

Rsync文件同步

Rsync文件同步

1. 本章结构



2. 关于rsync

1、一款增量备份工具，remote sync,远程同步，支持本地复制或者与其他SSH、rsync主机同步，官方网站:http://rsync.samba.org/。

Rsync（remote synchronize）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，也可以使用rsync同步本地硬盘中的不同目录。

Rsync是用户取代rcp的一个工具，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，可以参考How Rsync A Practical Overview进一步了解Rsync的运作机制。

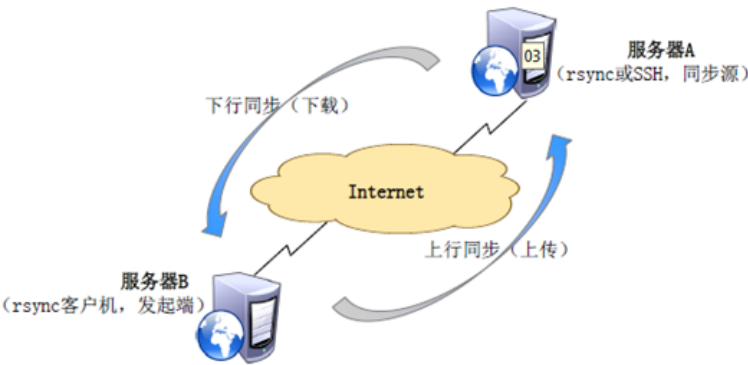
Rsync支持大多数的类Unix系统，无论是linux、solaris还是BSD上都经过了良好的测试，此外，它在windows平台下也有相应的版本，比较知名的有cwRsync和Sync2NAS。

2、特点

- 能更新整个目录树和文件系统
- 有选择性的保持符号链接、硬链接、文件属于、权限、设备以及时间等
- 对于安装来说，无任何特殊权限要求
- 对于多个文件来说，内部流水线减少文件等待的延时
- 能用rsh、ssh或直接端口作为传输输入端口
- 支持匿名rsync同步文件，是理想的镜像工具

3、同步源和发起源

Rsync同步源：指备份操作的远程服务器，也称为备份源，主要包括两种：rsync源、ssh源



4、文件格式

公告

昵称：[【风语】](#)
园龄：[3年4个月](#)
粉丝：[3](#)
关注：[0](#)
[+加关注](#)

< 2019年2月 >						
日	一	二	三	四	五	六
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	1	2
3	4	5	6	7	8	9

搜索

找找看

谷歌搜索

常用链接

- [我的随笔](#)
- [我的评论](#)
- [我的参与](#)
- [最新评论](#)
- [我的标签](#)

随笔档案

- [2015年11月 \(22\)](#)
- [2015年10月 \(8\)](#)

文章分类

服务

阅读排行榜

- [1. MySQL多主一从同步\(3064\)](#)
- [2. Idap主从同步\(2549\)](#)
- [3. DNS服务详解\(2099\)](#)
- [4. DHCP服务详解\(1917\)](#)
- [5. BIND+MySQL\(1758\)](#)

推荐排行榜

- [1. BIND+MySQL\(1\)](#)
- [2. Pytho知识总结2\(1\)](#)
- [3. Python知识总结\(1\)](#)
- [4. MySQL基本操作\(1\)](#)
- [5. MySQL多主一从同步\(1\)](#)

备份操作类型：

本地同步：rsync ... 本地目录1 本地目录2

rsync+ssh同步：

rsync ... ssh源 本地目录 （下行同步即下载）

rsync ... 本地目录 ssh源 （上行同步即上传）

rsync+rsync同步：

rsync ... rsync源 本地目录 （下行同步即下载）

rsync ... 本地目录 rsync源 （上行同步即上传）

5、rsync命令的用法：

基本格式：rsync [选项] 原始位置 目录位置

常用选项：

-a：归档模式，递归并保留对象属性，等同于-rlptgoD

-v:显示同步过程的详细(verbose)信息

-z：在传输文件时进行压缩(compress)

-H：保留硬链接文件

-A：保留ACL属性信息

--delete:删除目标位置有而原始位置没有的文件

-r：递归模式，包含目录及子目录中所有文件

-l：对于符号链接文件仍然复制为符号链接文件

-p：保留文件的权限标记

-t：保留文件的时间标记

-g：保留文件的属组标记（仅超级用户使用）

-o：保留文件的属主标记（仅超级用户使用）

-D：保留设备文件及其他特殊文件

6、同步的优缺点

①定期同步的不足

执行备份的时间固定

当同步源长期不变化时，密集的定期任务是不必要的

②实时同步的优点

一旦出现源出现变化，立即启动备份

只要同步源无变化，则不执行备份

7、linux内核的inotify机制从版本2.6.13开始提供，可以监控文件系统的变动情况，并作出通知响应，辅助软件:inotify-tools



8、调整内核的参数

max_queue_events:监控队列大小

max_user_instances:最多监控实例数

max_user_watches:每个实例最多监控文件数

9、安装inotify-tools辅助工具

inotifywait:用于持续监控，实时输出结果

inotifywatch: 用于短期监控，任务完成后再出结果

```
#inotifywait -mrq -e modify,create,move,delete /var/www/html
```

选项: -m (monitoring) 实时监控

-r (recursive) 递归

-q (quiet) 安静模式

三、案例1：rsync+ssh同步

试验准备：两台主机，A主机和B主机IP地址分别为192.168.131.130、192.168.131.136，共享目录/usr/local/nginx/html

A主机操作：

```
# useradd wdd
```

```
# passwd wdd
```

```
# setfacl -m u:wdd:rwX /usr/local/nginx/html/ #设置ACL权限
```

```
# setfacl -m d:wdd:rwX /usr/local/nginx/html/
```

主机B操作：

```
# mkdir /client
```

```
# rsync -avz wdd@192.168.131.130:/usr/local/nginx/html/* /client
```

上面操作只同步/usr/local/nginx/html/下的文件，如果写成/usr/local/nginx/html，则同步整个目录

1. 案例2：rsync+rsync

试验准备：A主机IP地址为192.168.131.130，B主机IP地址为192.168.131.136

A主机作为同步源：

```
# mkdir /server
```

```
# cd /server/
```

```
# chmod 777 /server/ #注意两边都要有相应的权限
```

```
# touch server.txt
```

```
# vim /etc/rsyncd.conf
```

```
use chroot = yes

address = 192.168.131.130

port 873

log file = /var/log/rsyncd.log

pid file = /var/run/rsyncd.pid

[share]

comment = rsync server

path = /server

read only = no

dont compress = *.gz #对后面格式不进行压缩

auth users = wdd

secrets file = /etc/rsyncd_users.db

# vim /etc/rsyncd_users.db

wdd:123456 #用户名和密码

# chmod 600 /etc/rsyncd_users.db

#注意一定要改这个权限，否则认证不通过

# killall -9 rsync

# rsync --daemon #只能这样启动rsync
```

主机B客户端：

进行下载操作

```
# rsync -avz wdd@192.168.131.130::share /client
```

也可以写成下面这样：

```
# rsync -avz rsync://wdd@192.168.131.130/share/* /client/ /
```

两种格式：

```
rsync -avz xxx@服务器地址::共享名 本地目录
```

```
rsync -avz rsync://xxx@服务器地址/共享名 本地目录
```

进行上传操作：

```
# cd /client/
```

```
# touch client.txt
```

```
# rsync -avz /client/client.txt rsync://wdd@192.168.131.130/share/
```

1. 案例3：rsync+inotify实现文件实时同步

试验准备：主机A的IP地址为192.168.131.130，主机B的IP地址为192.168.131.136,主机A作为同步源

1、主机A操作：

同案例2操作

2、主机B操作：

```
# tar xf inotify-tools-3.14.tar.gz
```

```
# cd inotify-tools-3.14
```

```
# ./configure
```

```
# make
```

```
# make install
```

安装工具后需更改内核参数：vi /etc/sysctl.conf

```
fs.inotify.max_queue_events=16384 监控队列大小
fs.inotify.max_user_instances=1024 最多监控实例数
fs.inotify.max_user_watches=1048576 每个实例最多监控文件数
```

```
# /usr/local/bin/inotifywait -mrq -e delete,create,modify,move /client/ #对/client目录进行实时监控
```

可以将上面的命令写成一个shell脚本，只要/client文件夹下面的文件发生修改，立即用rsync同步文件,脚本如下所示：

```
# vim inotify_check.sh
```

```
#!/bin/bash
```

```
SRCCMD="/usr/local/bin/inotifywait -mrq -e delete,create,modify,move /client/"
```

```
DESTCMD="rsync -avz /client/* rsync://wdd@192.168.131.130/share/"
```

```
${SRCCMD}while read DIRECTORY EVENTS FILE
```

```
do
```

```
${DESTCMD}
```

```
Done
```

```
# chmod +x inotify_check.sh
```

```
# export RSYNC_PASSWORD=123456
```

#将上面变量RSYNC_PASSWORD声明为全局变量，这样就不用每次运行上面脚本输入密码了

```
# ./inotify_check.sh #运行脚本
```

打开另外一个窗口，作如下操作：

```
# cd /client/
```

```
# touch 1.txt
```

只要主机B下的/client文件夹下面的文件发生修改、创建、删除、移动都会被检测到，并被同步到主机A下的/server下

```
[root@myrhel4 ~]# ./inotify_check.sh
sending incremental file list
1.txt

sent 81 bytes  received 27 bytes  216.00 bytes/sec
total size is 0  speedup is 0.00
```

1. 案例4：rsync+inotify+unison实现双机互相同步

实验准备：两台主机，A主机地址为192.168.131.130,主机B的IP地址为192.168.131.136

1. 操作主机A

1. 配置SSH

```
# ssh-keygen -t rsa
```

```
# ssh-copy-id 192.168.131.136
```

```
# ssh root@192.168.131.136#测试是否登录主机B不需要输入密码
```

2. 安装inotify

```
# tar xf inotify-tools-3.14.tar.gz
```

```
# cd inotify-tools-3.14
```

```
# ./configure
```

```
# make && make install
```

3. 安装unison

```
# tar xf ocaml-3.10.1.tar.gz #先安装unison的底层依赖包
```

```
# cd ocaml-3.10.1
```

```
# ./configure
```

```
# make world opt
```

```
# make install
```

```
# tar xf unison-2.13.16.tar.gz
```

```
# cd unison-2.13.16
```

```
# make UISTYLE=text STATIC=true THREADS=true
```

```
# make install
```

```
# cp unison /usr/local/bin/
```

4. 创建同步目录

```
# mkdir /backup1
```

5. # vim intify_unison.sh

```
#!/bin/bash
```

```
IP="192.168.131.136"
```

```
SRC="/backup1"
```

```
DEST="/backup2"
```

```
/usr/local/bin/inotifywait -mrq -e modify,create,delete,move ${SRC} while \
```

```
read DIRECTORY EVENTS FILE
```

```
do
```

```
/usr/local/bin/unison -batch ${SRC} ssh://${IP}/${DEST}
```

```
done
```

```
# chmod +x intify_unison.sh
```

1. 操作主机B

1. 配置SSH

```
# ssh-keygen -t rsa
```

```
# ssh-copy-id 192.168.131.130
```

```
# ssh 192.168.131.130 #测试登录主机A是否需要登录密码
```

2. 安装inotify

```
# tar xf inotify-tools-3.14.tar.gz
```

```
# cd inotify-tools-3.14
```

```
# ./configure
```

```
# make && make install
```

3. 安装unison

```
# tar xf ocaml-3.10.1.tar.gz #安装unison需要的底层依赖包
```

```
# cd ocaml-3.10.1
```

```
# ./configure

# make world opt

# make install

# make UISTYLE=text THREADS=true STATIC=true

# make install

# cp unison /usr/local/bin/
```

4. 创建同步目录

```
# mkdir /backup2
```

5. # vim intify_unison.sh

```
#!/bin/bash

IP="192.168.131.130"

SRC="/backup2"

DEST="/backup1"

/usr/local/bin/inotifywait -mrq -e modify,create,delete,move ${SRC} | while \

read DIRECTORY EVENTS FILE

do

/usr/local/bin/unison -batch ${SRC} ssh://${IP}/${DEST}

done
```

```
# chmod +x intify_unison.sh
```

1. 测试两主机是否互相同步数据

1. 在主机A上操作:

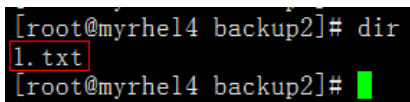
```
# ./intify_unison.sh
```

打开另一个对话窗口，作如下操作:

```
# cd /backup1/
```

```
# touch 1.txt
```

打开主机B发现了1.txt



```
[root@myrhel4 backup2]# dir
1.txt
[root@myrhel4 backup2]#
```

2. 在主机B上操作:

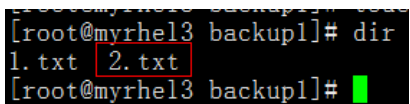
```
# ./intify_unison.sh
```

打开另一个对话窗口，作如下操作:

```
# cd /backup2/
```

```
# touch 2.txt
```

打开主机A发现了2.txt



```
[root@myrhel3 backup1]# dir
1.txt 2.txt
[root@myrhel3 backup1]#
```

好文置顶

关注我

收藏该文







【风语】

关注 - 0

粉丝 - 3

0

0

+加关注

« 上一篇: [Lnmp下安装memcached](#)
» 下一篇: [Linux双网卡绑定和解除绑定的实现](#)

posted @ 2015-11-04 21:16 [【风语】](#) 阅读(727) 评论(0) [编辑](#) [收藏](#)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

【推荐】超50万VC++源码: 大型组态工控、电力仿真CAD与GIS源码库！
【推荐】专业便捷的企业级代码托管服务 - Gitee 码云

- 相关博文:
- [RSync实现文件备份同步](#)
 - [rsync实现文件同步](#)
 - [Rsync 文件同步工具](#)
 - [rsync ssh文件同步](#)
 - [rsync 文件.数据同步](#)

- 最新新闻:
- [世界卫生组织公布预防听力损伤新标准，对智能手机提出新要求](#)
 - [字节跳动的支付业务终上正轨，但“逐梦金融圈”谈何容易](#)
 - [为什么说你应该停更“双微一抖”](#)
 - [“墨子号”科研团队获美国2018年度克利夫兰奖](#)
 - [苹果失去“美国人最亲密品牌”称号 迪斯尼取而代之](#)
- » [更多新闻...](#)