双轨数字学校

安装部署手册

目录

[安装jdk 3](#_Toc437251151)

[安装Tomcat6.0 4](#_Toc437251152)

[安装MySQL 5](#_Toc437251153)

[安装swftools 6](#_Toc437251154)

[安装libreoffice 7](#_Toc437251155)

[安装语言包xpdf-chinese-simplified.tar.gz 7](#_Toc437251156)

[安装Ngix1.2.7(或者更高版本、测试1.2.7版本线上能稳定运行) 7](#_Toc437251157)

[安装PHP5.3.19版本（注意:版本不能不能小于5.2） 13](#_Toc437251158)

Linux环境下Tomcat的安装和配置

**安装jdk**

1.下载一个Linux Platform的JDK，建议下载RPM自解压格式的

（比如：jdk-6u29-linux-i586.bin）；

下载网站为http://java.sun.com

2.拷贝到Linux环境下（本系统目录/usr/java），在shell下执行命令：

（注意应先切换到文件所在的目录，命令为cd）

# chmod 755 jdk-6u29-linux-i586.bin //更改文件权限为755

# ./ jdk-6u29-linux-i586.bin //执行文件

这时会有一段Sun的协议，敲几次空格键，当询问是否同意的时候，敲yes就可以了。当协议最后出现“Done”时，表示协议执行完毕。

3.设置环境变量

通过更改/etc/profile来实现，命令为：

#sudo gedit /etc/profile

然后在打开的profile里面添加如下代码：

export JAVA\_HOME=/usr/java/jdk1.6.0\_37(安装目录下jdk文件夹名称)

export PATH=$JAVA\_HOME/bin:$PATH/

export CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar

下面来验证一下变量设置是否生效（在验证前先logout一下，再重新登陆）；

export JAVA\_HOME=/usr/java/jdk1.6.0\_37

export JRE\_HOME=/usr/java/jdk1.6.0\_37/jre

export PATH=$PATH:$JAVA\_HOME/bin:$JRE\_HOME/bin

export CLASSPATH=.:$JAVA\_HOME/lib/jt.jar:$JAVA\_HOME/lib/tools.jar:$JRE\_HOME/lib

# java -version

JAVA version "1.6.0"

JAVA(TM) 2 Runtime Environment, Standard Edition (build 1.6.0)

JAVA HotSpot(TM) Client VM (build 1.6.0, mixed mode)

4. 至此，Linux上JDK的安装完毕。

5.卸载linux自带jdk

RHEL U3 自身带一个JDK，用起来对应用软件有影响

下面是卸载步骤：

用root用户登陆到系统，打开一个终端输入

# rpm -qa|grep gcj

显示内容其中包含下面两行信息

# java-1.4.2-gcj-compat-1.4.2.0-27jpp

# java-1.4.2-gcj-compat-devel-l.4.2.0-27jpp

卸载

# rpm –e --nodeps java-1.4.2-gcj-compat-devel-l.4.2.0-27jpp

# rpm -e --nodeps java-1.4.2-gcj-compat-l.4.2.0-27jpp

注意：如有错误可更换删除顺序。

**安装Tomcat6.0**

1. 配置tomcat

#cd /root

#tar zxvf apache-tomcat-6.0.35.tar.gz //解包

将解压后文件夹名称为apache-tomcat-6.0.35的文件夹拷贝到/usr/local/下

测试Tomcat能否正常启动：

启动Tomcat：

# cd /usr/local/apache-tomcat-6.0.35/bin

#./startup.sh

关闭Tomcat：

# cd /usr/local/apache-tomcat-6.0.35/bin

#./shutdown.sh

若看到以下信息：

Using CATALINA\_BASE: /usr/local/ apache-tomcat-6.0.35

Using CATALINA\_HOME: /usr/local / apache-tomcat-6.0.35

Using CATALINA\_TMPDIR: /usr/local / apache-tomcat-6.0.35 /temp

Using JAVA\_HOME: /usr/local /jdk1.6.0\_37

证明apache-tomcat-6.0.35配置OK了！

在/usr/local/ apache-tomcat-6.0.35 /conf/catalina/localhost目录下新建一个eschool.xml文件，

文件内容如下：

<?xml version="1.0" encoding="UTF-8"?>

<Context crossContext="false" debug="5" docBase="/home/eschool" path="/eschool" reloadable="false">

<Resource name="jdbc/eschool" auth="Container" type="javax.sql.DataSource" username="root" password="" driverClassName="com.mysql.jdbc.Driver" maxIdle="10" maxWait="-1" url="jdbc:mysql://服务器IP:3306/eschool" maxActive="50" defaultSessionTimeOut="-1"/>

</Context>

然后打开浏览器，输入http://localhost:8080，如果您能看到一只小猫tomcat安装成功!

配置项目位置：

将项目源码包复制到/home/目录下

**安装MySQL**

安装MySQL需要以下文件：

Cmake-2.8.3.tar.gz  
　　mysql-5.5.18.tar.gz　　  
创建用户  
useradd -M -s /sbin/nologin mysql

创建mysql文件夹

mkdir -p /usr/local/mysql/data

进入文件夹

解压

tar -zxvf cmake-2.8.4.tar.gz

cd cmake-2.8.4

编译安装

./bootstrap

make

make install

cd ../

安装mysql

tar -zxvf mysql-5.5.18.tar.gz

cd mysql-5.5.18

/usr/local/bin/cmake -DCMAKE\_INSTALL\_PREFIX=/usr/local/mysql \

-DMYSQL\_UNIX\_ADDR=/usr/local/mysql/mysql.sock \

-DDEFAULT\_CHARSET=utf8 \

-DDEFAULT\_COLLATION=utf8\_general\_ci \

-DWITH\_EXTRA\_CHARSETS:STRING=utf8,gbk,gb2312 \

-DWITH\_MYISAM\_STORAGE\_ENGINE=1 \

-DWITH\_INNOBASE\_STORAGE\_ENGINE=1 \

-DWITH\_MEMORY\_STORAGE\_ENGINE=1 \

-DWITH\_READLINE=1 \

-DENABLED\_LOCAL\_INFILE=1 \

-DMYSQL\_DATADIR=/usr/local/mysql/data \

-DMYSQL\_USER=mysql \

-DWITH\_DEBUG=0

make

make install

赋予文件夹权限

chmod +w /usr/local/mysql

chown -R mysql:mysql /usr/local/mysql

cp support-files/my-small.cnf /usr/local/mysql/my.cnf

chown mysql:mysql /usr/local/mysql/my.cnf

cp /usr/local/mysql/support-files/mysql.server /etc/rc.d/init.d/mysqld

chmod 755 /etc/init.d/mysqld

chkconfig --level 345 mysqld on

初始化mysql

######mysql\_install\_db mysql##########

/usr/local/mysql/scripts/mysql\_install\_db --user=mysql --basedir=/usr/local/mysql --datadir=/usr/local/mysql/data

启动mysql

######start mysqlserver#####

/sbin/service mysqld start

配置环境变量

#####modify profile######

echo 'export PATH=$PATH:/usr/local/mysql/bin' >> /etc/profile

source /etc/profile

#####modify mysql privileges####

##设置root密码mysql

/usr/local/mysql/bin/mysqladmin -u root password mysql

##给客户端登入的用户设置权限，

在mysql下：

grant all privileges on \*.\* to root@'%' identified by 'mysql' with grant option;

flush privileges;

###

/sbin/service mysqld restart

**安装swftools**

1. 下载swftools-0.9.2.tar.gz安装包

2. 将安装包拷贝到/usr/local目录下

3. 解压swftools 解压命令tar zxvf swftools-0.9.2.tar.gz

4. 进入 swftools-0.9.2 编译安装

5. 运行命令 ./configure && make &&make install

问题：报错　-o –L

1. find ./ |xargs grep -r -i "default\_viewer"
2. /swftools-0.9.2/swfs/Makefile

将-o -L去掉；

运行结束即可

**安装libreoffice**

下载解压下列包

tar -zxvf LibreOffice\_5.0.0\_Linux\_x86-64\_rpm.tar.gz

tar -zxvf LibreOffice\_5.0.0\_Linux\_x86-64\_rpm\_helppack\_zh-CN.tar.gz

tar -zxvf LibreOffice\_5.0.0\_Linux\_x86-64\_rpm\_langpack\_zh-CN.tar.gz

2.将下载好的包放到/usr/local下，首先给与目录执行权限

chmod 771

LibreOffice\_5.0.0\_Linux\_x86-64\_rpm.tar.gz LibreOffice\_5.0.0\_Linux\_x86-64\_rpm\_helppack\_zh-CN.tar.gz LibreOffice\_5.0.0\_Linux\_x86-64\_rpm\_langpack\_zh-CN.tar.gz -R

3.依次进入加压出来的目录中的 RPMS 文件夹 执行安装

安装顺序为 1 LibreOffice\_5.0.0.5\_Linux\_x86-64\_rpm 主文件

2 LibreOffice\_5.0.0.5\_Linux\_x86-64\_rpm\_langpack\_zh-CN 语言包

3 LibreOffice\_5.0.0.5\_Linux\_x86-64\_rpm\_helppack\_zh-CN 离线语言包

4.执行安装命令命令 rpm -ivh \*.rpm

****安装语言包xpdf-chinese-simplified.tar.gz****

1. 下载xpdf-chinese-simplified.tar.gz文件

2. 直接解压在/use/local 下即可

tar zxvf xpdf-chinese-simplified.tar.gz

安装Ngix1.2.7(或者更高版本、测试1.2.7版本线上能稳定运行)

1.安装系统依赖包

命令行中输入

sudo -s

LANG=C

yum -y install gcc gcc-c++ autoconf libjpeg libjpeg-devel libpng libpng-devel freetype freetype-devel libxml2 libxml2-devel zlib zlib-devel glibc glibc-devel glib2 glib2-devel bzip2 bzip2-devel ncurses ncurses-devel curl curl-devel e2fsprogs e2fsprogs-devel krb5 krb5-devel libidn libidn-devel openssl openssl-devel openldap openldap-devel nss\_ldap openldap-clients openldap-servers

以上步骤为通过yum命令安装系统依赖包（需要yum源，外网或光盘均可）

2.下载安装文件

mkdir -p /data0/software

cd /data0/software

wget http://blog.s135.com/soft/linux/nginx\_php/nginx/nginx-0.8.46.tar.gz

wget http://blog.s135.com/soft/linux/nginx\_php/php/php-5.3.19.tar.gz

wget http://blog.s135.com/soft/linux/nginx\_php/phpfpm/php-5.2.14-fpm-0.5.14.diff.gz

wget http://blog.s135.com/soft/linux/nginx\_php/libiconv/libiconv-1.13.1.tar.gz

wget http://blog.s135.com/soft/linux/nginx\_php/mcrypt/libmcrypt-2.5.8.tar.gz

wget http://blog.s135.com/soft/linux/nginx\_php/mcrypt/mcrypt-2.6.8.tar.gz

wget http://blog.s135.com/soft/linux/nginx\_php/memcache/memcache-2.2.5.tgz

wget http://blog.s135.com/soft/linux/nginx\_php/mhash/mhash-0.9.9.9.tar.gz

wget http://blog.s135.com/soft/linux/nginx\_php/pcre/pcre-8.10.tar.gz

wget http://blog.s135.com/soft/linux/nginx\_php/eaccelerator/eaccelerator-0.9.6.1.tar.bz2

wget http://blog.s135.com/soft/linux/nginx\_php/imagick/ImageMagick.tar.gz

wget http://blog.s135.com/soft/linux/nginx\_php/imagick/imagick-2.3.0.tgz

3.安装Ngix1.2.7

安装Nginx所需的pcre库

tar zxvf pcre-8.10.tar.gz

cd pcre-8.10/

./configure

make && make install

cd ../

安装Nginx

tar zxvf nginx-0.8.46.tar.gz

cd nginx-0.8.46/

./configure --user=www --group=www --prefix=/usr/local/nginx --with-http\_stub\_status\_module --with-http\_ssl\_module

make && make install

cd ../

/usr/sbin/groupadd www

/usr/sbin/useradd -g www www

创建Nginx日志目录

mkdir -p /data1/logs

chmod +w /data1/logs

chown -R www:www /data1/logs

创建Nginx配置文件

rm -f /usr/local/nginx/conf/nginx.conf

vi /usr/local/nginx/conf/nginx.conf

输入以下内容：

user www www;

worker\_processes 2;

error\_log /data1/logs/nginx\_error.log crit;

pid /usr/local/nginx/nginx.pid;

#Specifies the value for maximum file descriptors that can be opened by this process.

worker\_rlimit\_nofile 65535;

events

{

use epoll;

worker\_connections 65535;

}

http

{

include mime.types;

default\_type application/octet-stream;

#server\_tokens?off;

#charset gb2312;

server\_names\_hash\_bucket\_size 128;

client\_header\_buffer\_size 32k;

large\_client\_header\_buffers 4 32k;

client\_max\_body\_size 8m;

sendfile on;

tcp\_nopush on;

keepalive\_timeout 60;

tcp\_nodelay on;

fastcgi\_connect\_timeout 300;

fastcgi\_send\_timeout 300;

fastcgi\_read\_timeout 300;

fastcgi\_busy\_buffers\_size 128k;

fastcgi\_temp\_file\_write\_size 128k;

gzip on;

gzip\_min\_length 1k;

gzip\_buffers 4 16k;

gzip\_http\_version 1.0;

gzip\_comp\_level 2;

gzip\_types text/plain application/x-javascript text/css application/xml;

gzip\_vary on;

#limit\_zone crawler $binary\_remote\_addr 10m;

server

{

listen 80;

server\_name 192.168.60.50;

index index.html index.htm index.php;

root /data0/htdocs/www;

#limit\_conn crawler 20;

location /

{

#fastcgi\_pass unix:/tmp/php-cgi.sock;

rewrite ^(.\*)/(.+)/([0-9]+)/([0-9]+)/([0-9]+)/([0-9]+)/index\.php$ $1/index.php?ucode=$2&model=$3&f=$4&ct=$5&siteid=$6&$query\_string ;

fastcgi\_pass 127.0.0.1:9000;

fastcgi\_index index.php;

include fcgi.conf;

}

location ~ .\*\.(gif|jpg|jpeg|png|bmp|swf)$

{

expires 30d;

}

location ~ .\*\.(js|css)?$

{

expires 1h;

}

log\_format access '$remote\_addr - $remote\_user [$time\_local] "$request" '

'$status $body\_bytes\_sent "$http\_referer" '

'"$http\_user\_agent" $http\_x\_forwarded\_for';

access\_log /data1/logs/access.log access;

}

}

在/usr/local/nginx/conf/目录中创建fcgi.conf文件：

vi /usr/local/nginx/conf/fcgi.conf

输入

fastcgi\_param GATEWAY\_INTERFACE CGI/1.1;

fastcgi\_param SERVER\_SOFTWARE nginx;

fastcgi\_param QUERY\_STRING $query\_string;

fastcgi\_param REQUEST\_METHOD $request\_method;

fastcgi\_param CONTENT\_TYPE $content\_type;

fastcgi\_param CONTENT\_LENGTH $content\_length;

fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;

fastcgi\_param SCRIPT\_NAME $fastcgi\_script\_name;

fastcgi\_param REQUEST\_URI $request\_uri;

fastcgi\_param DOCUMENT\_URI $document\_uri;

fastcgi\_param DOCUMENT\_ROOT $document\_root;

fastcgi\_param SERVER\_PROTOCOL $server\_protocol;

fastcgi\_param REMOTE\_ADDR $remote\_addr;

fastcgi\_param REMOTE\_PORT $remote\_port;

fastcgi\_param SERVER\_ADDR $server\_addr;

fastcgi\_param SERVER\_PORT $server\_port;

fastcgi\_param SERVER\_NAME $server\_name;

# PHP only, required if PHP was built with --enable-force-cgi-redirect

fastcgi\_param REDIRECT\_STATUS 200;

启动Nginx

ulimit -SHn 65535

/usr/local/nginx/sbin/nginx

**备注: 隐藏nginx的版本号：**/　**中：**

http {  
......省略配置  
sendfile on;  
tcp\_nopush on;  
keepalive\_timeout 65;  
tcp\_nodelay on;  
**server\_tokens off; ##隐藏nginx的版本号（注释掉）**  
.......省略配置  
}

配置开机自启动

vi /etc/rc.local

在末尾增加以下内容：

ulimit -SHn 65535

/usr/local/php/sbin/php-fpm start

/usr/local/nginx/sbin/nginx

测试ngix.php页面

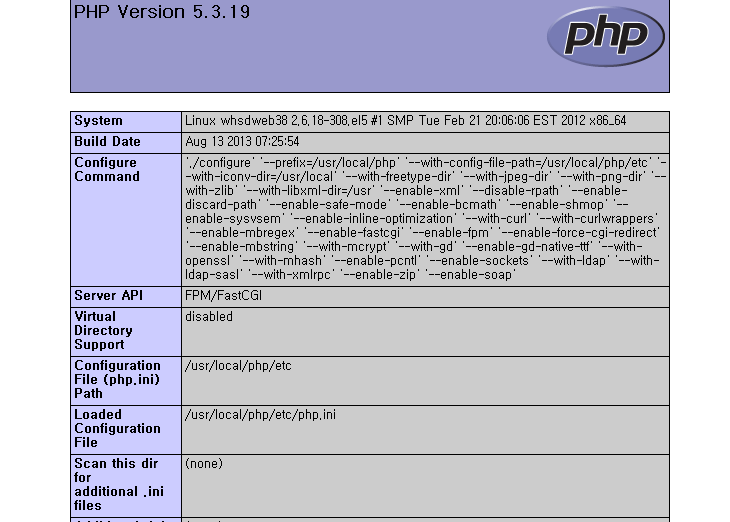
Vi /data0/htdocs/www/1.php

<?php

phpinfo()

?>

IE 输入：<http://localhost/1.php> 代表nginx php 环境搭建完成



****安装PHP5.3.19版本（注意:版本不能不能小于5.2）****

1.编译安装php所需支持库

tar zxvf libiconv-1.13.1.tar.gz

cd libiconv-1.13.1/

./configure --prefix=/usr/local

make

make install

cd ../

tar zxvf libmcrypt-2.5.8.tar.gz

cd libmcrypt-2.5.8/

./configure

make

make install

/sbin/ldconfig

cd libltdl/

./configure --enable-ltdl-install

make

make install

cd ../../

tar zxvf mhash-0.9.9.9.tar.gz

cd mhash-0.9.9.9/

./configure

make

make install

cd ../

ln -s /usr/local/lib/libmcrypt.la /usr/lib/libmcrypt.la

ln -s /usr/local/lib/libmcrypt.so /usr/lib/libmcrypt.so

ln -s /usr/local/lib/libmcrypt.so.4 /usr/lib/libmcrypt.so.4

ln -s /usr/local/lib/libmcrypt.so.4.4.8 /usr/lib/libmcrypt.so.4.4.8

ln -s /usr/local/lib/libmhash.a /usr/lib/libmhash.a

ln -s /usr/local/lib/libmhash.la /usr/lib/libmhash.la

ln -s /usr/local/lib/libmhash.so /usr/lib/libmhash.so

ln -s /usr/local/lib/libmhash.so.2 /usr/lib/libmhash.so.2

ln -s /usr/local/lib/libmhash.so.2.0.1 /usr/lib/libmhash.so.2.0.1

ln -s /usr/local/bin/libmcrypt-config /usr/bin/libmcrypt-config

tar zxvf mcrypt-2.6.8.tar.gz

cd mcrypt-2.6.8/

/sbin/ldconfig

./configure

make

make install

cd ../

2.编译安装php5.3.19

\*\*cp -frp /usr/lib64/libldap\* /usr/lib/ (6.3以上版本要执行否则报错)\*\*\*

cp -frp /usr/lib64/libldap\* /usr/lib/

tar zxvf php-5.3.19.tar.gz

cd php-5.3.19/

./configure --prefix=/usr/local/php --with-config-file-path=/usr/local/php/etc --with-iconv-dir=/usr/local --with-freetype-dir --with-jpeg-dir --with-png-dir --with-zlib --with-libxml-dir=/usr --enable-xml --disable-rpath --enable-discard-path --enable-safe-mode --enable-bcmath --enable-shmop --enable-sysvsem --enable-inline-optimization --with-curl --with-curlwrappers --enable-mbregex --enable-fastcgi --enable-fpm --enable-force-cgi-redirect --enable-mbstring --with-mcrypt --with-gd --enable-gd-native-ttf --with-openssl --with-mhash --enable-pcntl --enable-sockets --with-ldap --with-ldap-sasl --with-xmlrpc --enable-zip --enable-soap --with-mysql=mysqlnd --with-mysqli=mysqlnd --with-pdo-mysql=mysqlnd

#红色部分是安装mysql扩展插件，如果没有安装，后期可以另外安装，见

《php 安装mysql 扩展包方式.txt》

make ZEND\_EXTRA\_LIBS='-liconv'

make install

cp php.ini-development /usr/local/php/etc/php.ini

cd ../

3.安装php扩展

创建www用户和组

/usr/sbin/groupadd www

/usr/sbin/useradd -g www www

mkdir -p /data0/htdocs/www

chmod +w /data0/htdocs/www

chown -R www:www /data0/htdocs/www

可以不安装，根据需求：

**安装php扩展**

**tar zxvf memcache-2.2.5.tgz**

**cd memcache-2.2.5/**

**/usr/local/php/bin/phpize**

**./configure --with-php-config=/usr/local/php/bin/php-config**

**make**

**make install**

**cd ../**

**tar jxvf eaccelerator-0.9.6.1.tar.bz2**

**cd eaccelerator-0.9.6.1/**

**/usr/local/php/bin/phpize**

**./configure --enable-eaccelerator=shared --with-php-config=/usr/local/php/bin/php-config**

**make**

**make install**

**cd ../**

**tar zxvf ImageMagick.tar.gz**

**cd ImageMagick-6.5.1-2/**

**./configure**

**make**

**make install**

**cd ../**

**tar zxvf imagick-2.3.0.tgz**

**cd imagick-2.3.0/**

**/usr/local/php/bin/phpize**

**./configure --with-php-config=/usr/local/php/bin/php-config**

**make**

**make install**

**cd ../**

修改php.ini文件

查找/usr/local/php/etc/php.ini中的extension\_dir = "./"修改为extension\_dir = "/usr/local/php/lib/php/extensions/no-debug-non-zts-20090626/"

并在此行后增加以下几行，然后保存：

首先要安装插件

extension = "memcache.so"

　　extension = "imagick.so"

如果有报错：

　　＃extension = "memcache.so"

　　＃extension = "imagick.so"

　　再查找output\_buffering = Off

　　修改为output\_buffering = On

再查找; cgi.fix\_pathinfo=0 改为0

配置eAccelerator加速php

创建php-fpm配置文件

cat > /usr/local/php/etc/php-fpm.conf << EOF

[global]

[www]

user = www

group = www

listen = 127.0.0.1:9000

pm = dynamic

pm.max\_children = 200

pm.start\_servers = 2

pm.min\_spare\_servers = 1

pm.max\_spare\_servers = 3

slowlog = log/$pool.log.slow

rlimit\_files = 65535

rlimit\_core = 0

EOF

echo '/usr/local/php/sbin/php-fpm' >> /etc/rc.local

启动php进程，监听127.0.0.1的9000端口

ulimit -SHn 65535

参考：http://www.cnblogs.com/mchina/archive/2012/05/17/2507102.html