inotify+rsync、sersync 实时备份

原创 GeorgeKai 2018-01-27 15:43:46 评论(1) 808人阅读

inotify+rsync、**sersync** 实时备份

1.1 定时备份缺点：

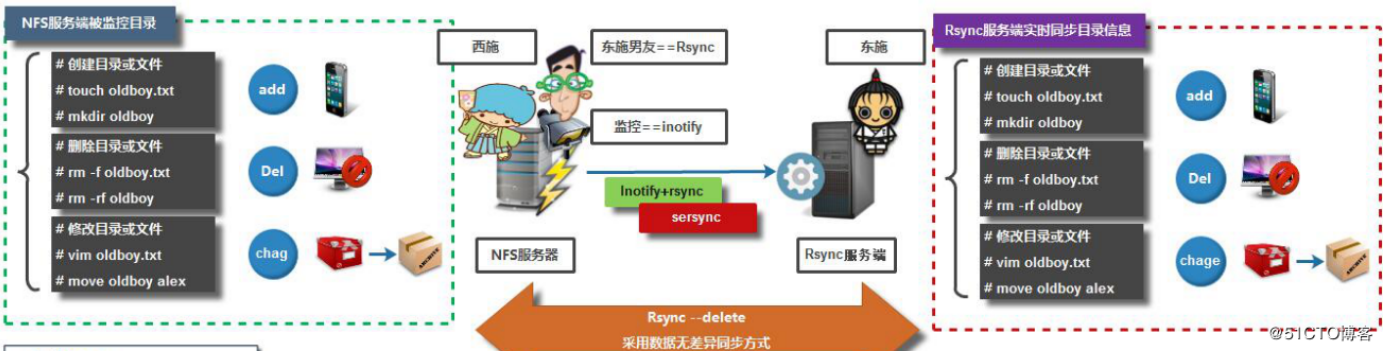
1. 浪费系统性能（数据没变化到时间也会备份）
2. 数据安全性不高（定时任务最短1分钟同步一次，如果1分钟内数据变化后，服务器宕机了，就 会丢失数据）

1.2 inotify+rsync

1.2.1 inotify是什么？

1. 是一种强大的，细粒度的，异步的文件系统事件监控系列
2. linux内核从2.6.13起，加入了inotify的支持
3. 作用：通过inotify可以监控文件系统中的添 加，删除，修改，移动等各种事件

1.2.2 实时同步原理



1）划分存储与备份服务器
2）在存储服务器上部署监控服务，监控相应文件或目录中信息的变化
3）变化后，将文件或目录数据信息进行推送，实现实时同步到rsync服务器

1.2.3 实时备份部署原理过程

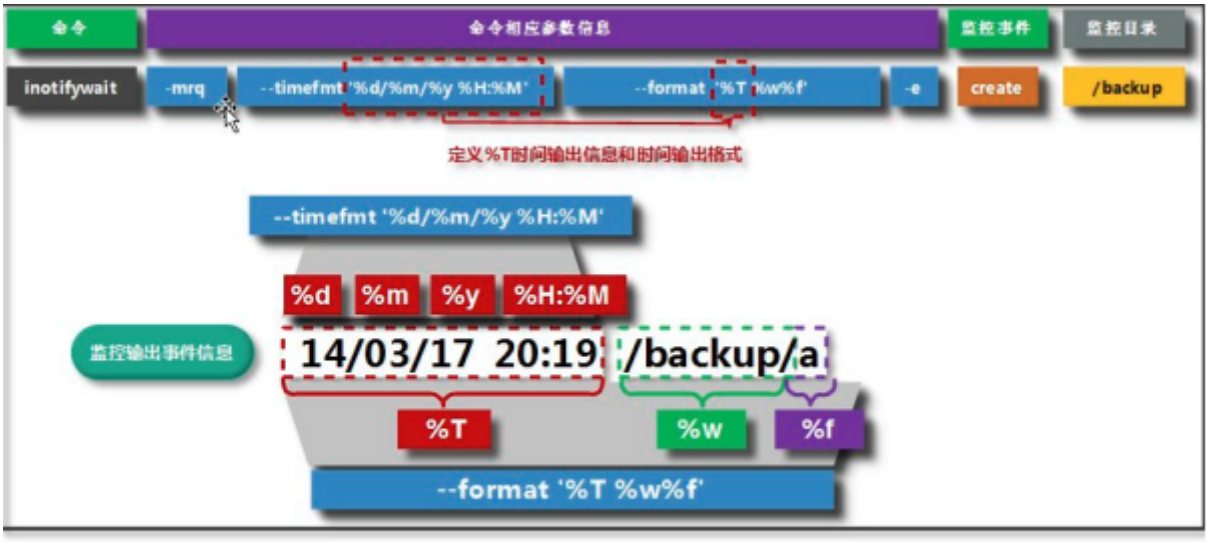
1. 部署rsync守护进程模式
2. 部署inotify实时监控数据变化的服务

1）安装inotify

```
yum install inotify-tools -y
```

inotify-tools包含2个工具：
主要使用inotifywait：在被监控的文件或目录中等待系统事件的发生
inotifywatch：统计被监控的目录或文件中，事件的发生次数

2）事件目录监控命令总结
inotifywait命令语法格式说明图：



实时同步命令参数示意图。 @51CTO博客

inotifywait -mrq --timefmt "%F" --format "%T %w%f 事件信息:%:e" -e create /data

注：下面有inotifywait命令每个参数的详解

3）利用脚本方式，将rsync与inotify服务串联在一起

vim inotify.sh

脚本内容：

```
#!/bin/bash
#author:georgekai
inotifywait -mrq  --format "%w%f" -e create,close_write,delete,moved_to /data|\
while read line
do
rsync  -az /data/ --deletersync_backup@172.16.1.41::backup--password-file=/etc/rsync.password
done
```

注：脚本在无限循环执行时：（用以下一种方法可中断执行过程）

1. 利用ctrl+z,放到后台暂停脚本运行，在killall -f inotify，在切换到前台时，已被杀死
   注： jobs #查看后台停止的进程和对应的序号
   fg + 进程序号 #将后台的命令放前台继续执行
   bg #将前台运行的命令放入后台继续执行
2. pkill -f inotify #-f：带inotify字符的进程都杀死
3. kill -9 9857 9859 #kill后面接机进程对应的PID号

shell脚本循环方式：

1. for循环 ： 指定循环条件，循环条件不满足会停止循环
2. while循环：指定循环条件，当循环条件满足时，会无限循环
3. until循环：指定循环条件，当循环条件不满足时，会无限循环

4）写好脚本，并赋予x执行权限，放入/etc/rc.local中

1.2.4 inotifywait常用参数

-m #保持实时监控，前台监控
-d #类似-m，在后台监控
-r #递归监控
-q #输出信息少，安静模式
--timefmt #指定时间输出的格式
--format #指定指定的输出类型格式（%w:监控目录）
-e #指定事件类型（如：modify、create、close_write、move、delete、attrib）

注：-e指定的事件类型，下面有详详解

1.2.5 inotify中- -format的常用参数：

%f #监控目录中哪个文件触发了这个事件
%w #监控的目录
%T #--timefmt定义的时间格式

%e #产生事件的信息，多个事件默认以逗号分割
%Xe #输出连续事件信息时，X表示以什么符号分割

1.2.6 inotify中 -e参数的常用事件类型：

close_write #文件或目录在写入模式打开之后关闭
close_nowrite #文件或目录在只读模式打开之后关闭
moved_to #将文件或目录移动（拉）到监控目录中
moved_from #将文件或目录从监控目录中移动（推）出去
create #在监控目录中创建文件或目录
delete #在监控目录中删除文件或目录

测试在监控目录中对文件的各种操作所触发的监控事件类型：
1. 创建文件逻辑过程：
touch /data/123.txt
会触发：/data/ CREATE 123.txt
/data/ OPEN 123.txt
/data/ ATTRIB 123.txt
/data/ CLOSE_WRITE,CLOSE 123.txt

2. 删除文件逻辑过程：
\rm 999.txt
会触发：/data/ DELETE 999.txt

3. 修改文件逻辑过程：
echo 123 >> 123.txt
会触发：/data/ OPEN 123.txt
/data/ MODIFY 123.txt
/data/ CLOSE_WRITE,CLOSE 123.txt

4. 移动文件逻辑过程：
1）将hosts文件移动到data目录
mv /etc/hosts /data
触发：/data/ MOVED_TO hosts
2）将data目录中的hosts文件，移动到/etc下
mv /data/hosts /etc/
触发：/data MOVED_FROM hosts

5. 修改文件属性逻辑过程：
mv kai.txt wang.txt ——重命名
触发：/data/ MOVED_FROM kai.tx
/data/ MOVED_TO wang.txt

1.2.7 inotify服务优化

1. 在/proc/sys/fs/inotify/目录下有三个文件，对inotify机制有一定的限制：
   1）max_user_watches #设置inotifywait或inotifywatch命令可以监控的文件数量（单进程）
   注：默认监控文件数8192个
   2）max_user_instances #设置每个用户可以运行的inotifywait或inotifywatch命令的进程数
   注： 1. 一个服务识别不同的配置文件，启动多个进程，就会开启多个不同的socket条目 生成多个不同的端 口信息，以上操作就实现了一个服务的多实例创建
   2. 默认监控进程数128个
   3）max_queued_events #设置inotify事件（event）队列可容纳的事件数量
   注：1. 默认队列中可容纳16384个事件
   2. 监控目录中文件变化越频繁，这个值就越大，超过设置的最大值就会被丢弃
2. 对 max_user_watches max_user_instances max_queued_events 三个文件进行优化
   建议设置为最大值：50000000，并设置开机自启（有时间重启可能会恢复）
   echo “50000000” >/proc/sys/fs/inotify/max_user_watches
   echo “50000000” >/proc/sys/fs/inotify/max_queued_events
   echo ″50000000″ >/proc/sys/fs/inotiofy/max_user_istances
   将这三条命令追加到/etc/rc.local中。

3 查看inotify man手册

man inotify #有显示inotify（7），7表示第七个级别

man man #可能查看man的七个级别的含义

man 7 inotify #即可查看inotify中max_queque_event等三个文件的使用帮助

1.2.8 inotify软件的优点和缺点

优点：监控文件系统事件发生变化，会通过rsync实时同步数据

缺点： 1. 发现大于200个文件（大小4-100k），同步时会有延时

2. 前面写的脚本，每次都是全部推送一次，但确实是增量的。也可以只同步变化文件，不变的可以不理。

3. 监控到事件后，rsync同步是单线程的（效率低），serync同步是多线程（效率高）

注：一个进程可管理多个线程

4. inotify实现实时同步需要编写shell脚本

1.3 sersync软件（国内金山开发）

1.3.1 sersync功能：

1. 支持配置文件管理
2. 真正的守护守护进程是socket
3. 可以对失败文件定时重传（定时任务的功能）
4. 第三方的HTTP接口（例如CDN缓存）
5. 默认多线程同步，效率高（inotify是单线程）

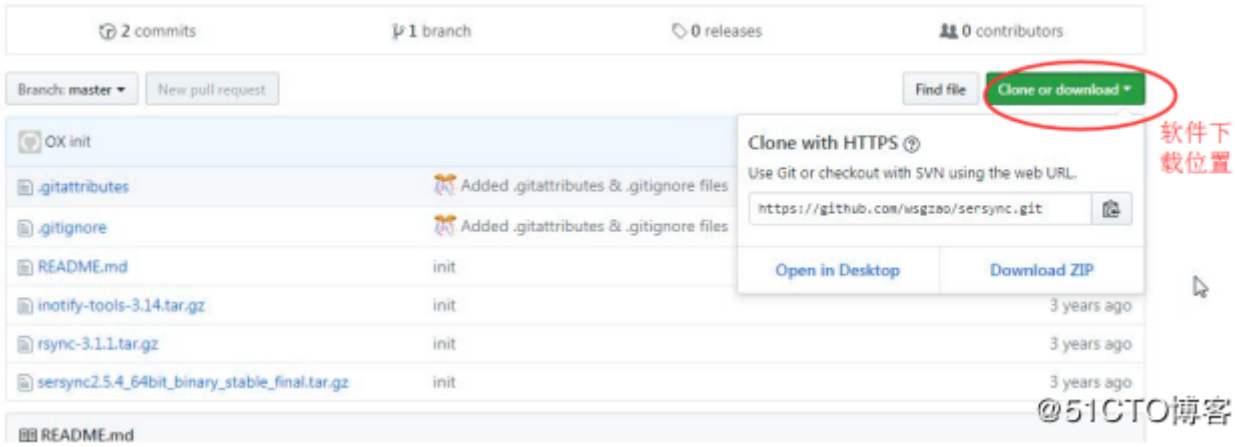
1.3.2 sersync软件服务部署流程：

1. 部署好rsync守护进程服务
2. 部署sersync实时监控服务

1）确认sersync软件是否安装

▲ sersync绿色软件包的安装方式（二进制包安装方式）

sersync软件官方地址（下载）：https://github.com/wsgzao/sersync



▲下载好后，上传到NFS服务器中

▲因为zip格式的，需要unzip 解压

▲解压后将软件包mv到合适位置/usr/local/下

2）修改sersync配置文件（修改前先备份源文件）

vim /usr/local/sersync/conf/confxml.xml

以下标签只需了解：

▲. <filter start="false"> #false表示不过滤（默认）

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

注：filter标签用来实现实时同步的排除功能

▲. <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>
```

注：指定监控文件或目录变化的事件信息

3. 启动sersync服务

1）chmod +x sersync #先授予sersync命令执行权限

2）/bin/sersync -h #查看帮助（都是中文的）

3）/bin/sersync -dro /usr/local/sersync/conf/confxml.xml

注：不要用xinetd管理rsync服务（sersync会调用inotify+rsync服务,配置文件中有）

sersync参数信息：

参数-d:启用守护进程模式

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

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

4. 查看服务是否启动

```
ps -ef |grep sersync
```

1.4 互联网常见数据同步方法总结：

1. inotify(sersync)+rsync ，是文件级别的。
2. drbd数据同步软件，基于block块同步
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

感觉最新排版好累，思路不清都没法排

小伙伴们可以关注我的微信公众号：linux运维菜鸟之旅

关注“中国电信天津网厅”公众号，首次绑定可免费领2G流量，为你的学习提供流量！