能，只用一个 `--exclude` 参数。

```
$ rsync -av --exclude={'file1.txt','dir1/*'} source/ destination
```

如果排除模式很多，可以将它们写入一个文件，每个模式一行，然后用 `--exclude-from` 参数指定这个文件。

```
$ rsync -av --exclude-from='exclude-file.txt' source/ destination
```

### 指定必须同步的文件模式

`--include` 参数用来指定必须同步的文件模式，往往与 `--exclude` 结合使用。

```
$ rsync -av --include="*.txt" --exclude='*' source/ destination
```

上面命令指定同步时，排除所有文件，但是会包括 TXT 文件。

获取服务器上文件列表

```
$ rsync -avz [USER@][HOST]::SRC
```

从服务器下载到本地

```
$ rsync -avz [USER@][HOST]::SRC/file.log ./
```

From:

<https://rd.irust.top/> - 学习笔记

Permanent link:

<https://rd.irust.top/doku.php?id=command:rsync>

Last update: **2021/10/15 14:58**

