Rsync 服务器同步

linux下两台服务器文件实时同步方案设计和实现

假设有如下需求：

假设两个服务器：

192.168.0.1 源服务器  有目录 /opt/test/

192.168.0.2 目标服务器  有目录 /opt/bak/test/

实现的目的就是保持这两个服务器某个文件目录保持实时同步

实现方式： 通过rsync+inotify-tools结合来实现

需要安装软件：

1.  rsync 同步软件

   在 源服务器 和 目标服务器 都需要安装

   源服务器： 是rsync客户端，不需要配置

 目标服务器： 是rsync服务器端，需要配置/etc/rsyncd.conf里的内容

  安装后需要新建配置文件：/etc/rsyncd.conf

  配置文件在： /etc/

文件内容如下：

uid = root  
gid = root  
use chroot = no  
max connections = 10  
strict modes = yes  
pid file=/var/run/rsyncd.pid  
lock file=/var/run/rsyncd.lock  
log file= =/var/run/rsyncd.log

[www]  
path= /opt/bak/test  
comment= analyse  
read only = false  
hosts allow = \*

2.  inotify-tools 工具

该工具为文件实时监控工具，需要linux操作系统内核支持，内核支持需要至少版本为2.6.13

检查操作系统是否支持，执行如下：

uname -r  查看版本

返回：

2.6.32-220.4.1.el6.x86\_64

则表示版本2.6.32 大于2.6.13，则支持。

执行：

ll /proc/sys/fs/inotify  
total 0  
-rw-r--r-- 1 root root 0 Oct 18 12:18 max\_queued\_events  
-rw-r--r-- 1 root root 0 Oct 18 12:18 max\_user\_instances  
-rw-r--r-- 1 root root 0 Oct 18 12:18 max\_user\_watches

有三项输出,则表示默认支持inotify,可以安装inotify-tools工具.

如果不支持，需要采用新版本的linux操作系统

版本达到要求，就可以安装了。

安装inotify-tools后会在相关安装目录下生成如下两个文件：

ll /usr/local/bin/  
total 88  
-rwxr-xr-x 1 root root 44327 Oct 10 15:32 inotifywait  
-rwxr-xr-x 1 root root 41417 Oct 10 15:32 inotifywatch

则表示安装成功。

注意： 在 源服务器上需要安装，目标服务器上不需要安装inotify。

3. 相关脚本：

 在源服务器上新建脚本：

inotify\_bak.sh

#!/bin/bash  
src=/opt/test/  
/usr/local/bin/inotifywait -mrq --timefmt '%d/%m/%y %H:%M' --format '%T %w%f%e' -e close\_write,delete,create,attrib $src |  while read file  
do  
      
      /usr/bin/rsync -arzuq $src 192.168.0.2::www/

      echo "  ${file} was rsynced" >>/opt/soft/log/rsync.log 2>&1  
done

注意： 这里的 www 是在目标服务器/etc/rsyncd.conf里配置的模块名称：[www]

赋予执行权限： chmod +x  inotify\_bak.sh

然后执行： inotify\_bak.sh &  放入后台执行

4. 关于启动

   目标服务器：先启动rsync后台服务： /usr/bin/rsync --daemon

   来源服务器： 执行 inotify\_bak.sh &

5. 测试：

   在来源服务器目录中新建目录和文件，inotify\_bak.sh脚本会检测到，然后同步到目标服务器的相关目录下

   可以查看日志文件： /opt/soft/log/rsync.log 命令如下：观察实时同步的情况。

   tail -f  /opt/soft/log/rsync.log