什么是sersync？

sersync主要用于服务器同步，web镜像等功能。基于boost1.43.0,inotifyapi,rsynccommand.开发。目前使用的比较多的同步解决方案是inotify-tools+rsync，另外一个是google开源项目Openduckbill（依赖于inotify-tools），这两个都是基于脚本语言编写的。相比较上面两个项目，本项目优点是：sersync是使用c++编写，而且对linux系统文件系统产生的临时文件和重复的文件操作进行过滤，所以在结合rsync同步的时候，节省了运行时耗和网络资源。因此更快。

单纯使用rsync和使用serync+rsync的区别：

使用rsync+crontab做定时同步时，主服务器端开启rsync守护进程，而镜像服务器是运行rsync客户端，平时一般会利用crontab定时获取rsync服务器上的数据。但使用rsync+sersync做实时同步时，用于推送文件的服务器运行sersync服务，用于接收文件的服务器则运行rsync守护进程，简单来说就是sersync会利用rsync命令将文件推送到rsync服务器，实际线上使用一般会把sersync作为主服务器，rsync作为镜像服务器，实现数据同步备份，web镜像等功能。前提条件：在使用sersync之前，我们必须安装配置好rsync服务器。这里我们需要注意的是，纯粹的使用rsync做单向同步时，rsync的守护进程是运行在文件推送的服务器上，而接收的服务器是运行rsync客户端。使用sersync做文件实时同步刚好相反，用于接收文件的服务器运行rsync守护进程。

一、安装sersync：

下载地址：https://code.google.com/p/sersync/downloads/list

[root@mastersrc]#lssersync2.5.4\_64bit\_binary\_stable\_final.tar.gz

[root@mastersrc]#tar xzf sersync2.5.4\_64bit\_binary\_stable\_final.tar.gz

[root@mastersrc]#ls GNU-[Linux](http://www.it165.net/os/oslin/)-x86

[root@mastersrc]#cd GNU-[Linux](http://www.it165.net/os/oslin/)-x86/

[root@masterGNU-Linux-x86]#ll

为了方面使用，我们把它cp一下

[root@masterGNU-Linux-x86]#mkdir /usr/local/sersync

[root@masterGNU-Linux-x86]#mv \* /usr/local/sersync/

二、基本配置：

2.1接收端创建密码文件：

|  |
| --- |
| [root@masterGNU-Linux-x86]#cd/usr/local/sersync/  [root@mastersersync]#echo"9865321">rsync.pas  [root@mastersersync]#chmod 600 rsync.pas |

2.2编辑主配置文件：

以下红色部分为我修改的

[root@mastersersync]#viconfxml.xml

<?xmlversion="1.0"encoding="ISO-8859-1"?><headversion="2.5"><hosthostip="localhost"port="8008"></host><debugstart="false"/><fileSystemxfs="false"/>

<!--监控事件的过程中过滤特定文件，和特定文件夹的文件--><filterstart="true"><excludeexpression="(.\*)\.html"></exclude><excludeexpression="(.\*)\.php"></exclude>

<excludeexpression="(.\*)\.jpg"></exclude>

<excludeexpression="(.\*)\.gif"></exclude>

<excludeexpression="(.\*)\.png"></exclude>

<excludeexpression="(.\*)\.css"></exclude>

<excludeexpression="(.\*)\.js"></exclude>

<excludeexpression="^tmp/\*"></exclude></filter>

<!--设置要监控的事件--><inotify><deletestart="true"/><createFolderstart="true"/><createFilestart="true"/><closeWritestart="true"/><moveFromstart="true"/><moveTostart="true"/><attribstart="false"/><modifystart="true"/></inotify>

<sersync>

<!--设定监控目录--><localpathwatch="/var/www/html">

<!--指定远端rsync的地址和模块名-->

<remoteip="192.168.204.129"name="webhome"/><!--<remoteip="192.168.204.128"name="tongbu"/>--><!--<remoteip="192.168.8.40"name="tongbu"/>--></localpath>

<rsync><commonParamsparams="-artuz"/>

<!--是否启用验证，并指定密码存放文件--><authstart="true"users="apache"pass[word](http://www.it165.net/edu/ebg/)file="/usr/local/sersync/rsync.passwd"/><userDefinedPortstart="false"port="874"/><!--port=874--><timeoutstart="false"time="100"/><!--timeout=100--><sshstart="false"/></rsync><failLogpath="/tmp/rsync\_fail\_log.sh"timeToExecute="60"/><!--defaultevery60minsexecuteonce—>

<!--是否启用执行完整rsync，并指定执行周期--><crontabstart="true"schedule="1200"><!--600mins--><crontabfilterstart="true">

<excludeexpression="(.\*)\.html"></exclude><excludeexpression="(.\*)\.php"></exclude>

<excludeexpression="(.\*)\.jpg"></exclude>

<excludeexpression="(.\*)\.gif"></exclude>

<excludeexpression="(.\*)\.png"></exclude>

<excludeexpression="(.\*)\.css"></exclude>

<excludeexpression="(.\*)\.js"></exclude>

<excludeexpression="^tmp/\*"></exclude>

</crontabfilter></crontab><pluginstart="false"name="command"/></sersync>

<pluginname="command"><paramprefix="/bin/sh"suffix=""ignoreError="true"/><!--prefix/opt/tongbu/mmm.shsuffix--><filterstart="false"><includeexpression="(.\*)\.php"/><includeexpression="(.\*)\.sh"/></filter></plugin><pluginname="socket"><localpathwatch="/opt/tongbu"><deshostip="192.168.138.20"port="8009"/></localpath></plugin><pluginname="refreshCDN"><localpathwatch="/data0/htdocs/cms.xoyo.com/site/"><cdninfodomainname="ccms.chinacache.com"port="80"username="xxxx"passwd="xxxx"/><sendurlbase="http://pic.xoyo.com/cms"/><regexurlregex="false"match="cms.xoyo.com/site([/a-zA-Z0-9]\*).xoyo.com/images"/></localpath></plugin></head>

三、启用同步功能：

再次强调，所有sersync客户端都需要安装rsync，并启动为服务。在上述配置文件中须将所有需要同步的客户端IP加入。

[root@mastersersync]#./sersync2-rsetthesystemparamexecute：echo50000000>/proc/sys/fs/inotify/max\_user\_watchesexecute：echo327679>/proc/sys/fs/inotify/max\_queued\_eventsparsethecommandparamoption:-rrsyncallthelocalfilestotheremoteserversbeforethesersyncworkdaemonthreadnum:10parsexmlconfigfilehostip:localhosthostport:8008willignoretheinotifycreateFileeventStartthecrontabEvery600minutesrsyncallthefilestotheremoteserversentirely开启crontab过滤功能:作整体同步的时候会对如下文件进行过滤\*.phpinfo/\*usersyncpassword-file:userisclientpasswordfileis/usr/local/sersync/rsync.passwdconfigxmlparsesuccesspleaseset/etc/rsyncd.confmaxconnections=0Manuallysersyncworkingthread12=1(primarythread)+1(failretrythread)+10(daemonsubthreads)Maxthreadsnumbersis:22=12(Threadpoolnums)+10(Subthreads)pleaseaccordingyourcpu，use-nparamtoadjustthecpurate------------------------------------------rsyncthedirectoryrecursivlytotheremoteserversonceworkingpleasewait...executecommand:cd/var/www/html&&rsync-artuz-R--delete./client@172.20.45.48::webhome--password-file=/usr/local/sersync/rsync.passwd--exclude="\*.php"--exclude="info/\*">/dev/null2>&1runthesersync:watchpathis:/var/www/html

如下命令进行初始化同步，并监视指定目录变化情况：

Tip：如果需要将sersync运行前，已经存在的所有文件或目录全部同步到远程，要以-r参数运行sersync，将本地与远程整体同步一次。如果设置了过滤器，即在xml文件中，filter为true，则暂时不能使用-r参数进行整体同步。-r参数将会无效。相关参数：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c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-mrefreshCDN开启刷新CDN模块参数-m:单独启用其他模块，使用-msocket开启socket模块参数-m:单独启用其他模块，使用-mhttp开启http模块不加-m参数，则默认执行同步程序可选功能与xml高级配置

<host hostip="localhost" port="8008"></host>

hostip与port是针对插件的保留字段，对于同步功能没有任何作用，保留默认即可。**filter文件过滤功能对于sersync监控的文件，会默认过滤系统的临时文件(以“.”开头，以“～”结尾)，除了这些文件外，可以自定义其他需要过滤的文件。**

**<filter start="true">**

**<exclude expression="(.\*)\.gz"></exclude>**

**<exclude expression="^info/\*"></exclude>**

**</filter>**

**将start设置为true，在exclude标签中，填写正则表达式，默认给出两个例子分别是过滤以”.gz”结尾的文件与过滤监控目录下的info路径（监控路径/info/\*），可以根据需要添加，但开启的时候，自己测试一下，正则表达式如果出现错误，控制台会有提示。相比较使用rsync的exclude功能，被过滤的路径，不会加入监控，大大减少rsync的通讯量。inotify监控参数设定（优化）对于inotify监控参数可以进行设置，根据您项目的特点优化srsync。**

**<inotify>**

**<delete start="true"/>**

**<createFolder start="true"/>**

**<createFile start="true"/>**

**</inotify>**

**对于大多数应用，可以尝试把createFile（监控文件事件选项）设置为false来提高性能，减少rsync通讯。因为拷贝文件到监控目录会产生create事件与close\_write事件，所以如果关闭create事件，只监控文件拷贝结束时的事件close\_write，同样可以实现文件完整同步。注意：强将createFolder保持为true，如果将createFolder设为false，则不会对产生的目录进行监控，该目录下的子文件与子目录也不会被监控。所以除非特殊需要，请开启。默认情况下对创建文件（目录）事件与删除文件（目录）事件都进行监控，如果项目中不需要删除远程目标服务器的文件（目录），则可以将delete参数设置为false，则不对删除事件进行监控。**

**Debug开启**

**<debug start="false"/>**

**设置为true，开启debug模式，会在sersync正在运行的控制台，打印inotify事件与rsync同步命令。XFS文件系统**

**<fileSystem xfs="false"/>**

**对于xfs文件系统的用户，需要将这个选项开启，才能使sersync正常工作.**

**文件监控与远程同步设置**

**<localpath watch="/opt/tongbu">**

**<remote ip="192.168.0.104" name="tongbu1"/>**

**<!--<remote ip="192.168.8.39" name="tongbu"/>-->**

**<!--<remote ip="192.168.8.40" name="tongbu"/>-->**

**</localpath>**

**Rsync参数配置**

**<rsync>**

**<commonParams params="-artuz"/>**

**<auth start="false" users="root" passwordfile="/etc/rsync.pas"/>**

**<userDefinedPort start="false" port="874"/><!-- port=874 -->**

**<timeout start="false" time="100"/><!-- timeout=100 -->**

**<ssh start="false"/>**

**</rsync>**

**commonParams可以用户自定义rsync参数，默认是-artuzauthstart=”false”设置为true的时候，使用rsync的认证模式传送，需要配置user与passwrodfile(–password-file=/etc/rsync.pas)，来使用。userDefinedPort当远程同步目标服务器的rsync端口不是默认端口的时候使用(–port=874)。timeout设置rsync的timeout时间（–timeout=100）。ssh使用rsync－essh的方式进行传输。失败日志脚步配置**

**<failLog path="/tmp/rsync\_fail\_log.sh" timeToExecute="60"/>**

**对于失败的传输，会进行重新传送，再次失败就会写入rsync\_fail\_log，然后每隔一段时间（timeToExecute进行设置）执行该脚本再次重新传送，然后清空该脚本。可以通过path来设置日志路径。**

**Crontab定期整体同步功能**

**<crontab start="false" schedule="600"><!--600mins-->**

**<crontabfilter start="false">**

**<exclude expression="\*.gz"></exclude>**

**<exclude expression="info/\*"></exclude>**

**</crontabfilter>**

**</crontab>**

**crontab可以对监控路径与远程目标主机每隔一段时间进行一次整体同步，可能由于一些原因两次失败重传都失败了，这个时候如果开启了crontab功能，还可以进一步保证各个服务器文件一致，如果文件量比较大，crontab的时间间隔要设的大一些，否则可能增加通讯开销。schedule这个参数是设置crontab的时间间隔，默认是600分钟如果开启了filter文件过滤功能，那么crontab整体同步也需要设置过滤，否则虽然实时同步的时候文件被过滤了，但crontab整体同步的时候如果不单独设置crontabfilter，还会将需过滤的文件同步到远程，crontab的过滤正则与filter过滤的不同，也给出了两个实例分别对应与过滤文件与目录。总之如果同时开启了filter与crontab，则要开启crontab的crontabfilter，并按示例设置使其与filter的过滤一一对应。**

**插件设置**

**<plugin start="false" name="command"/>**

**当设置为true的时候，将文件同步到远程服务器后会调用name参数指定的插件。**

**command插件**

**当文件同步完成后，会调用command插件，如同步文件是test.php，则test.php文件在改动之后，调用rsync同步到远程服务器后，调用command插件，执行**

**/bin/sh test.php suffix >/dev/null 2>&1**

**如果suffix设置了，则会放在inotify事件test.php之后如果ignoreError为true，则会添加>/dev/null2>&1当然还可以设置command的filter，当filter为ture，include可以只对正则匹配到的文件，调用command。**

**“refreshCDN”，就在同步过程中将文件发送到目的服务器后刷新cdn接口。如果不想使用，则将start属性设为false即可。如果需要使用其他插件，则查看其他plugin标签，将插件名称改为xml中其它插件的名称即可。该模块根据chinaCDN的协议，进行设计，当有文件产生的时候，就向cdn接口发送需要刷新的路径位置。刷新CDN模块需要配置的xml文件如下。**

**<plugin name="refreshCDN">**

**<localpath watch="/data0/htdocs/cms.xoyo.com/site/">**

**<cdninfo domainname="ccms.chinacache.com" port="80" username="xxxx" passwd="xxxx"/>**

**<sendurl base="http://pic.xoyo.com/cms"/>**

**<regexurl regex="false" match="cms.xoyo.com/site([/a-zA-Z0-9]\*).xoyo.com/images"/>**

**</localpath>**

**</plugin>**

**其中localpathwatch是需要监控的目录。**

**cdnifo标签制定了cdn接口的域名，端口号，以及用户名与密码。**

**sendurl标签是需要刷新的url的前缀。**

**regexurl标签中的，regex属性为true时候，使用match属性的正则语句匹配inotify返回的路径信息，并将正则匹配到的部分作为url一部分，**

**举例：**

**如果产生文件事件为：/data0/htdoc/cms.xoyo.com/site/jx3.xoyo.com/image/a/123.txt**

**经过上面的match正则匹配后，最后刷新的路径是:**

**http://pic.xoyo.com/cms/jx3/a/123.txt;**

**如果regex属性为false，最后刷新的路径是**

**http://pic.xoyo.com/cms/jx3.xoyo.com/images/a/123.txt;**

**socket插件**

**socket插件，开启该模块，则向指定ip与端口发送inotify所产生的文件路径信息。**

**Http插件**

**http插件，可以向指定域名的主机post,inotify监控的事件。**

**单独运行插件**

**插件也可以单独使用，即不对远程目标机进行同步，直接调用插件:**

**只调用command插件**

**./sersync -d -m command**

**只调用refreshCDN插件**

**./sersync -d -m refreshCDN**

**只调用socket插件**

**./sersync -d -m socket**

**只调用http插件**

**./sersync -d -m http**