**SVN简介和工作原理**

subversion（简称svn）是近几年崛起的版本管理软件，是cvs的接班人，目前绝大多数开源软件都使用svn作为代码版本管理软件。Subversion支持[linux](http://www.ha97.com/tag/linux)和windows，但较多安装在linux下。

svn服务器有两种运行方式：独立服务器和借助于apache。 svn://或http://

svn客户端tortoisesvn

svn的基本工作原理： 在一台服务器上建立一个源代码库，库里可以存放许多不同项目的源程序。有源代码库管理员统一管理这些源程序。每个用户在使用源代码库之前，首先要把源代码库里德项目文件下载到本地，然后开发人员可以在本地修改，左后用svn命令进行提交，游源代码库统一管理修改。

版本控制解决了：

\*代码管理混乱

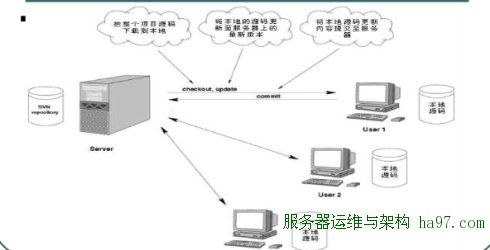
\*解决代码冲突困难

\*在代码整合期间引发bug

\*无法对代码的拥有者进行权限控制

\*项目不同版本的发布困难

SVN工作原理示意图：



subversion目录说明：

\*dav目录：是提供apache与mod\_dav\_svn使用的目录，让他们存储内部数据

\*db目录：就是所有版本控制的数据存放文件

\*hooks目录：放置hook脚本文件的目录

\*locks目录：用来放置subversion见艰苦锁定数据的目录，用来追踪存取文件库的客户端

\*format文件：是一个文本文件，里面只放了一个整数。表示当前文件库配置的版本号

\*conf目录：是这个仓库的配置文件（仓库的用户访问账号、权限等）

**SVN Server详细配置手册**

系统环境

CentOS 5.8 最小化安装(关闭iptables和selinux) + ssh + yum

一，安装必须的软件包.

yum install subversion [mysql](http://www.ha97.com/tag/mysql)-server httpd mod\_dav\_svn mod\_perl sendmail wget gcc-c++ make [unzip](http://www.ha97.com/tag/unzip) perl\* ntsysv vim-enhanced

说明：

subversion (SVN服务器)

mysql-server (用于codestriker)

httpd mod\_dav\_svn mod\_perl (用于支持WEB方式管理SVN服务器)

sendmail (用于配置用户提交代码后发邮件提醒)

wget gcc-c++ make unzip perl\* (必备软件包)

ntsysv vim-enhanced (可选)

二，基本的SVN服务器配置

1，新建一个目录用于存储SVN所有文件

# mkdir /home/svn

2，新建一个版本仓库

# svnadmin create /home/svn/project

3，初始化版本仓库中的目录

# mkdir project project/server project/client project/test (建立临时目录)

# svn import project/ <file:///home/svn/project> -m “初始化SVN目录”

# rm -rf project (删除临时建立的目录)

4，添加用户

要添加SVN用户非常简单，只需在/home/svn/project/conf/passwd文件添加一个形如“username=password”的条目就可以了。为了测试，我添加了如下内容:

[users]

# harry = harryssecret

# sally = sallyssecret

pm = pm\_pw

server\_group = server\_pw

client\_group = client\_pw

test\_group = test\_pw

5，修改用户访问策略

/home/svn/project/conf/authz记录用户的访问策略，以下是参考:

[groups]

project\_p = pm

project\_s = server1,server2,server3

project\_c = client1,client2,client3

project\_t = test1,test1,test1

[project:/]

@project\_p = rw

\* =

[project:/server]

@project\_p = rw

@project\_s = rw

\* =

[project:/client]

@project\_p = rw

@project\_c = rw

\* =

[project:/doc]

@project\_p = rw

@project\_s = r

@project\_c = r

@project\_t = r

\* =

**说明：以上信息表示，只有project\_p用户组有根目录的读写权。r表示对该目录有读权限，w表示对该目录有写权限，rw表示对该目录有读写权限。最后一行的\* =表示，除了上面设置了权限的用户组之外，其他任何人都被禁止访问本目录。这个很重要，一定要加上！**

6，修改svnserve.conf文件,让用户和策略配置升效.

svnserve.conf内容如下:

[general]

anon-access = none

auth-access = write

password-db = /home/svn/project/conf/passwd

authz-db = /home/svn/project/conf/authz

7，启动服务器

# svnserve -d -r /home/svn

**注意：如果修改了svn配置，需要重启svn服务，步骤如下：**

# ps -aux|grep svnserve

# kill -9 ID号

# svnserve -d -r /home/svn

8，测试服务器

# svn co svn://192.168.60.10/project

Authentication realm: <svn://192.168.60.10:3690> 92731041-2dae-4c23-97fd-9e1ed7f0d18d

Password for 'root':

Authentication realm: <svn://192.168.60.10:3690> 92731041-2dae-4c23-97fd-9e1ed7f0d18d

Username: server\_group

Password for 'server\_group':

svn: Authorization failed ( server\_group没用根目录的访问权 )

# svn co svn://192.168.60.10/project

Authentication realm: <svn://192.168.60.10:3690> 92731041-2dae-4c23-97fd-9e1ed7f0d18d

Password for ‘root’:

Authentication realm: <svn://192.168.60.10:3690> 92731041-2dae-4c23-97fd-9e1ed7f0d18d

Username: pm

Password for ‘pm’:

A    project/test

A    project/server

A    project/client

Checked out revision 1.  ( 测试提取成功 )

# cd project/server

# vim main.c

# svn add main.c

# svn commit main.c -m “测试一下我的C程序,看什么看,不行啊??”

Adding         main.c

Transmitting file data .

Committed revision 2.  ( 测试提交成功 )

三，配置SVN服务器的HTTP支持

1，转换SVN服务器的密码

由于SVN服务器的密码是明文的，HTTP服务器不与支持，所以需要转换成HTTP支持的格式。我写了一个Perl脚本完成这个工作.

脚本内容如下:

# cd /home/svn/project/conf/

# vim PtoWP.pl

#!/usr/bin/perl  
# write by huabo, 2009-11-20

use warnings;  
use strict;

#open the svn passwd file  
open (FILE, "passwd") or die ("Cannot open the passwd file!!!\n");

#clear the apache passwd file  
open (OUT\_FILE, ">webpasswd") or die ("Cannot open the webpasswd file!!!\n");  
close (OUT\_FILE);

#begin  
foreach (<FILE>) {  
if($\_ =~ m/^[^#].\*=/) {  
$\_ =~ s/=//;  
`htpasswd -b webpasswd $\_`;  
}  
}

# chmod +x PtoWP.pl

# ./PtoWP.pl

Adding password for user pm

Adding password for user server\_group

Adding password for user client\_group

Adding password for user test\_group

现在目录下会多一个webpasswd文件。

2，修改httpd.conf，添加关于SVN服务器的内容

编辑/etc/httpd/conf/httpd.conf，在最后添加如下信息:

<Location /project>

DAV svn

SVNPath /home/svn/project/

AuthType Basic

AuthName "svn for project"

AuthUserFile /home/svn/project/conf/webpasswd

AuthzSVNAccessFile /home/svn/project/conf/authz

Satisfy all

Require valid-user

</Location>

3，修改svn目录的属主为apache帐号：chown -R apache.apache /home/svn/project/

（**注意：原文少了这一步，会出权限问题。**）

4，重启Web服务器：

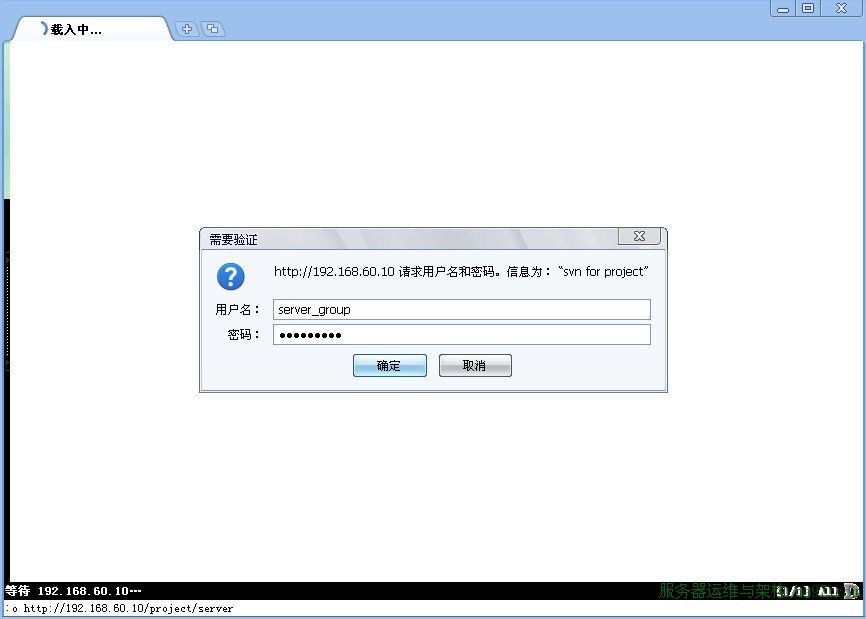
# /etc/init.d/httpd restart

Stopping httpd: [FAILED]

Starting httpd: [ OK ]

5，用浏览器访问http://192.168.60.10/project/server/测试

测试结果如下图所示：



( 测试成功 )

四，配置邮件提醒支持

1，安装Perl模块Module::Build

# wget <http://search.cpan.org/CPAN/authors/id/D/DA/DAGOLDEN/Module-Build-0.36_11.tar.gz>

# tar xvf Module-Build-0.36\_11.tar.gz

# cd Module-Build-0.36\_11

# perl Build.PL

# ./Build

# ./Build test

# ./Build install

# cd ..

2，安装Perl模块Authen::SASL

# wget <http://search.cpan.org/CPAN/authors/id/G/GB/GBARR/Authen-SASL-2.15.tar.gz>

# tar xvf Authen-SASL-2.15.tar.gz

# cd Authen-SASL-2.15

# perl Makefile.PL

# make test

# make install

# cd ..

3，安装Perl模块Net::SMTP\_auth

# wget <http://search.cpan.org/CPAN/authors/id/A/AP/APLEINER/Net-SMTP_auth-0.08.tar.gz>

# tar xvf Net-SMTP\_auth-0.08.tar.gz

# cd Net-SMTP\_auth-0.08

# perl Makefile.PL

# make test

# make install

# cd ..

4，安装Perl模块SVN::Notify

# wget <http://search.cpan.org/CPAN/authors/id/D/DW/DWHEELER/SVN-Notify-2.80.tar.gz>

# tar xvf SVN-Notify-2.80.tar.gz

# cd SVN-Notify-2.80

# perl Build.PL

# ./Build

# ./Build test

# ./Build install

# cd ..

5，启动邮件服务器

# service sendmail restart

Shutting down sendmail:                                    [FAILED]

Starting sendmail:                                         [  OK  ]

Starting sm-client:                                        [  OK  ]

6，配置自动发邮件脚本

修改post-commit脚本,以支持邮件通知功能.

# cd /home/svn/project/hooks/

# vim post-commit

内容如下:

#!/bin/sh

REPOS="$1"

REV="$2"

/usr/bin/svnnotify –repos-path “$1” –revision “$2” –to caodaijun@pica.com –from caodaijun@feinno.com –handler “HTML::ColorDiff”  –with-diff –smtp localhost –smtp-user root –smtp-pass 5201314318 -c “UTF-8” -g zh\_CN -o raw –svnlook /usr/bin/svnlook –subject-prefix ‘[SVN Update]’

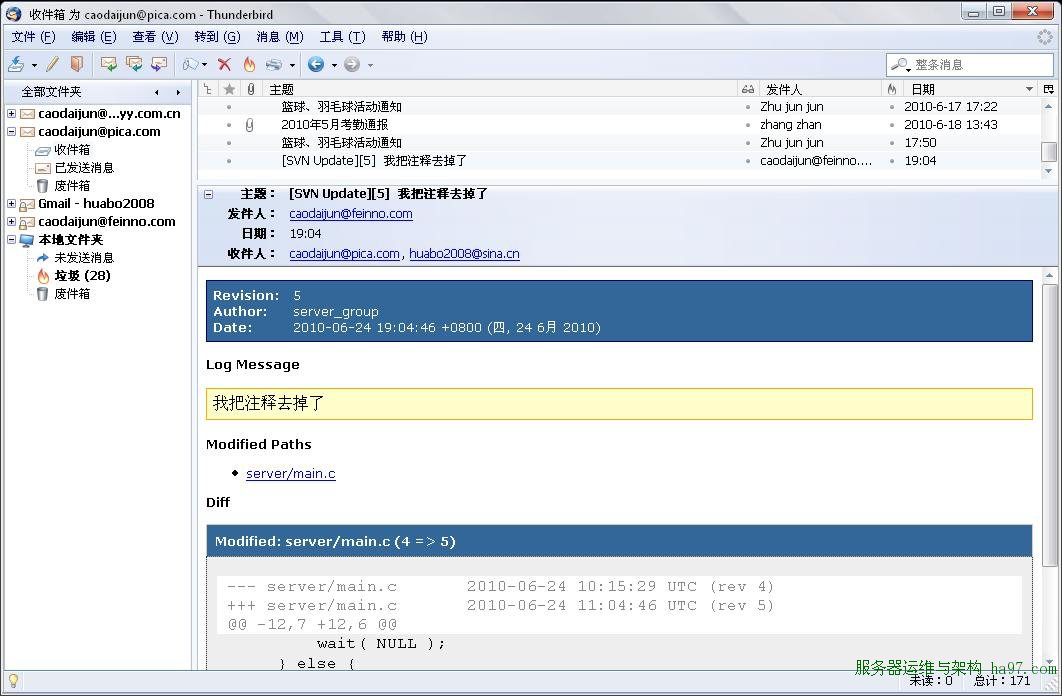
(to参数代表接收邮件的地址，可以有多个，当你有多个老大的时候，这就很重要了，：）。from参数是虚拟的，代表你的发送地址，一般情况下，这个参数 不重要，但如果接收者的邮件服务器有反垃圾邮件的功能，需要判定源地址的话，这个参数是否合法就显得很重要了)

再给该脚本添加可执行权限

# chmod +x post-commit

7，再次提交时，就会给指定邮件地址发信了。

如下图所示:



五，其它常用配置

1，强制写log脚本

配置pre-commit文件，要求用户每次更新文件都必须写log.

# cd /home/svn/project/hooks/

# vim pre-commit

文件内容如下:

#!/bin/sh

REPOS="$1"

TXN="$2"

SVNLOOK=/usr/bin/svnlook

LOGMSG=`$SVNLOOK log -t "$TXN" "$REPOS" | grep "[a-zA-Z0-9]" | wc -c`

if [ "$LOGMSG" -lt 5(要求的log长度，依实际需要修改) ];

then

echo -e "\nEmpty log message not allowed. Commit aborted!" 1>&2

exit 1

fi

配置完成后，给本件加上可执行权限。再提交代码时，就必须按要求写注释了，：）

2，可修改log脚本

配置pre-revprop-change文件，此文件在show log中修改log时会运行，得到修改的权限，否则会报错:DAV request failed; it’s possible that the repository’s pre-revprop-change hook either failed or is non-existent. At least one property change failed; repository is unchanged

# cd /home/svn/project/hooks/

# vim pre-revprop-change

文件内容如下:

REPOS="$1"

REV="$2"

USER="$3"

PROPNAME="$4"

if ["$PROPNAME" = "svn:log"];then exit 0;fi

exit 1

配置完后加可执行权限升效。

六，备份管理

svn服务器的定期备份是很重要的，最简单的方式是定时备份仓库目录。

1，新建备份目录

# mkdir /opt/project\_backup

2，编写备份脚本

# cd /home/svn/

# vim project\_backup.sh

内容如下:

#!/bin/bash

#write by huabo, 2009-11-20

cd /home/svn

now=`/bin/date +%Y%m%d`

/bin/tar czvf “project\_backup\_$now.tar.gz” project/ && rm -rf /opt/project\_backup/\* && /bin/mv project\_backup\_\*.tar.gz /opt/project\_backup/

if [ $? == 0 ]

then

result=”OK!!”

else

result=”False!!”

fi

#send mail to administrator

/bin/mail caodaijun@pica.com -s “project\_backup\_$now” <<MESSAGE

Result: `/bin/echo $result`

MESSAGE

给该脚本添加可执行权限。

3，设定每天定时执行该脚本.

# crontab -e

输入如下内容:

0 23 \* \* \* /home/svn/project\_backup.sh

表示每天晚上23点运行此脚本。

经过以上三步操作，就可以自动备份SVN资料了，且不论备份是否成功，都会给用户发送邮件信息。

七，用svnstat[分析](http://www.ha97.com/tag/%e5%88%86%e6%9e%90)SVN数据.

1，安装JAVA

svnstat是JAVA应用程序，需要先安装JAVA环境。

下载jre，URL: <http://javadl.sun.com/webapps/download/AutoDL?BundleId=39484>

安装:

# chmod +x jre-6u20-linux-i586-rpm.bin

# ./jre-6u20-linux-i586-rpm.bin

2，下载svnstat

# wget <http://downloads.sourceforge.net/project/svnstat/svnstat/Release-1.0/SvnStat-1.0.zip?use_mirror=jaist>

# unzip SvnStat-1.0.zip

3，更新代码

# pwd

/root

# svn co svn://192.168.60.10/project

A    project/test

A    project/server

A    project/server/main.c

A    project/client

Checked out revision 5.

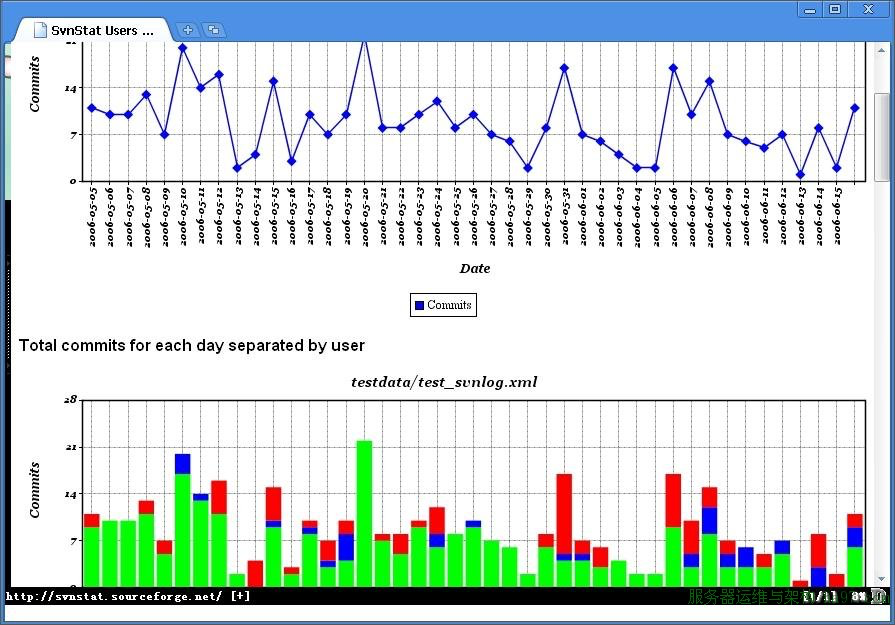
4，生成svnstat数据

# svn log project -v –xml –non-interactive > project.log

# cd SvnStat-1.0

# java -classpath SvnStat-all.jar de.agentlab.svnstat.SvnStat -jar SvnStat-all.jar -r /root/project.log -d /var/www/html/

5，用浏览器登录即可看到许多统计出来的图表。大致如下图所示：



八，用statsvn分析SVN数据

1，下载statsvn

# wget <http://downloads.sourceforge.net/project/statsvn/statsvn/0.7.0/statsvn-0.7.0.zip?use_mirror=jaist>

# unzip statsvn-0.7.0.zip

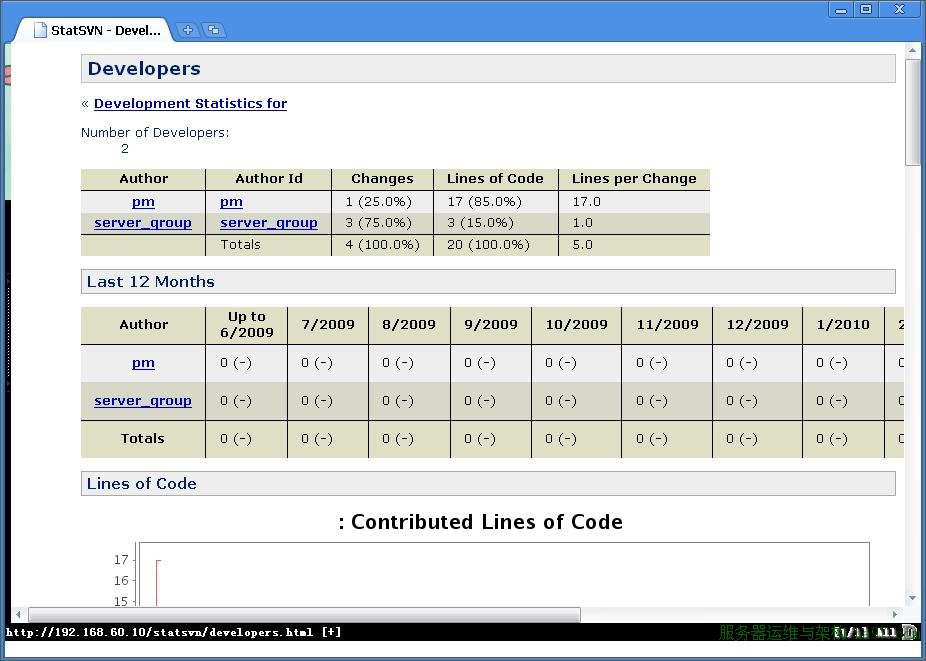
# cd statsvn-0.7.0

2，生成statsvn数据

# mkdir /var/www/html/statsvn

# java -jar statsvn.jar -verbose -output-dir /var/www/html/statsvn/ /root/project.log /root/project

3，用浏览器测测试效果如下图：



九，配置codestriker.

1.安装codestriker依赖的perl包.

# perl -MCPAN -e ‘install “Template”‘

2.下载codestriker

# wget <http://downloads.sourceforge.net/project/codestriker/codestriker/1.9.10/codestriker-1.9.10.tar.gz?use_mirror=jaist&ts=1279246587>

# mkdir /var/www/codestriker

# cd /var/www/codestriker

# tar xvf /path/codestriker-1.9.10.tar.gz

# chown -R apache.apache codestriker-1.9.10

3.配置数据库

# service mysqld restart

# mysql -uroot mysql

执行:

CREATE [DATABASE](http://www.ha97.com/category/database) codestrikerdb CHARACTER SET utf8;

GRANT SELECT,INSERT,UPDATE,DELETE,INDEX,ALTER,CREATE,DROP,REFERENCES ON codestrikerdb.\* TO codestriker@localhost IDENTIFIED BY ‘cspasswd’;

FLUSH PRIVILEGES;

quit

4.配置codestriker

# cd codestriker-1.9.10/

# vim codestriker.conf

注意以下几点(详细可查看codestriker的安装文档)

a.数据库的用户名密码要配对

b.svn的数据仓库要配对，我的如下：

@valid\_repositories =

(

‘svn:file:///home/svn/project’,

)

5.执行codestriker的安装脚本

# cd bin/

# ./install.pl

6.配置http支持

# vim /etc/httpd/conf/httpd.conf

在最后面加上如下内容:

Alias /codestriker/  /var/www/codestriker/codestriker-1.9.10/cgi-bin/

Alias /codestrikerhtml/  /var/www/codestriker/codestriker-1.9.10/html/

<Directory “/var/www/codestriker/codestriker-1.9.10/cgi-bin/”>

SetHandler perl-script

PerlHandler ModPerl::Registry

Options +ExecCGI

</Directory>

<Directory “/var/www/codestriker/codestriker-1.9.10/html/”>

AllowOverride None

Allow from all

</Directory>

7.重启Web服务

# service httpd restart

8.在浏览器中输入http://192.168.60.10/codestriker/codestriker.pl即可访问，如下图：



参考：http://wiki.centos.org/HowTos/Subversion

*来自 <*[*http://www.ha97.com/4467.html*](http://www.ha97.com/4467.html)*>*