# Tomcat应用服务器脚本更新方案说明

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 撰稿人 | 参与 | 审核 | 日期 |
| V1.0 | 张凯 | 陈智慧 | 坞扬帆 | 2016年9月18日 |
|  |  |  |  |  |

目录

[Tomcat应用服务器脚本更新方案说明 1](#_Toc461962243)

[一、应用场景说明 2](#_Toc461962244)

[二、Unison 和Inotify 简单介绍 2](#_Toc461962245)

[三、Unison 和Inotify安装及脚本 3](#_Toc461962246)

[四、. 配置双机信任(SSH) 4](#_Toc461962247)

[五、Unison 同步脚本参数 5](#_Toc461962248)

[六、Inotify 监控脚本 7](#_Toc461962249)

[七、Inotify 脚本 7](#_Toc461962250)

[八、自动化安装脚本： 9](#_Toc461962251)

# 一、应用场景说明

由于目前生产上keepalived设置了不抢占模式，若master的配置文件经常改了之后 ，slave未同步，容易造成访问问题；因此引入Inotify，监控文件系统操作，通过和Unison配合可以做到将数据实时的进行同步，保持主备两服务器的配置文件实时同步。

# 二、Unison 和Inotify 简单介绍

1） Unison是一款跨windows/linux/MAC OS平台的文件同步工具，不仅支持本地对本地同步，也支持通过SSH、RSH和Socket等网络协议进行同步。更棒的是，Unison支持双向同步操作。

2）Inotify可以监控文件系统操作，比如读取、写入和创建等，通过和Unison配合可以做到将数据实时的进行同步。

# 三、Unison 和Inotify安装及脚本

在安装unison之前需要首先安装ocaml软件

1）Ocaml安装：

#wget http://caml.inria.fr/pub/distrib/ocaml-4.03/ocaml-4.03.0.tar.gz

# tar -zxf ocaml-4.03.0.tar.gz

# cd  ocaml-4.03.0

# ./configure

# make world opt

# make install

#make clean

2）Unison的安装：

#wget http://fossies.org/linux/misc/unison-2.48.4.tar.gz

# tar -zxvf unison-2.48.4.tar.gz

# cd  src

# make UISTYLE=text

# mkdir ~/bin #否则提示找不到/root/bin文件，无法完成unison执行文件的拷贝。

# make install #第一次执行出现错误，再次执行一次就可以了。

3）Inotify 安装

#wget http://tenet.dl.sourceforge.net/project/inotify-tools/inotify-tools/3.13/inotify-tools-3.13.tar.gz

#cd inotify-tools-3.13

# ./configure

# make

# sudo make install

# 四、. 配置双机信任(SSH)

###在server1中执行.

#ssh-keygen –t rsa #在server1中生成授权密钥，不要输入密钥密码.

#scp ~/.ssh/id\_rsa.pub root@server2:/root #将server1产生的密钥拷贝至server2.

####以下在server2中执行.

#chmod 700 /root/.ssh -R

#touch  /root/.ssh/authorized\_keys

#cat /root/id\_rsa.pub >> /root/.ssh/authorized\_keys

#chmod 600 /root/.ssh/authorized\_keys

###在server2中执行.

#ssh-keygen –t rsa #在server2中生成授权密钥，不要输入密钥密码.

#scp ~/.ssh/id\_rsa.pub root@server2:/root #将server2产生的密钥拷贝至server2.

####以下在server1中执行.

#chmod 700 /root/.ssh -R

# touch  /root/.ssh/authorized\_keys

#cat /root/id\_rsa.pub >> /root/.ssh/authorized\_keys

#chmod 600 /root/.ssh/authorized\_keys

# 五、Unison 同步脚本参数

/root/bin/unison -servercmd=/root/bin/unison /root/test root@192.168.1.10//root/test -path=test1/ -path=test2/ -times -owner=true -group=true -fastcheck=true –batch –perms=-1 -silent -logfile=/root/.unison/unison.log

参数说明：

force表示会以本地所指定文件夹为标准，将该目录同步到远端。这里需要注意，如果指定了force参数，那么Unison就变成了单项同步了，也就是说会以force指定的文件夹为准进行同步，类似与rsync。

Unison双向同步基本原理是：假如有A B两个文件夹，A文件夹把自己的改动同步到B，B文件夹也把自己的改动同步到A，最后A B两文件夹的内容相同，是AB文件夹的合集。

Unison双向同步的一个缺点是，对于一个文件在两个同步文件夹中都被修改时，unison是不会去同步的，因为unison无法判断以那个为准。

ignore = Path表示忽略指定目录，即同步时不同步它。

batch = true，表示全自动模式，接受缺省动作，并执行。

-fastcheck true 表示同步时仅通过文件的创建时间来比较，如果选项为false，Unison则将比较两地文件的内容。

log = true 表示在终端输出运行信息。

logfile 指定输出的log文件。

另外，Unison有很多参数，这里仅介绍常用的几个，详细的请参看Unison手册。

-auto //接受缺省的动作，然后等待用户确认是否执行。

-batch //batch mode, 全自动模式，接受缺省动作，并执行。

-ignore xxx //增加 xxx 到忽略列表中

-ignorecase [true|false|default] //是否忽略文件名大小写

-follow xxx //是否支持对符号连接指向内容的同步

owner = true //保持同步过来的文件属主

group = true //保持同步过来的文件组信息

perms = -1 //保持同步过来的文件读写权限

repeat = 1 //间隔1秒后,开始新的一次同步检查

retry = 3 //失败重试

sshargs = -C //使用ssh的压缩传输方式

xferbycopying = true"

-immutable xxx //不变目录，扫描时可以忽略

-silent //安静模式

-times //同步修改时间

-path xxx 参数 //只同步 -path 参数指定的子目录以及文件，而非整个目录，-path 可以多次出现

# 六、Inotify 监控脚本

同步目录：/root/test/test1

Unison 首次执行同步脚本：

#/root/bin/unison -servercmd=/root/bin/unison /root/test/ ssh://root@$IP//root/test/ -path=test1/ -path=test2 -times -owner=true -group=true -fastcheck=true -perms=-1 -logfile=/root/.unison/unison.log

  注：首次进行同步时，必须一一台服务器为基准进行同步，否则双方文件不一致，unison将忽略不一致文件的同步。

# 七、Inotify 脚本

Server1服务器：

#/bin/bash

#源目录

src1="/root/test1"

src2="/root/test2/"

#目的目录

dst1="/root/test1/"

dst2="/root/test2/"

####

/usr/local/bin/inotifywait -mrq -e create,delete,modify,move,attrib $src1 $src2 | while read line; do

/root/bin/unison -servercmd=/root/bin/unison -times -owner=true -group=true -fastcheck=true –perms=-1 –silent -batch $src1 ssh://192.168.1.11/$dst1

/root/bin/unison -servercmd=/root/bin/unison -times -owner=true -group=true -fastcheck=true –perms=-1 –silent -batch $src1 ssh://192.168.1.12/$dst2

echo -n "$line " >> /var/log/inotify.log

echo `date | cut -d " " -f1-5` >> /var/log/inotify.log

done

server2服务器：

#/bin/bash

#源目录

src1="/root/test1"

src2="/root/test2/"

#目的目录

dst1="/root/test1/"

dst2="/root/test2/"

 /usr/local/bin/inotifywait -mrq -e create,delete,modify,move,attrib $src1 $src2 | while read line; do

 /root/bin/unison -servercmd=/root/bin/unison -times -owner=true -group=true -fastcheck=true –perms=-1 –silent -batch $src1 ssh://192.168.1.10/$dst1

 /root/bin/unison -servercmd=/root/bin/unison -times -owner=true -group=true -fastcheck=true –perms=-1 –silent -batch $src1 ssh://192.168.1.10/$dst2

  echo -n "$line " >> /var/log/inotify.log

  echo `date | cut -d " " -f1-5` >> /var/log/inotify.log

 done

启动两台服务器的inotify脚本文件，有文件修改时软件将进行实时同步。

注：此软件不合适不同服务器对同一文件的修改，同一文件同时被修改时，unison无法判断以谁为准无法进行同步。

# 八、自动化安装脚本：

#/bin/env /bin/bash

#在安装unison之前需要首先安装ocaml软件。

osaml\_install()

{

[ ! -f ./ocaml-4.03.0.tar.gz ] && wget http://caml.inria.fr/pub/distrib/ocaml-4.03/ocaml-4.03.0.tar.gz

[ ! -d ./ocaml-4.03.0 ] && tar -zxf ocaml-4.03.0.tar.gz

cd ocaml-4.03.0

./configure

make world opt

make install

make clean

}

unison\_install()

{

[ ! -f ./unison-2.48.4.tar.gz ] && wget http://fossies.org/linux/misc/unison-2.48.4.tar.gz

[ ! -d ./src ] && tar -zxvf unison-2.48.4.tar.gz

cd src

make UISTYLE=text

mkdir /root/bin #否则提示找不到/root/bin文件，无法完成unison执行文件的拷贝。

mkdir ~/bin

make install #第一次执行出现错误，再次执行一次就可以了。

if (($?==0));then

echo "install success"

else

make install

fi

}

inotify\_install()

{

# Inotify 安装

[ ! -f ./inotify-tools-3.13.tar.gz ] && wget http://tenet.dl.sourceforge.net/project/inotify-tools/inotify-tools/3.13/inotify-tools-3.13.tar.gz

[ ! -d ./inotify-tools-3.13 ] && tar zxf inotify-tools-3.13.tar.gz

cd inotify-tools-3.13

./configure

make

sudo make install

}

osaml\_install

unison\_install

inotify\_install