

SERSYNC实现数据实时同步

惨绿少年 Linux运维, 玩转Linux, 运维基本功 0评论 来源: 本站原创 30°C 字体: 小 中 大

1.1 第一个里程碑：安装sersync软件

1.1.1 将软件上传到服务器当中并解压

1、上传软件到服务器上 rz -E

为了便于管理上传位置统一设置为 /server/tools 中

2、解压软件包

```
[root@backup sersync_installdir_64bit]# tree
.
├── sersync
│   ├── bin
│   │   └── sersync
│   ├── conf
│   │   └── confxml.xml
│   └── logs
```

1.1.2 二进制包安装方法

二进制包安装软件方法（绿色软件安装方法）：

直接解压就可以使用

```
[root@nfs01 sersync_installdir_64bit]# mv sersync/ /usr/local/
[root@nfs01 tools]# tree /usr/local/sersync/
/usr/local/sersync/
├── bin
│   └── sersync
├── conf
│   └── confxml.xml
└── logs

3 directories, 2 files
```

1.2 第二个里程碑：编写sersync配置文件

1.2.1 常见的语法格式

rsync 配置文件编写：ini语法

sersync配置文件编写：xml语法

ansible配置文件编写:yml 语法

1.2.2 修改配置文件

编写前备份

```
[root@backup conf]# ll
total 4
-rw-r--r-- 1 root root 2214 Oct 26 2011 confxml.xml
[root@backup conf]# cp confxml.xml{,.bak}
```

6-11行表示排除同步的数据，等价于 `--exclude` 功能，表示排除

```
<filter start="false">
  <exclude expression="(.)\.svn">exclude</exclude>
  <exclude expression="(.)\.gz">exclude</exclude>
  <exclude expression="^info/*">exclude</exclude>
  <exclude expression="^static/*">exclude</exclude>
</filter>
```

12-21行是利用inotify的功能监控指定的事件，等价与 `-e create, delete.....` 表示指定监控事件信息

```
<inotify>
  <delete start="true"/>
  <createFolder start="true"/>
  <createFile start="false"/>
  <closeWrite start="true"/>
  <moveFrom start="true"/>
  <moveTo start="true"/>
  <attrib start="false"/>
  <modify start="false"/>
</inotify>
```

24-28行：推送到哪里 `name=模块` 是rsync服务器的地址

```
<localpath watch="/data"> #监控那个目录
  <remote ip="172.16.1.41" name="backup"/>

</localpath>
```

29-35行 定义rsync推送时的参数信息。

注意：不要有单词拼写错误（true），否则程序不能正常启动，卡死

```
<rsync>
  <commonParams params="-az"/>
  <auth start="true" users="rsync_backup" passwordfile="/etc/rsync.password"/>
  <userDefinedPort start="false" port="874"/>
  <timeout start="false" time="100"/>
  <ssh start="false"/>
</rsync>
```

配置文件最终内容：



```
1 [root@nfs01 tools]# cat /usr/local/sersync/conf/confxml.xml
2  xml version="1.0" encoding="ISO-8859-1"?>
3  <head version="2.5">
4      <host hostip="localhost" port="8008">host</host>
5      <debug start="false"/>
6      <fileSystem xfs="false"/>
7      <filter start="false">
8          <exclude expression="(.*).svn">exclude</exclude>
9          <exclude expression="(.*).gz">exclude</exclude>
10         <exclude expression="^info/*">exclude</exclude>
11         <exclude expression="^static/*">exclude</exclude>
12     </filter>
13     <inotify>
14         <delete start="true"/>
15         <createFolder start="true"/>
16         <createFile start="false"/>
17         <closeWrite start="true"/>
18         <moveFrom start="true"/>
19         <moveTo start="true"/>
20         <attrib start="false"/>
21         <modify start="false"/>
22     </inotify>
23
24     <sersync>
25         <localpath watch="/data">
26             <remote ip="172.16.1.41" name="nfsbackup"/>
27
28         </localpath>
29     </sersync>
30     <rsync>
31         <commonParams params="-az"/>
32         <auth start="true" users="rsync_backup" passwordfile="/etc/rsync.password"/>
33         <userDefinedPort start="false" port="874"/>
34         <timeout start="false" time="100"/>
35         <ssh start="false"/>
36     </rsync>
37     <failLog path="/tmp/rsync_fail_log.sh" timeToExecute="60"/>
38     <crontab start="false" schedule="600">
39         <crontabfilter start="false">
40             <exclude expression="*.php">exclude</exclude>
41             <exclude expression="info/*">exclude</exclude>
42         </crontabfilter>
43     </crontab>
44     <plugin start="false" name="command"/>
45 </sersync>
46
47 <plugin name="command">
48     <param prefix="/bin/sh" suffix="" ignoreError="true"/>
49     <filter start="false">
50         <include expression="(.*).php"/>
51         <include expression="(.*).sh"/>
52     </filter>
53 </plugin>
54
55 <plugin name="socket">
56     <localpath watch="/opt/tongbu">
57         <deshost ip="192.168.138.20" port="8009"/>
```

```
58     localpath>
59     plugin>
60     <plugin name="refreshCDN">
61     <localpath watch="/data0/htdocs/cms.xoyo.com/site/">
62         <cdninfo domainname="ccms.chinacache.com" port="80" username="xxxx" passwd="xxxx"/>
63         <sendurl base="http://pic.xoyo.com/cms"/>
64         <regexurl regex="false" match="cms.xoyo.com/site(/[a-zA-Z0-9]*).xoyo.com/images"/>
65     localpath>
66     plugin>
67 head>
```

View Code 配置文件最终内容

1.3 第三里程碑：启动sersync

1.3.1 修改文件的权限（可执行）

首先让程序让文件有执行权限

```
[root@nfs01 bin]# chmod a+x sersync
[root@nfs01 bin]# ll
total 1768
-rwxr-xr-x 1 root root 1810128 Oct 26 2011 sersync
```

1.3.2 查看软件的帮助信息

```
[root@nfs01 bin]# ./sersync -h
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
```

重要参数-d:启用守护进程模式

重要参数-r:在监控前，将监控目录与远程主机用rsync命令推送一遍

参数-n: 指定开启守护线程的数量，默认为10个

重要参数-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

1.3.3 在程序的bin目录下启动程序

```
./sersync -dro /usr/local/sersync/conf/confxml.xml
```

1.3.4 启动方法二

将/usr/local/sersync/bin/程序的bin目录添加到PATH中

```
export PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin:/usr/local/se
```

然后sersync命令就能直接使用

```
[root@nfs01 scripts]# sersync -dro /usr/local/sersync/conf/confxml.xml
set the system param
execute: echo 5000000 > /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 remote servers before the sersync work
option: -o      config xml name: /usr/local/sersync/conf/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: 22 = 12(Thread pool nums) + 10(Sub threads)
please according your cpu , use -n param to adjust the cpu rate
-----
rsync the directory recursively to the remote servers once
working please wait...
execute command: cd /data && rsync -az -R --delete ./ rsync_backup@172.16.1.41::nfsbackup --pass
run the sersync:
watch path is: /data
```

1.4 Inotify与 sersync总结对比

1.4.1 Inotify实时并发：

结论：经过测试，每秒200文件并发，数据同步几乎无延迟（小于1秒）

1.4.2 inotify 优点：

1) 监控文件系统事件变化，通过同步工具实现实时数据同步。

1.4.3 inotify 缺点

- 1) 并发如果大于200个文件（10-100k），同步就会有延迟
- 2) 我们前面写的脚本，每次都是全部推送一次，但确实是增量的。也可以只同步变化的文件，不变化的不理。
- 3) 监控到事件后，调用rsync同步是单进程的，而sersync为多进程同步。既然有了inotify-tools，为什么还要开发sersync？

1.4.4 sersync功能多：（inotify+rsync命令）

1) 支持通过配置文件管理

- 2) 真正的守护进程socket
- 3) 可以对失败文件定时重传（定时任务功能）
- 4) 第三方的HTTP接口（例如：更新cdn缓存）
- 5) 默认多进程rsync同步

1.4.5 高并发数据实时同步方案小结：

- 1) inotify（sersync）+ rsync，是文件级别的。
- 2) drbd文件系统级别，文件系统级别，基于block块同步，缺点：备节点数据不可用
- 3) 第三方软件的同步功能：mysql同步（主从复制），oracle，mongodb
- 4) 程序双写，直接写两台服务器。
- 5) 利用产品业务逻辑解决（读写分离，备份读不到，读主）

2.1 man命令的级别

centos6

```
[root@nfs01 ~]# man man
The standard sections of the manual include:

1      User Commands                #用户命令
2      System Calls                 #系统调用
3      C Library Functions          # C库函数
4      Devices and Special Files    #设备和特殊文件
5      File Formats and Conventions #文件格式和约定
6      Games et. Al.                #游戏等。
7      Miscellanea                  #杂记
8      System Administration tools and Daemons #系统管理工具和程序

Distributions customize the manual section to their specifics,
which often include additional sections.
```

centos7

```
[root@clsn tuichu]# man ~
1 Executable programs or shell commands
2 System calls (functions provided by the kernel)
3 Library calls (functions within program libraries)
4 Special files (usually found in /dev)
5 File formats and conventions eg /etc/passwd
```

- 6 Games
- 7 Miscellaneous (including macro packages and conventions), e.g. `man(7)`, `groff(7)`
- 8 System administration commands (usually only `for` root)
- 9 Kernel routines [Non standard]

赞0

如无特殊说明，文章均为本站原创，转载请注明出处

- 转载请注明来源：sersync实现数据实时同步
- 本文永久链接地址：<https://www.nmtui.com/clsn/lx906.html>

该文章由 惨绿少年 发布



惨绿少年Linux www.nmtui.com