www-proxy.exu.ericsson.se 8080

www-proxy.ao.ericsson.se 8080

部署：

0099azaz.ddns.ms

yanzhengma.f3322.net

1号机器

内网IP:192.168.0.101

113.3.196.96

ssh端口:10001

VNC端口:5901

使用端口:10010-10020

2号机器

内网IP:192.168.0.102

ssh端口:10002

VNC端口:5902

使用端口:20010-20020

3号机器

内网IP:192.168.0.103

ssh端口:10003

VNC端口:5903

使用端口:30010-30020

4号机器：

内网IP:192.168.3.104

ssh端口:10004

使用端口:40010-40020

5号机器

内网IP:192.168.3.105

ssh端口:10005

VNC端口:5905

使用端口:50010-50020

6号机器：

内网IP:192.168.3.106

ssh端口:10006

VNC端口:5906

使用端口:60010-60020

内网IP:192.168.3.107

7号机器

内网IP:192.168.3.107

ssh端口:10007

使用端口:60050-60060

8号机器

内网IP:192.168.3.108

ssh端口:10008

使用端口:60070-60080

#9号机器

内网IP:192.168.3.109

ssh端口:10009

使用端口:60090-60100

10号机器

内网IP:192.168.3.110

ssh端口:11010

使用端口:60100-60110

server:

###captcha7890&\*()

7e727d90-7915-11e4-89be-e41f13653bfc

充值：

221.206.124.137

DNS:

/etc/resolv.conf

https://ssl.captcha.qq.com/getimage?aid=522205405&r=0.616197673836723&uin=3231232321@qq.com

每天统计调用次数：

show events;

SET GLOBAL event\_scheduler = 1;

CREATE EVENT called\_details

ON schedule EVERY 1 DAY starts timestamp '2014-10-21 00:00:00'

DO INSERT INTO detailcallrecord (USERID,leftCount,calledCount,SUCCESSCOUNT,DATE)

select \* from ((select ID,leftCount,calledCount,SUCCESSCOUNT from test.userinfo) a, (select CURRENT\_TIMESTAMP) b);

ALTER EVENT called\_details DISABLE;

ALTER EVENT called\_details ENABLE;

drop event called\_details;

select \* from detailcallrecord;

mysql -u root -p test

启动：用root登陆后依次执行：SET GLOBAL event\_scheduler = 1;

                                             ALTER EVENT called\_details ENABLE;

停止：用root登陆后依次执行：ALTER EVENT called\_details DISABLE;

                                             SET GLOBAL event\_scheduler = 0;

查看DNS：

vi /etc/resolv.conf

查看CPU：

cat /proc/cpuinfo

查看操作系统位数：

getconf LONG\_BIT

ping -c1 0099azaz.ddns.ms|awk -F'[(|)]' 'NR==1{print $2}'

issue:

现在有两个地方可能有问题，1是支付宝的解析页面，在web/alipayParser.py，这个写的比较简单，也没好好测

2.是pyobfuscate这个代码混淆工具不能与web.py很好的配合呢

code->rest-thread->restServer-threading-cuda.py

1、查看Linux系统版本：cat /proc/version

   查看逻辑CPU个数：cat /proc/cpuinfo |grep "processor"|wc -l

2、

lihttpd启动的话是/etc/init.d/lighttpd start

配置文件在/etc/lighttpd/lighttpd.conf

          /etc/lighttpd/fastcgi.conf

一个是配基本功能的，一个是配fastcgi的

lighttpd使用的py文件在/www目录下

要注意的是lighttpd不能以root执行，我新建了个lighttpd用户

启动日志：/www/errorlog

装flup：pip install flup

3、

Nginx:

bind:

/usr/local/nginx/sbin/nginx -c /usr/local/nginx/conf/nginx.conf

启动：

/usr/local/nginx/sbin/nginx start

关闭：

/usr/local/nginx/sbin/nginx stop

重启：

/usr/local/nginx/sbin/nginx –s reload

uwsgi --ini conf-1.ini

uwsgi --emperor . --daemonize = 127.0.0.1:4001

uwsgi --emperor . --daemonize = 127.0.0.1:5001

查看流量：

iftop -i eth0

ps -ef | grep wsgi | grep -v grep | cut -c 9-15 | xargs kill -s 9

4、

Python fastcgi:

python fcgi.py --method=prefork/threaded minspare=50 maxspare=50 maxchildren=1000

spawn-fcgi -f /usr/local/nginx/fcgi.py -a 127.0.0.1 -p 8007 -u www -F 4

-f WSGIServer脚本的决定路径

-a fastcgi绑定的ip

-p fastcgi绑定的端口

-u 已什么身份运行

-F 要启动的进程数（-C是PHP专用的，这里要用-F）

5、

安装pypy（顺序不能错）：

rpm -ivh http://dl.fedoraproject.org/pub/epel/6/x86\_64/pypy-libs-2.0.2-1.el6.x86\_64.rpm

rpm -ivh http://dl.fedoraproject.org/pub/epel/6/x86\_64/pypy-2.0.2-1.el6.x86\_64.rpm

rpm -ivh http://dl.fedoraproject.org/pub/epel/6/x86\_64/pypy-devel-2.0.2-1.el6.x86\_64.rpm

6、

tornado测试：

service lighttpd stop

共1000个请求，分100个线程处理（apache benchmark命令）：

ab -n 1000 -c 100 http://127.0.0.1:8088/sleep

ab -n 1000 -c 100 http://127.0.0.1:8081/index

ab -n 10000 -c 100 http://127.0.0.1:8081/do/

传图片：

ab -n 10000 -c 100 -p myfile.jpg http://1.56.180.195:8081/do

Server端：

length=int(environ.get('CONTENT\_LENGTH','0'))

print environ['wsgi.input'].read(length)

python多进程:

captchaWeb\captcha\src\client\dll\1.56.180.195

Performance.py

VNC启动web：

221.206.124.211:5901

密码captcha123!@#

然后：python webserver.py

mysql -h localhost -u test -p test

查看mysql连接数：mysqladmin -uroot -p  processlist

日志统计（统计访问量最高的100个IP及访问次数）：

awk '{print $1}' access.log | sort | uniq -c | sort -n -k 1 -r | head -n 100

定时任务：

vi /var/spool/cron/root

0 0 \* \* \* /usr/local/nginx/clear\_log.sh

0 0 \* \* \* /usr/local/nginx/dellog.sh

\*/1 \* \* \* \* /usr/local/nginx/detectIp.sh

定时任务启动与关闭：

cron stop

cron start

阿里云：

114.215.109.141

root

tXl482382

看配置：cat /proc/cpuinfo

新阿里云：

182.92.228.120

root

密码Qq666566

http代理：

http://www.daili999.com/?a=getip

订单号：800200qq

数量：随便（1000）

http://221.206.124.11

qqaazz112233!

http://182.92.228.120:8080

http://182.92.228.120:8080/static/doc/Dll.zip

QQ:

1、代理（总数在10000个左右，可以每次获取几千个，放到一个队列里，用完再换一批）：

http://daili999.com/api.php?key=800200qq&getnum=10

2、

javah -jni com.captcha.CaptchaExample

1、服务器

221.206.124.211

root

wwssxx885522

113.3.192.237

(1.56.180.195)

113.3.197.170

113.3.193.100

113.7.117.56

113.7.118.247

0099azaz.ddns.ms:22:root/666566

已开的端口：8081-8085

测试机器：

221.206.124.211 root qqaazz778899

vnc密码 captcha123!@#

import theano

print theano.config.device

gpu

Using gpu device 0: GeForce GTX 285

echo $LANG，那个centos默认为中文utf-8编码，所以为了不乱码，xshell要设置编码为utf-8，http://wenku.baidu.com/view/80ed41670b1c59eef8c7b4c0.html

useradd zhangda，创建zhangda用户，passwd zhangda，为zhangda用户设置密码

查看系统详细版本  rpm -q centos-release

解压tar.xz文件，需要yum install xz，把xz工具装上

$xz -d \*\*\*.tar.xz

$tar -xvf  \*\*\*.tar

opencv old api

img = cv.LoadImage(Constants.IMG\_DIR\_TENCENT\_TEST + "0.jpg", 0)

    print(img)

    arr = numpy.asarray(img[:,:])

    #data = img.imageData

    print(type(arr))

    print(arr.shape)

    print(arr[1:50, 1:50])

0.基本安装包

yum install gcc gcc-gfortran gcc-c++ blas-devel lapack-devel atlas-devel automake autoconf libtool make zlib-devel  openssl\* bzip2\*

1.vncserver安装和配置

参考http://www.unixmen.com/install-vnc/

yum groupinstall  "General Purpose Desktop" "Desktop Platform"

yum -y install vnc-server

yum install tigervnc-server

设置密码

vncpasswd

修改配置文件

vi /etc/sysconfig/vncservers

VNCSERVERS="1:root"

VNCSERVERARGS[1]="-geometry 1280x1024"

chkconfig vncserver on

service vncserver start

暂时关掉防火墙

service iptables  stop

下面的不要参考了

yum groupinstall "X Window System" "Desktop"（CentOS 6.x安装GNOME桌面环境）

yum install tigervnc

yum install tigervnc-server

之后按这个上面讲的配置，一定要每步都做：http://jingyan.baidu.com/article/da1091fbde71ea027849d692.html

http://www.ha97.com/4634.html  这两个网页参考一下就行了，如果出现10060（Timed out），说明没配置好

2.安装或更新python

默认的是pyton 2.6.6

linux貌似自带的python是根据系统来的，系统是64位，那么python就是64位

或者可以

import platform

platform.architecture()

('64bit', 'ELF')

这样来判断，不知是否准备。

看到一个说法，centos很多程序不支持python 2.7，所以暂时不更新了。

看到说Python 2.7生成的pyc不能再python 2.6上运行，所以还是更新。

wget http://python.org/ftp/python/2.7.3/Python-2.7.3.tar.bz2

tar -jxvf Python-2.7.3.tar.bz2

cd Python-2.7.3

vi Modules/Setup.dist

#SSL=/usr/local/ssl

#\_ssl \_ssl.c \

#       -DUSE\_SSL -I$(SSL)/include -I$(SSL)/include/openssl \

#       -L$(SSL)/lib -lssl -lcrypto

......

#zlib zlibmodule.c -I$(prefix)/include -L$(exec\_prefix)/lib -lz

把zlib那一行的注释去掉

./configure --prefix=/usr/local/python2.7 -enable-shared CFLAGS=-fPIC

make all

make install

make clean

make distclean

mv /usr/bin/python /usr/bin/python2.6.6

ln -s /usr/local/python2.7/bin/python /usr/bin/python

修正yum无法使用的问题

vi /usr/bin/yum

将文件头部的

#!/usr/bin/python

改成

#!/usr/bin/python2.6.6

之后pyhon可能无法启动，提示ImportError: libpython2.7.so.1.0: cannot open shared object file: No such file or directory

这样：

2. cp /usr/local/python2.7/lib/libpython2.7.so.1.0 /usr/local/lib

     cd /usr/local/lib

     ln -s libpython2.7.so.1.0 libpython2.7.so

  3. 使用命令whereis libpython2.7.so.1.0得到如下结果就说明

    libpython2.7.so.1: /usr/local/lib/libpython2.7.so.1.0

  4. 如果whereis没有结果，或者还有import错误，可以尝试如下操作：

    在/etc/ld.so.conf中加入新行/usr/local/lib

    保存后，运行

    /sbin/ldconfig

    /sbin/ldconfig -v

2.1 安装easy\_install, pip

先安装setup-tools

wget https://pypi.python.org/packages/2.7/s/setuptools/setuptools-0.6c11-py2.7.egg  --no-check-certificate

chmod +x setuptools-0.6c11-py2.7.egg

sh setuptools-0.6c11-py2.7.egg

安装pip

wget https://pypi.python.org/packages/source/p/pip/pip-1.3.1.tar.gz --no-check-certificate

cp pip-1.3.1.tar.gz /usr/src/

tar zxvf pip-1.3.1.tar.gz

cd pip-1.3.1

python setup.py install

ln -s /usr/local/python2.7/bin/pip /usr/bin/pip

3.安装theano

Installation steps

    1. sudo yum install python-devel python-nose python-setuptools gcc gcc-gfortran gcc-c++ blas-devel lapack-devel atlas-devel

    2. sudo easy\_install pip

    3. sudo pip install numpy==1.6.1

    4. sudo pip install scipy==0.10.1

    5. sudo pip install Theano

Test the newly installed packages

    1. NumPy (~30s): python -c "import numpy; numpy.test()"

    2. SciPy (~1m): python -c "import scipy; scipy.test()"

    3. Theano (~30m): python -c "import theano; theano.test()"

执行下面的可以更新theano和其依赖项

pip install --upgrade theano

验证numpy是否真的成功依赖BLAS编译，用以下代码试验：

>>> import numpy

>>> id(numpy.dot) == id(numpy.core.multiarray.dot)

False

4. 安装opencv

更新：不用opencv了，用PIL即可。

yum install opencv\*

这里安装的opencv只是针对python 2.6的，之后还要

cd /usr/lib64/python2.6/site-packages/

cp cv.so /usr/local/python2.7/lib/python2.7/site-packages/

cp -r opencv/ /usr/local/python2.7/lib/python2.7/site-packages/

这个是可以用的，不过是老版本的opencv，有些api要变

老版API参考

http://stackoverflow.com/questions/3666779/python-open-cv-loading-image

http://www.iteye.com/topic/463668

很麻烦，不一定装的起来了

yum install cmake python-devel numpy gcc gcc-c++ gtk2-devel libdc1394-devel libv4l-devel ffmpeg-devel gstreamer-plugins-base-devel

可选的依赖

yum install libpng-devel libjpeg-turbo-devel jasper-devel openexr-devel libtiff-devel libwebp-devel tbb-devel eigen3-devel texlive

配置cmake

cmake -D CMAKE\_BUILD\_TYPE=RELEASE -D CMAKE\_INSTALL\_PREFIX=/usr/local  -D WITH\_TBB=ON -D BUILD\_TESTS=OFF -D BUILD\_PERF\_TESTS=OFF -D BUILD\_EXAMPLES=OFF  -D WITH\_OPENCL=OFF -D WITH\_CUDA=OFF -D BUILD\_opencv\_gpu=OFF -D BUILD\_opencv\_gpuarithm=OFF -D BUILD\_opencv\_gpubgsegm=OFF -D BUILD\_opencv\_gpucodec=OFF -D BUILD\_opencv\_gpufeatures2d=OFF -D BUILD\_opencv\_gpufilters=OFF -D BUILD\_opencv\_gpuimgproc=OFF -D BUILD\_opencv\_gpulegacy=OFF -D BUILD\_opencv\_gpuoptflow=OFF -D BUILD\_opencv\_gpustereo=OFF -D BUILD\_opencv\_gpuwarping=OFF

默认的cmake可能版本太低，需要这样：

wget http://files1.directadmin.com/services/custombuild/cmake-2.8.11.tar.gz

tar xzf cmake-2.8.11.tar.gz

cd cmake-2.8.11

./configure

make

make install

安装好后修改环境变量

vi /etc/profile

在文件末尾追加以下两行代码：

PATH=/usr/local/bin:$PATH

export PATH

source /etc/profile 就好了

5.安装python-dmidecode

最好跑一下这个   yum groupinstall "Development Tools"

这个是检测系统硬件信息的，很简单yum install python-dmidecode 即可

python封装版的实在不行的话就yum install dmidecode，然后再python里call linux命令吧，一样。

6.安装mysql

yum install -y mysql-server mysql mysql-devel

之后rpm -qi mysql-server可以查看安装的mysql版本

service mysqld start  启动mysql

chkconfig mysqld on    添加到开机自启动

/usr/bin/mysqladmin -u root password 'new-password'     为root用户设置密码

mysql -u root -p    可以登录mysql了

此时root用户应该不能远程登录mysql，这样：

mysql -u root -p

use mysql

update user set host = '%' where user = 'root';

flush privileges;

这样应该就可以了，不过最好是用其他用户，限定一些权限，不要用root。root就限定本地访问好了。

新建用户：

mysql -u root -p

insert into mysql.user(Host,User,Password) values('localhost','captcha',password('1234'));

flush privileges;

create database captcha;

grant all privileges on captcha.\* to captcha@localhost identified by '1234';

flush privileges;

6.1 安装mysqld

yum安装，可能不行

yum install MySQL-python

源码安装

yum install mysql-devel\* -y

wget --no-check-certificate https://pypi.python.org/packages/source/M/MySQL-python/MySQL-python-1.2.3.tar.gz

tar xzvf MySQL-python-1.2.3.tar.gz

cd MySQL-python-1.2.3

python setup.py build

python setup.py install

如果import MySQLdb能成功就行了

7.代码混淆

用pyobfuscate做代码混淆，之后只发布pyc文件就行。源码在此：https://github.com/astrand/pyobfuscate

直接pyobfuscate t.py > out.py就行了。参数啥的可以用默认，windows下也是可以的。

web.py  很简单，下载安装包，直接python setup.py install，如果出现编码问题，就把error信息拷贝下来，网上有解决方法，一般是python27/Lib/mimetypes.py出了问题，可能装mysqldb时给改了

MySqldb   python 操作mysql必装了，linux上没有任何问题，windows上32位python很好安装，有exe。64位的要先安装MINGW 64bit，如果版本不对，会报错。所以windows上就先用32位的跑。

8.lighttpd配置

这里主要还是权限的问题，因为lighttpd不能以root运行，所以要新建一个账户，然而新建的账户很有可能没有python的执行权限，而我的实验的机器python版本极其混乱，所以走了很多弯路。

基本的步骤

1.安装包

web.py

flup   pip install flup

有个正则的东西要装 pcre 想着yum install pcre-devel就可以

lighttpd 源码安装

./configure即可，不用加其他的config

make

make install

之后把/doc/initscripts/rc.lighttpd.redhat copy 到 /etc/init.d/lighttpd

在/etc/lighttpd/下建配置文件，如果没有此文件夹，就建立

两个配置文件：

1.lighttpd.conf

server.document-root = "/www/"

server.port = 80

server.username = "lighttpd"

server.groupname = "lighttpd"

mimetype.assign = (

  ".html" => "text/html",

  ".txt" => "text/plain",

  ".jpg" => "image/jpeg",

  ".png" => "image/png"

)

static-file.exclude-extensions = (".py", ".fcgi", ".php", ".rb", "~", ".inc" )

# index-file.names = ( "index.html" )

server.modules              = (

            "mod\_access",

            "mod\_alias",

            "mod\_accesslog",

            "mod\_compress",

)

server.errorlog = "/www/error.log"

accesslog.filename = "/www/access.log"

include "fastcgi.conf"

2.fastcgi.conf

server.modules += ( "mod\_fastcgi" )

server.modules  += ( "mod\_rewrite" )

fastcgi.debug = 1

fastcgi.server = ( "/2.py" =>

(( "socket" => "/tmp/fastcgi.socket",

    "bin-path" => "/root/Enthought/Canopy\_64bit/User/bin/python /www/2.py",

    "max-procs" => 1,

   "bin-environment" => (

     "REAL\_SCRIPT\_NAME" => ""

   ),

   "check-local" => "disable"

))

)

url.rewrite-once = (

   "^/favicon.ico$" => "/static/favicon.ico",

   "^/static/(.\*)$" => "/static/$1",

   "^/(.\*)$" => "/2.py/$1"

)

之后在/www/目录下建立2.py，内容如下：

#!/usr/bin/env python

import web

urls = (

    '/py/test', 'index'

)

class index:

    def GET(self):

        return "Hello, world!"

if \_\_name\_\_ == "\_\_main\_\_":

    app = web.application(urls, globals())

    app.run()

之后/etc/init.d/lighttpd start就可以了

出现问题后先检查权限，一定要切换到lighttpd用户，执行命令bin-path看是不是真正的能成功，很多时候其实是不行的。其实就是因为我的python环境太乱了。

出现的问题:

1.yum 出现 database disk image is malformed

console输入: yum clean dbcache

2.zlib can not decompress

就是修改配置文件再重新编译python吧，好像也没什么别的办法，2.7.7的可能不太好用，用2.7.3的也行

3.dmidecode不好用

有时是缺少libxml2，ftp://xmlsoft.org/libxml2/python/ 从这里下载，解压，python setup.py install就行了

如果yum 安装的dmidecode不好用，就git clone git://git.fedorahosted.org/python-dmidecode.git， 然后make all, make install应该也能装上，但装上归装上，实际调用时还是产生段错误，没用。

4.python编译为pyc

import py\_compile

py\_compile.compile(r'1.py')

5.Anaconda linux -x64 安装包

http://09c8d0b2229f813c1b93-c95ac804525aac4b6dba79b00b39d1d3.r79.cf1.rackcdn.com/Anaconda-2.0.1-Linux-x86\_64.sh

Anaconda有个很bug的地方，就是所有的so文件都必须放在他自己的lib目录下，默认安装时就是/root/anaconda/lib下面，所以安装完后要把/root/lib64/atlas目录下的所有atlas.a和.so文件都拷贝过来，就不会提示找不到cannot find -lf77blas了。

按理说这种找不到so时一般解决方法就是安装上面的python，编辑/etc/ld.so.conf中加入新行“/usr/local/lib”，这个是新的lib路径，保存后，运行/sbin/ldconfig，这样会更新缓存，/sbin/ldconfig -v可以查看目前的缓存

6.python 生成pyo

python -O -m py\_compile performanceTest.py

-m指import相应的module了

7./root/Canopy/canopy  开启canopy的图形界面

Canopy 的安装包是放在/root/Enthought/Canopy\_64bit/User/lib/python2.7/site-packages

8.python profile分析程序性能

python -m profile test.py

9.web.py 通过python setup.py install安装

selenium 是这样：  pip install selenium

10.theano  lighttpd

没有别的问题，还是权限问题，不知为何，theano默认会建立一个/root/.theano/的文件夹，所以以lighttpd用户绝对是没权限的了，简单暴力的解决方法就是chown -R lighttpd.lighttpd /root, 这样就行了，其他问题不大了。

要做的事：

1.rest server上增加传文件的功能

3.增加一个每天每个用户调用次数的table，并且rest server中要增加更新操作

4.将code都整合到一个文件里，写一个自动生成检测硬件信息的constants的程序，这样通过读配置文件的方式，constants就不要卸载python里面了。

5.整合测试，优化配置

7.支付宝交易号目前只检测第一页的，以后再加分页检查的

9.再做一个管理员页面

已做完：

1.在后台管理页面显示用户名

2.交易记录显示倒序

3.测试支付宝，改好充值页面的sql

6.加一个统一的log

8.重复的订单号就直接不能提交就行

\*/1 \* \* \* \* /usr/local/nginx/detectIp.sh

detectIp.sh

#!/bin/bash

IP=`ping -c1 0099azaz.ddns.ms|awk -F'[(|)]' 'NR==1{print $2}'`

echo $IP

OLDIP=`tail /usr/local/nginx/changedips.log -n -1`

echo $OLDIP

if [ $IP = $OLDIP ]; then

echo eq

else

echo not eq

echo $IP >> /usr/local/nginx/changedips.log

`/usr/local/nginx/sbin/nginx -s reload`

fi

221.206.124.137:51888

张达 2014/9/25 20:35:17

你登这个吧

张达 2014/9/25 20:35:22

idc01

张达 2014/9/25 20:35:27

qaz123!@#是密码

1001  作者用户名或密码错误，或作者token传递错误

1002  用户需要充值

1003  客户端参数传递错误

1004  服务器其他错误

1005  DLL客户端错误（例如：网络连接超时）

1006  当前子用户已存在

1007  子用户不存在

1008  子用户密码错误

1012  服务器内部错误，500 error

linux 查看内存：free（单位KB）

1.启动Memcache的服务器端：

/usr/local/bin/memcached -d -m 100 -u root -l 221.206.125.7 -p 12000 -c 7200 -P /tmp/memcached.pid

/usr/local/bin/memcached -d -m 300 -u root -l 221.206.125.7 -p 12000 -c 10000 -P /tmp/memcached.pid

-d选项是启动一个守护进程，

-m是分配给Memcache使用的内存数量，单位是MB，我这里是10MB，

-u是运行Memcache的用户，我这里是root，

-l是监听的服务器IP地址，如果有多个地址的话，我这里指定了服务器的IP地址192.168.0.200，

-p是设置Memcache监听的端口，我这里设置了12000，最好是1024以上的端口，

-c选项是最大运行的并发连接数，默认是1024，我这里设置了256，按照你服务器的负载量来设定，

-P是设置保存Memcache的pid文件，我这里是保存在 /tmp/memcached.pid，

2.如果要结束Memcache进程，执行：

# kill `cat /tmp/memcached.pid`

3、安装python API组件：

   下载python-memcached-1.39.tar.gz

   tar xvzf python-memcached-1.39.tar.gz

   python setup.py install

telnet 221.206.125.7 12000

stats

import memcache, time

mc = memcache.Client(['221.206.124.211:12000'], debug=0)

mc.set("some\_key", "Some value")

time.sleep(2)

value = mc.get("some\_key")

print value

host='221.206.124.11'

user='captcha'

passwd='1234'

db='test'

port=3306

memServer='221.206.124.11:12000'

http://www.kqdama.com:8009/basic\_status

新首页：

http://www.kqdama.com/

221.206.125.7

captcha12345!@#$%

管理界面：

http://www.kqdama.com:9000/login

用户名admin,密码7890&\*()

查看状态：

http://www.kqdama.com:8009/basic\_status

http://www.kqdama.com/status.html

新开screen： screen

查看screen: screen -ls

attach到screen: screen -r 13599

在screen里执行：python sync.py &

断开screen：screen -d 13599

set http\_proxy=www-proxy.ao.ericsson.se:8080

show variables like '%sche%';

select user\_name,count(\*) as count from user\_table group by user\_name having count>1;

select leftcount from test.detailcallrecord where date='2014-11-4 00:00:00' and userid=169;

1560712

select sum(leftcount),sum(successcount) from test.detailcallrecord where date='2014-11-4 00:00:00' and userid in (select id from test.userinfo where parentid=169);

'22262746'

select sum(successcount) from test.detailcallrecord where date='2014-11-4 00:00:00' and userid in (select id from test.userinfo where parentid=169);

'48197708'

total 72021166

select leftcount from test.detailcallrecord where date='2014-11-5 00:00:00' and userid=169;

2030875

select sum(leftcount) from test.detailcallrecord where date='2014-11-5 00:00:00' and userid in (select id from test.userinfo where parentid=169);

'18187145'

select sum(successcount) from test.detailcallrecord where date='2014-11-5 00:00:00' and userid in (select id from test.userinfo where parentid=169);

'49703309'

total  69921329

select successcount/calledcount as result from userinfo;

阿里云：（29916）

root@iZ28ulq54teZ:~/code# export JAVA\_HOME=/usr/java/jre1.8.0\_25

root@iZ28ulq54teZ:~/code# export PATH=$JAVA\_HOME/bin:$PATH

root@iZ28ulq54teZ:~/code# export CLASSPATH=$JAVA\_HOME/jre/lib/rt.jar:.

CREATE EVENT called\_details

ON schedule EVERY 1 HOUR starts timestamp '2014-12-01 17:00:00'

DO INSERT INTO detailcallrecord (USERID,leftCount,calledCount,SUCCESSCOUNT,DATE)

select \* from ((select ID,leftCount,calledCount,SUCCESSCOUNT from test.userinfo) a, (select CURRENT\_TIMESTAMP) b);

http://www.kqdama.com:7000/

admin

123qwe!@#

http://www.kqdama.com:8080/regbyadmin

密钥密码：

qscwdv!#%

新版输出：

1、

uwsgi --udp 127.0.0.1:9100

这是新版check的输出

2、

uwsgi --udp 127.0.0.1:9193

这是新版mdb的输出

mysqladmin -u root -p[oldpass] password newpass

mysql> select sum(calledcount),sum(leftcount),sum(successcount) from userinfo;

+------------------+----------------+-------------------+

| sum(calledcount) | sum(leftcount) | sum(successcount) |

+------------------+----------------+-------------------+

|        748538369 |       75106728 |         353202043 |

+------------------+----------------+-------------------+

1 row in set (0.00 sec)

4:40

select \* from detailcallrecord where date='2014-12-04 09:00:00' and userid not in (select userid from detailcallrecord where date='2014-12-03 22:00:00');

alter table detailcallrecord add column RATIO float(6,3);

alter table detailcallrecord drop column RATIO;

时间取整：

SELECT  now() - interval (TIME\_TO\_SEC(now()) mod 3600) second from dual

select CURRENT\_TIMESTAMP-INTERVAL 1 HOUR;

CREATE EVENT called\_details

ON schedule EVERY 1 HOUR starts timestamp '2014-12-06 16:00:00'

DO INSERT INTO detailcallrecord (USERID,leftCount,calledCount,SUCCESSCOUNT,DATE)

select \* from ((select ID,leftCount,calledCount,SUCCESSCOUNT from test.userinfo) a, (select CURRENT\_TIMESTAMP) b);

CREATE EVENT called\_dynamicratio

ON schedule EVERY 1 HOUR starts timestamp '2014-12-06 16:00:00'

DO INSERT INTO dynamicratio (USERID,RATIO,DATE)

select \* from (select uid,(suc2-suc1)/(called2-called1),time from (select a.calledCount called1,a.SUCCESSCOUNT suc1, b.id uid,b.calledCount called2,b.SUCCESSCOUNT suc2 from detailcallrecord a,userinfo b where a.DATE=(select CURRENT\_TIMESTAMP-INTERVAL 1 HOUR) and b.id=a.userid) tmp, (select CURRENT\_TIMESTAMP time) c) tmp2;

10元=5千码

50元=3万码

100元=7万码

500元=50万码

1000元=150万码

冲的越多优惠越多。日消费1000万码以上联系我谈价。

我们的平台是新的模式，不是分成模式。是把码卖给你，你自己经营（你自己卖给自己的用户，卖10元/100个是自己的本事）

只打包不压缩：

tar -cvf code.tar code/

压缩：

tar -zcvf code.tar.gz code/

tar -jcvf code.tar.bz2 code/

1、\*.tar 用 tar –xvf 解压

2、\*.gz 用 gzip -d或者gunzip 解压

3、\*.tar.gz和\*.tgz 用 tar –xzf 解压

4、\*.bz2 用 bzip2 -d或者用bunzip2 解压

5、\*.tar.bz2用tar –xjf 解压

6、\*.Z 用 uncompress 解压

7、\*.tar.Z 用tar –xZf 解压

8、\*.rar 用 unrar e解压

9、\*.zip 用 unzip 解压

set http\_proxy=www-proxy.ao.ericsson.se:8080

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1、机器

阿里云：

sdjnwfd@126.com

wfdong123@

114.215.82.1

captPwD141223~@$^\*)+

（内网10.163.114.47）

台式机

180.164.247.42

10001

10002

10003

10004

captPwD141223~@$^\*)+

2、清理脚本

\* \*/1 \* \* \* /usr/local/nginx/clear\_log.sh

\* \*/1 \* \* \* /usr/local/nginx/dellog.sh

/etc/init.d/crond  start

3、网页

新首页：http://114.215.82.1:8080/

http://114.215.82.1:9111/

http://114.215.82.1:3333/static/status.html

netstat -n | awk '/^tcp/ {++S[$NF]} END {for(a in S) print a, S[a]}'

http://114.215.82.1:8009/basic\_status

http://114.215.82.1:8080/static/index/frozen/a518518/index.html

select \* from userinfo where parentid=169 and successcount/calledcount < 0.2;

python androidSerevr.py 3333 &

4、统计

网络流量：

sar -n DEV interval count

sar -n DEV 1 4

5、启动

/usr/local/nginx/sbin/nginx -c /usr/local/nginx/conf/nginx.conf

uwsgi --ini conf-check-v3.ini

uwsgi --ini conf-mdb-v3.ini

/usr/local/bin/memcached -d -m 1200 -u root -l 10.163.114.47 -p 12000 -c 10000 -P /tmp/memcached.pid

python sync.py

python androidSerevr.py 3333

python webServer.py

python webServer.py 9111

6、teamviewer：

zhangda733@gmail.com

captPwD123!@#

若快的登录选用户登录哈，不要选开发者登录：

http://www.ruokuai.com/

用户名1536255624

密码Uutest123

训练：

1.切分图片，2.降切分的图片保存到数据文件，3.从数据文件读取数据进行训练

前两个步骤在src->dl->PrepareDat.py里

PrepareData.py里的main函数有切分图片和保存数据的函数

刚开始图片好像是放到img->train->tencent下面的，记不清了，切分后的放到img->train->split下面

最后训练是用CNNWithoutCV2\_10.py

还有这个dirName，在PrepareData.py和CNNWithoutCV2\_10.py应该都有定义，就是数据存到哪个文件夹和从哪个文件夹取数据来训练

数据时一万张切分后的小图片存为一个文件，这样一个个的来存

D:->Projects->python->captchaTrain

scp -P 10001 test  root@180.164.187.128:/home/

ssh root@180.164.187.128 -p 10001

captPwD141223~@$^\*)+

service iptables stop

DLL:

http://pan.baidu.com/s/1hqJ3Y08 密码: rxkp

1536255624@qq.com

dadadongdong123

1:

c4ca4238a0b923820dcc509a6f75849b

本地git配置文件：

~/.gitconfig

[user]

    name = wfdong

    email = sdjnwfd@126.com

[http]

    proxy = www-proxy.exu.ericsson.se:8080

NEW:

service iptables  stop

1、

199.33.121.227

root密码64db110f71

域名www.kmdama.com

ip.kmdama.com是dll里的地址 gl.kmdama.com是用户登录的地址

dlltest:

set http\_proxy=www-proxy.ao.ericsson.se:8080

C:\Python2732\python example.py

首页：

http://221.206.125.7/

http://221.206.125.7:9992/

看速度：

http://221.206.125.7:3333/static/status.html

http://221.206.125.7:8080/static/index/frozen/804123951/index.html

http://221.206.125.7/static/index/frozen/804123951/index.html

server:

221.206.125.7

7e727d90-7915-11e4-89be-e41f13653bfc

台式机：

yanzhengma.f3322.net

666566

1号：

10001

2号：

10002

3号：

10003

6号：

10006

7号：

10007

8号：

10008

10号：

11010

uwsgi --ini conf-check-v3.ini

uwsgi --ini conf-mdb-v3.ini

/usr/local/bin/memcached -d -m 2800 -u root -l 221.206.125.7 -p 12000 -c 10000 -P /tmp/memcached.pid

/usr/local/nginx/sbin/nginx -c /usr/local/nginx/conf/nginx.conf

uwsgi --emperor /home/code/uwsgi-v4 --daemonize = 127.0.0.1:5001

/static/doc/e\_call\_sample/e\_sample.zip

er6e@qq.com  weiwei666566

eer33@qq.com hezuo666566