Caffe环境安装配置

**1、**设置root用户，步骤如下：

打开终端，输入

$ sudo gedit /usr/share/lightdm/lightdm.conf.d/50-ubuntu.conf

在文件50-ubuntu.conf中加入：greeter-show-manual-login=true 保存关闭。

再在终端中输入：

$ sudo passwd root

设置root用户的密码。

以下操作都在root用户下进行。

使用root账户登录时候报Error found when loading /root/.profile错误。解决方法如下：

在打开/root/.profile文件后，我们把mesg n 进行注释，增加一行 tty -s && mesg n。然后执行命令init 6。进行系统的重启。

进入root后可能出现屏幕显示分辨率不对，此时可能没有正确安装nvidia驱动，不要着急，此步不安装nvidia驱动，在下面安装cuda的时候会自动安装nvidia驱动。然后重新进入系统后屏幕分辨率将显示正常。

**2、**屏蔽nouveau，步骤如下：

创建文件/etc/modprobe.d/blacklist-nouveau.conf，在文件中加入内容：

blacklist nouveau

options nouveau modeset=0

更新kernel initramfs，

$ sudo update-initramfs -u

**3、**安装开发所需的依赖包

sudo apt-get update

$ sudo apt-get install build-essential

$ sudo apt-get install vim cmake git

sudo apt-get install libatlas-base-dev

sudo apt-get install freeglut3-dev

$ sudo apt-get install libprotobuf-dev libleveldb-dev libsnappy-dev libopencv-dev libboost-all-dev libhdf5-serial-dev

安装glog (已下载)

$ wget <https://google-glog.googlecode.com/files/glog-0.3.3.tar.gz> （访问不了谷歌，可自行下载,若已经下载,不用执行此步）

$ tar zxvf glog-0.3.3.tar.gz

$ cd glog-0.3.3

$./configure

$ make && make install

1.下载

git clone https://github.com/google/glog

2.配置

sudo apt-get install autoconf automake libtool

3.编译&安装

进入源码根目录

./autogen.sh

./configure

make -j 24

sudo make install

卸载cmake2.8 安装cmake-3.1.0

先卸载

sudo apt-get remove cmake

再安装

tar zxvf cmake-3.1.0.tar.gz

cd cmake-3.1.0

sudo ./bootstrap

sudo ./configure --prefix=/usr/local/cmake

sudo make

sudo make install

瞬间就安装完成。可以看到已成功安装到/usr/local/cmake目录下。

sudo gedit /etc/profile

export PATH="$PATH:/usr/local/cmake/bin"

source /etc/profile

echo $PATH

安装gflags

git clone https://github.com/gflags/gflags

$ cd gflags

$ mkdir build

$ cd build

$ export CXXFLAGS="-fPIC"

$ cmake ..

$ make VERBOSE=1

$ make

$ make install

安装lmdb (已经下载,名称为liblmdb”)

unzip lmdb\_mdb.master.zip

cd lmdb\_mdb.master/libraries/liblmdb

make

sudo make install

$ git clone git://gitorious.org/mdb/mdb.git (若已经下载,不必执行此步骤)

$ cd mdb/libraries/liblmdb

$ make

$ make install

安装pip

$ sudo apt-get install python-pip python-dev build-essential

$ sudo pip install --upgrade pip

$ sudo pip install --upgrade virtualenv

<https://github.com/jayrambhia/Install-OpenCV>

**4、**安装 Atlas

$ sudo apt-get install libatlas-base-dev

**5、**安装OpenCV（暂未安装）

需重新下载

sudo apt-get update

sudo apt-get upgrade

解压Install-OpenCV-master.zip, 此为安装脚本,若没有下载请先下载,下载网址为:<https://github.com/jayrambhia/Install-OpenCV>.

进入Ubuntu目录，给所有shell脚本加上可执行权限

$ chmod +x \*.sh

为了避免[OpenCV LIBTIFF link errors](http://www.cnblogs.com/platero/p/4141063.html)，在opencv\_install.sh文件中的cmake加入-D BUILD\_TIFF=ON，然后运行安装文件：

$ sudo ./opencv\_latest.sh

**6、**安装protobuf

先安装 automake 等工具

$  sudo apt-get install autoconf automake libtool curl

解压protobuf-master.zip,若没下载,请先从<https://github.com/google/protobuf>下载

进入protobuf-master 目录后，打开autogen.sh文件，搜索关键字gmock,查看gmock下载地址是否为google链接,如果不是,则不用处理gmock; 如果是则用“#”将“curl $curlopts -O https://googlemock.googlecode.com/files/gmock-1.7.0.zip”注释掉，因为国内无法链接google。手动下载gmock(我已经下载)，将其放入protobuf-master 目录下。

运行以下指令，安装protobuf

$ cd protobuf-master/

$ sudo ./autogen.sh

$ sudo ./configure

$ make -j8

$ make check -j8

$ make install

$ sudo ldconfig

cd python

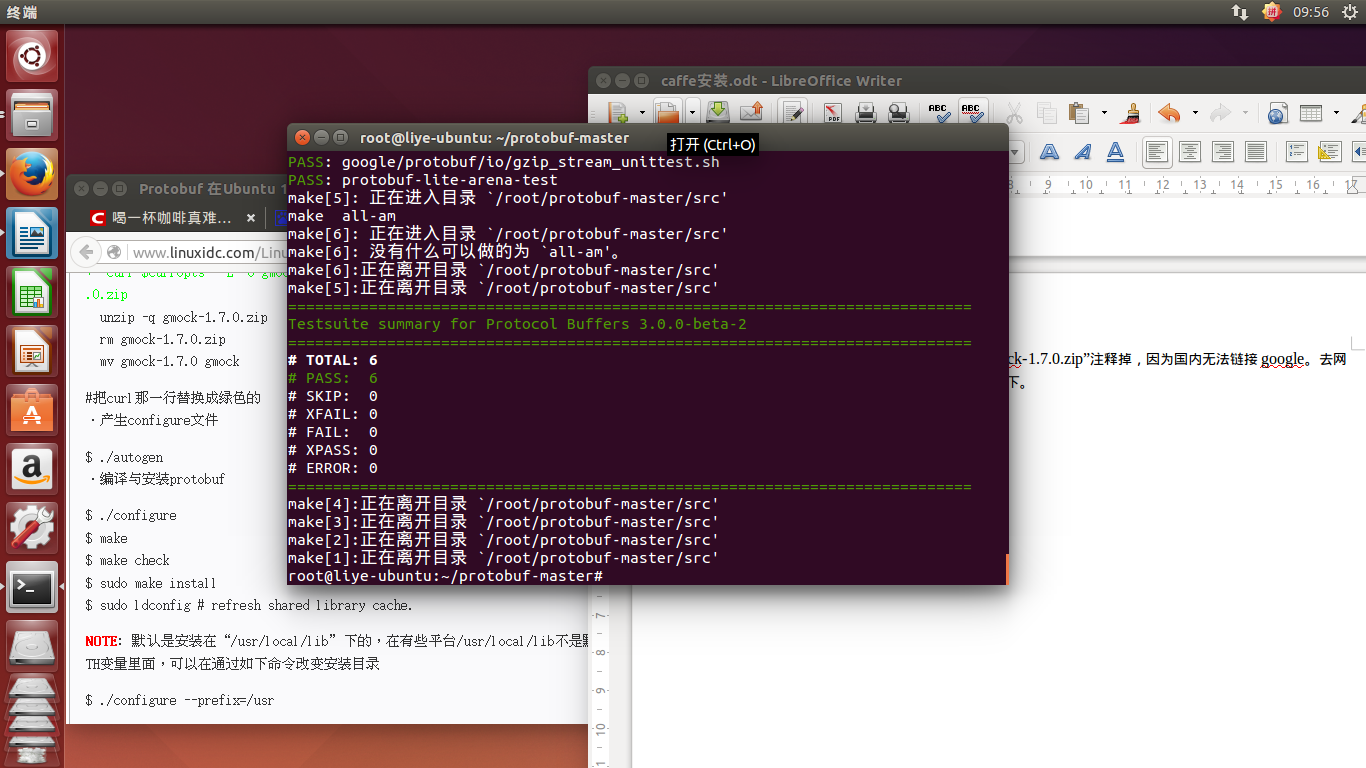
python setup.py build

python setup.py test

python setup.py install

马

结果如下，则安装成功



**7、**安装matlab

注意：在安装前请查看安装目录是否有足够空间！完全安装大概需要12G的空间！

(1). 安装程序放在Matlab2015b-linux文件夹中, 挂载ISO镜像文件

$ sudo mkdir /media/matlab

$ sudo mount -o loop R2015b\_glnxa64.iso /media/matlab

(2). 执行安装过程，选择不联网安装，序列号在~/crack/readme.txt文件中

$ cd /media/matlab

$ sudo ./install

根据弹出的安装图形界面提示安装, 如果安装的时候没有弹出图形安装界面，直接显示installing… finish! 则需要gksudo 或kdesudo 来安装

(3). 安装完毕并且将~/crack/bin/中的文件复制到~/MATLAB/Rxxxx/bin中

(4).首次运行matlab要用root权限（否则无法写文件），采用不联网激活，找到Crack中相应的激活文件\*.lic，导入激活。

$ cd /usr/local/MATLAB/R2015b/bin

$ sudo ./matlab

(5). 卸载ISO镜像。

$ sudo umount /media/matlab

(6). 添加桌面快捷方式。

新建一个桌面配置文件，文件名为Ｍatab2015b.desktop：

$ sudo gedit /usr/share/applications/Matlab2015b.desktop ，

在文件中写入以下内容：若安装在其他地方，更改/usr/local/为安装目录即可

[Desktop Entry]  
Type=Application  
Name=Matlab  
GenericName=Matlab 2015b  
Comment=Matlab:The Language of Technical Computing  
Exec=sh /usr/local/MATLAB/R2015b/bin/matlab -desktop  
Icon=/usr/local/MATLAB/R2015b/toolbox/shared/dastudio/resources/MatlabIcon.png  
Terminal=false  
Categories=Development;Matlab;

重启后，在/usr/share/applications中会出现对应的matlab桌面图标，将此图标复制到桌面即可

如果安装不正确请卸载:

$ sudo rm -rf /usr/local/MATLAB/R2015b

$ sudo rm /usr/local/bin/matlab /usr/local/bin/mcc /usr/local/bin/mex /usr/local/bin/mbuild   
仅供参考，这些文件可能不存在

(7). 将export PATH=/usr/local/MATLAB/R2015b/bin:$PATH路径放入~/.bashrc中

如遇到关于Matlab不能启动的问题:matlab has encountered an internal problem and needs to close

解决方法:强制Matlab使用操作系统提供的库即可, 进入Matlab的安装目录,继续导航到该目录下的sys/os/glnxa64目录, 然后重命名该目录下的文件libstdc++.so.6为libstdc++.so.6.old, 问题解决. 此问题会出现在15.04或更高版本的Ubuntu系统上, 比如我就是从14.10升级到16.04后就出现这个问题.

---------------------

**8、**安装CUDA，

(1) 先进行准备工作，

Verify You Have a CUDA-Capable GPU：

$ lspci | grep -i nvidia

Verify You Have a Supported Version of Linux:

$ uname -m

$ cat /etc/\*release

Verify the System Has gcc Installed:

$ gcc –version

Verify the System has the Correct Kernel Headers and Development Packages Installed:

$ uname -r

$ sudo apt-get install linux-headers-$(uname -r)

(2) 准备工作完成后,安装cuda,将cuda\_7.5.18\_linux.run文件放在/root的根目录下

Ctrl+Alt+F1进入命令行，在命令行界面执行以下操作

$ sudo service lightdm stop

$ sudo sh cuda\_7.5.18\_linux.run

安装时一般都按默认提示操作，但当提示OpenGL是否安装时选择no，否则登录桌面会出问题。

(3) Device Node Verification

Check that the device files /dev/nvidia\* exist and have the correct (0666) file permissions.

如果没有/dev/nvidia\*文件存在，则手动添加：

将你的启动脚本建立到 /etc/init.d目录下

$ sudo gedit /etc/init.d/nvidia

在nvidia.sh脚本中加入以下内容：

#!/bin/bash

/sbin/modprobe nvidia

if [ "$?" -eq 0 ]; then

# Count the number of NVIDIA controllers found.

NVDEVS=`lspci | grep -i NVIDIA`

N3D=`echo "$NVDEVS" | grep "3D controller" | wc -l`

NVGA=`echo "$NVDEVS" | grep "VGA compatible controller" | wc -l`

N=`expr $N3D + $NVGA - 1`

for i in `seq 0 $N`; do

mknod -m 666 /dev/nvidia$i c 195 $i

done

mknod -m 666 /dev/nvidiactl c 195 255

else

exit 1

fi

/sbin/modprobe nvidia-uvm

if [ "$?" -eq 0 ]; then

# Find out the major device number used by the nvidia-uvm driver

D=`grep nvidia-uvm /proc/devices | awk '{print $1}'`

mknod -m 666 /dev/nvidia-uvm c $D 0

else

exit 1

fi

设置脚本文件的权限

$ sudo chmod 755 /etc/init.d/nvidia

执行如下命令将脚本放到启动脚本中去：

$ cd /etc/init.d

$ sudo update-rc.d nvidia defaults 99

注：其中数字99是脚本启动的顺序号，按照自己的需要相应修改即可。在你有多个启动脚本，而它们之间又有先后启动的依赖关系时你就知道这个数字的具体作用了。该命令的输出信息参考如

(4) 如果以上启动脚本设置有问题,则卸载启动脚本,方法如下,若没有问题不执行此步骤

$ cd /etc/init.d

$ sudo update-rc.d -f nvidia remove

命令输出的信息参考如下：

Removing any system startup links for /etc/init.d/test ...

/etc/rc0.d/K95test

/etc/rc1.d/K95test

/etc/rc2.d/S95test

/etc/rc3.d/S95test

/etc/rc4.d/S95test

/etc/rc5.d/S95test

/etc/rc6.d/K95test

**9、**设置cuda的环境变量

将export PATH=/usr/local/cuda-7.5/bin:$PATH 、export LD\_LIBRARY\_PATH=/usr/local/cuda-7.5/lib64:$LD\_LIBRARY\_PATH添加到~/.bashrc文件中

在/etc/ld.so.conf.d/中添加cuda.conf文件

$ sudo gedit /etc/ld.so.conf.d/cuda.conf

cuda.conf文件中加入

/usr/local/cuda-7.5/lib64

保存后，执行下列命令使之立刻生效

$ sudo ldconfig

**10、**跑cuda例子

进入 /usr/local/cuda/samples , 执行下列命令来 build samples

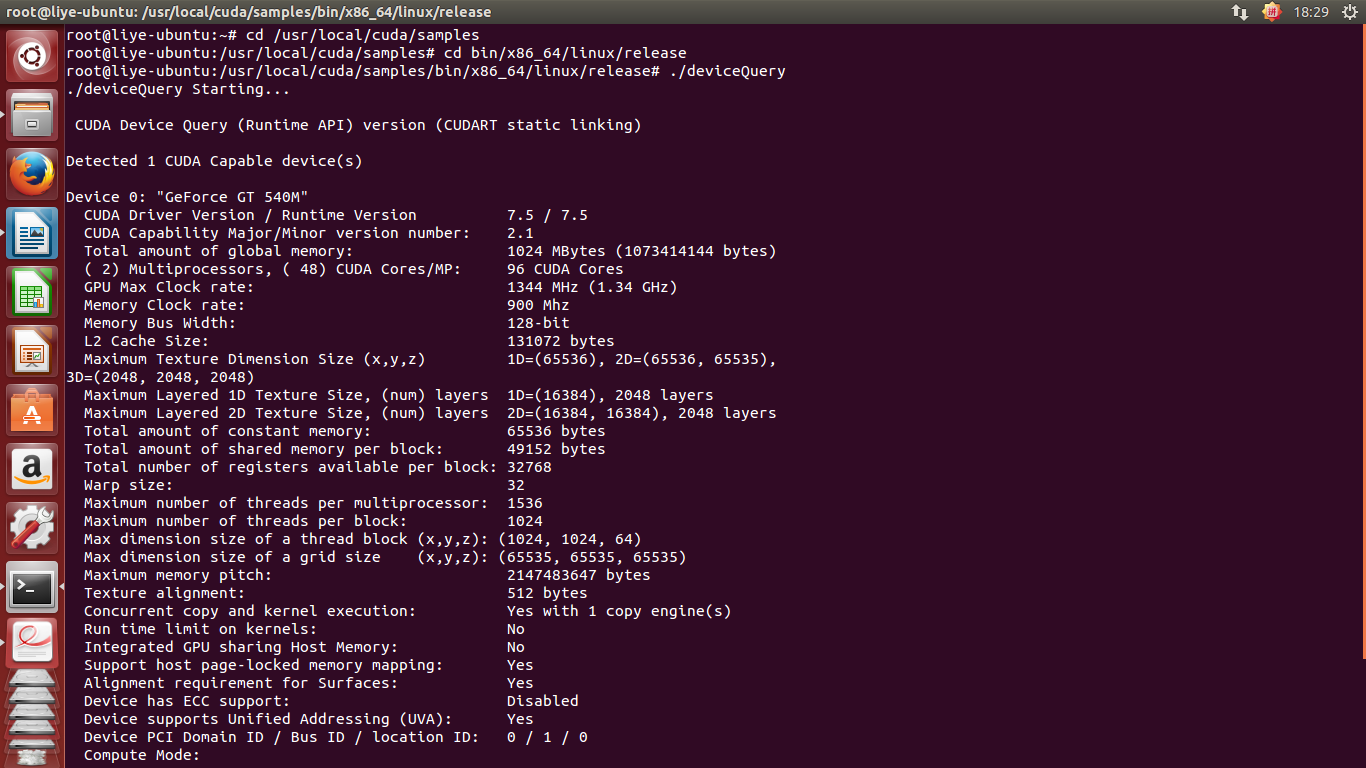
$ cd /usr/local/cuda/samples

$ sudo make all -j8

整个过程大概 10 分钟左右 , 全部编译完成后, 进入 samples/bin/x86\_64/linux/release , 运行 deviceQuery

$ ./deviceQuery

如果出现显卡信息, 则驱动及显卡安装成功，如下所示



若跑cuda例子出现以下问题：

1)/usr/bin/ld: 找不到 -lglut

解决方法： sudo apt-get install freeglut3-dev

**11、**cuda安装补全

apt-get install nvidia-cuda-toolkit

$ nvcc -V

**12、**安装cudnn

准备好cudnn-7.0-linux-x64-v4.0-prod.tgz文件,解压此文件

$ tar -zxvf cudnn-7.0-linux-x64-v4.0-prod.tgz

解压后为cuda文件夹，修改文件夹中lib\*文件的权限，否则在caffe make runtest时会出现找不到libcudnn\*库的情况

$ cd cuda (此步为进入cuda文件夹,请根据自己的cuda路径设置此处路径)

$ sudo chmod +x \*

$ sudo chmod u=rwx,g=rx,o=rx lib64/lib\*

$ sudo cp lib64/lib\* /usr/local/cuda-7.5/lib64/ （注意换成自己的cuda安装路径）

$ sudo cp include/cudnn.h /usr/local/cuda-7.5/include/（注意换成自己的cuda安装路径）

更新软链接

$ cd /usr/local/cuda-7.5/lib64/

$ sudo rm -rf libcudnn.so libcudnn.so.4

$ sudo chmod u=rwx,g=rx,o=rx lib\*

$ sudo chmod u=rwx,g=rx,o=rx libcudnn.so.4.0.7

$ sudo ln -s libcudnn.so.4.0.7 libcudnn.so.4

$ sudo ln -s libcudnn.so.4 libcudnn.so

$ sudo ldconfig

cudnn-7.0-linux-x64-v4.0-rc.tgz

更新：

$ sudo tar xvf cudnn-7.0-linux-x64-v4.0-rc.tgz

$ cd cuda/include

$ sudo cp \*.h /usr/local/include/

$ cd ../lib64

$ sudo cp lib\* /usr/local/lib/

$ cd /usr/local/lib

$ sudo chmod +r libcudnn.so.4.0.7

$ sudo ln -sf libcudnn.so.4.0.7 libcudnn.so.4

$ sudo ln -sf libcudnn.so.4 libcudnn.so

$ sudo ldconfig

**13、**caffe编译

准备caffe-master.zip, 解压后进入caffe-master文件夹，复制Makefile.config.example文件，并重命名为Makefile.config

修改caffe-master文件夹下Makefile.config文件，修改其中cuda、matlab安装路径、python路径。 可从已配置好环境的机器中拷贝Makefile.config文件,直接使用,我放在\root根目录下了

1、只需要把 Makefile.config里的 #USE\_PKG\_CONFIG := 这一行前面的#给去掉，然后在他下一行添加

LIBRARIES += glog gflags protobuf leveldb snappy \

lmdb boost\_system hdf5\_hl hdf5 m \

opencv\_core opencv\_highgui opencv\_imgproc opencv\_imgcodecs

保存，然后再编译即可

2、 结果提示：

./build/examples/openpose/openpose.bin: error while loading shared libraries: libopencv\_core.so.3.4: cannot open shared object file: No such file or directory

是共享链接库出现了问题，所以就搜索libopencv\_core.so.3.4，我最后是在/usr/local/lib/x86\_64-linux-gnu下面找到的，所以要把这个路径添加到系统进去。可以根据你这个库的名字新建一个。

sudo gedit /etc/ld.so.conf.d/libopencv\_core.conf

我这里是libopencv\_core.conf，你可以改为对应你的xxx.conf。然后把我们找到的libopencv\_core.so.3.4的路径以及它的上一个路径添加进去，我是这样子添加的：

/usr/local/lib/

/usr/local/lib/x86\_64-linux-gnu/

修改caffe-master文件夹下Makefile文件，在libraries+=中加入opencv\_imgcodecs

3、.build\_release/lib/libcaffe.so：对‘leveldb::DB::Open(leveldb::Options const&, std::string const&, leveldb::DB\*\*)’未定义的引用 .build\_release/lib/libcaffe.so：对‘leveldb::Status::ToString() const’未定义的引用

解决方法：首先查找leveldb的位置 ，我的位置是在/usr/include/中

将位置添加进Makefile.config 的

INCLUDE\_DIRS := $(PYTHON\_INCLUDE) /usr/local/include /usr/lib/x86\_64-linux-gnu/hdf5/serial/include /usr/include/leveldb

LIBRARY\_DIRS := $(PYTHON\_LIB) /usr/local/lib /usr/lib /usr/lib/x86\_64-linux-gnu /usr/lib/x86\_64-linux-gnu/hdf5/serial /usr/include/leveldb

再重新编译即可

4、] SGDSolverTest/0.TestSnapshotShare, where TypeParam = caffe::CPUDevice<float>

[ FAILED ] AdaGradSolverTest/0.TestSnapshotShare, where TypeParam = caffe::CPUDevice<float>

解决方法：export MKL\_CBWR=”AUTO”

5、[ FAILED ] RMSPropSolverTest/0.TestSnapshotShare, where TypeParam = caffe::GPUDevice<float>

解决方法：

export CUDA\_VISIBLE\_DEVICES=0

然后重新make runtest

6、若4，5更改之后还是不行，需要进行以下操作：

首先：(卸载已经安装的Boost库，只适用于安装Boost库时使用的是命令sudo apt-get install libboost-all-dev的情况。源码安装的删除方法还不明确。)

sudo apt-get autoremove libboost-all-dev

然后按照以下步骤操作（源码安装Boost1.55.0，因为Ubuntu16.04使用apt-get install安装的是Boost1.58.0，所以这里用源码安装。）

1. 手动下载Boost库1.55。

wget -O boost\_1\_55\_0.tar.gz http://sourceforge.net/projects/boost/files/boost/1.55.0/boost\_1\_55\_0.tar.gz/download

tar xzvf boost\_1\_55\_0.tar.gz

cd boost\_1\_55\_0/

2. 安装依赖库。（建议跳过这一步）

sudo apt-get update

sudo apt-get install build-essential g++ python-dev autotools-dev libicu-dev build-essential libbz2-dev

3. 用Boost库的bootstrap方式安装。

./bootstrap.sh --prefix=/usr/local

4. 如果需要使用MPI，就得设置user-config.jam文件中标志。

user\_configFile=`find $PWD -name user-config.jam`

echo "using mpi ;" >> $user\_configFile

5. 查询CPU核的数目。

n=`cat /proc/cpuinfo | grep "cpu cores" | uniq | awk '{print $NF}'`

6. 并行安装Boost库。

sudo ./b2 --with=all -j $n install

7. 如果系统中/usr/local/lib不存在，就需要添加到LD LIBRARY PATH中。

sudo sh -c 'echo "/usr/local/lib" >> /etc/ld.so.conf.d/local.conf'

8. 重置ldconfig。

sudo ldconfig

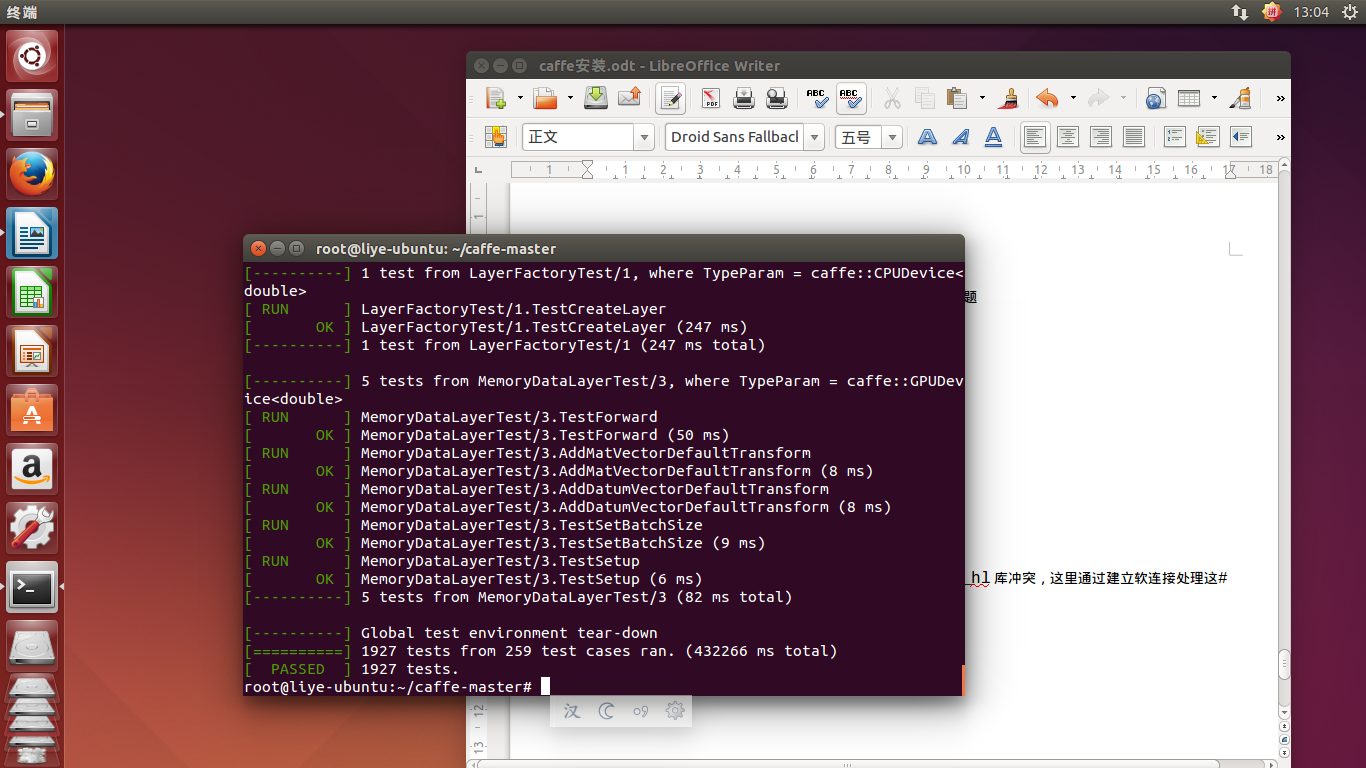
开始编译caffe

$ sudo make all -j8

$ sudo make test -j8

$ sudo make runtest -j8

若编译成功，则显示



编译caffe时可能出现以下问题,解决方法如下:

1）当make all时出现//usr/lib/...lib\*\*.so.0:undefined reference...@GLIBC\_2.5的问题

解决方案：

$ rm ~/anaconda/lib/libm.\*

2） 当make rutest时出现.build\_release/test/...:error while loading shared libraries:libhdf5\_hl.so.8:cannot open shared object file:No such file or directory

这是由于hdf5\_hl库与系统中hdf5\_hl库冲突引起，通过建立软连接处理这个问题。

解决方法：

$ cd /usr/lib/x86\_64-linux-gnu

$ sudo ln -s libhdf5.so.7 libhdf5.so.8

$ sudo ln -s libhdf5\_hl.so.7 libhdf5\_hl.so.8

$ sudo ldconfig

3）caffe编译时如果出现hdf5相关错误，可能与python库冲突有关，即anaconda中python与系统自带的python冲突，注意在caffe文件夹中Makefile.config中python的设置。

如果还不行，执行

$ conda update –all

更新系统中anaconda

4）当出现“no CUDA-capable device is detected”，则在make前加入optirun，用于切换显卡

$ sudo optirun make runtest -j4

执行上面指令后可能会再报错，虽然报错但显卡应该切换到nvidia，此时再执行

$ sudo make runtest -j4

**14、**数据预处理

可以用下载好的数据集，也可以重新下载，具体操作如下：

新版的caffe只能在他的caffe-master主目录下操作，不然会报错  
$ sudo sh ./data/mnist/get\_mnist.sh

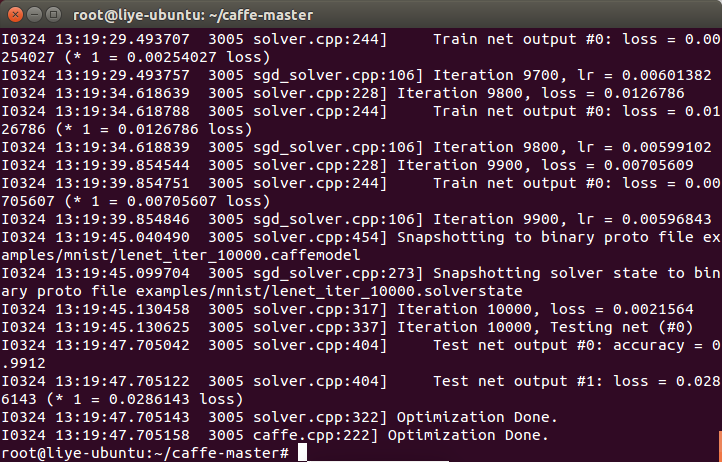
重建LDB文件，就是处理二进制数据集为Caffe识别的数据集，以后所有的数据，包括jpe文件都要处理成这个格式

$ sudo sh ./examples/mnist/create\_mnist.sh

生成mnist-train-leveldb/ 和 mnist-test-leveldb/文件夹，这里包含了LDB格式的数据集

训练mnist  
$ sudo sh ./examples/mnist/train\_lenet.sh

等待数秒之后会发现迭代了1000次，以及输出的数据等。如下图所示



**15、**编译Python wrapper

$ make pycaffe -j8

**16、**Eclipse 开发环境安装

(1) 安装Eclipse: sudo apt-get install eclipse

sudo apt-get install eclipse-platform

修改eclipse.ini 中的数字—尽量大

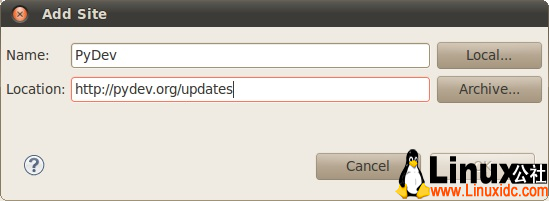
(2) 给Eclipse安装PyDev插件:

启用Eclipse，第一次启动会让你选择一个工作空间，按缺省设置，勾选一下不再提醒，就可以了。

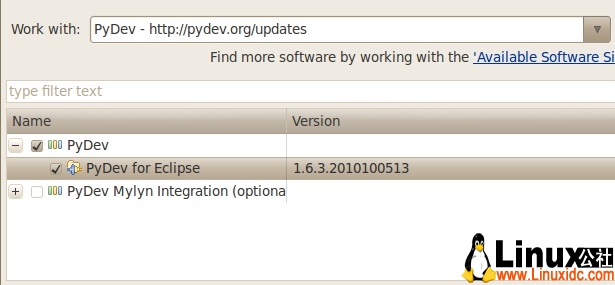
在Help菜单中，选择InstallNewSoftware···，

选择Add按钮，Name：PyDev（这个随便起），Location：http://pydev.org/updates（PyDev的更新地址），OK

http://update-production-pydev.s3.amazonaws.com/pydev/updates/site.xml



选择PyDev下的PyDevforEclipse，别的都不要选，否则依赖检查那关过不去。（注2）



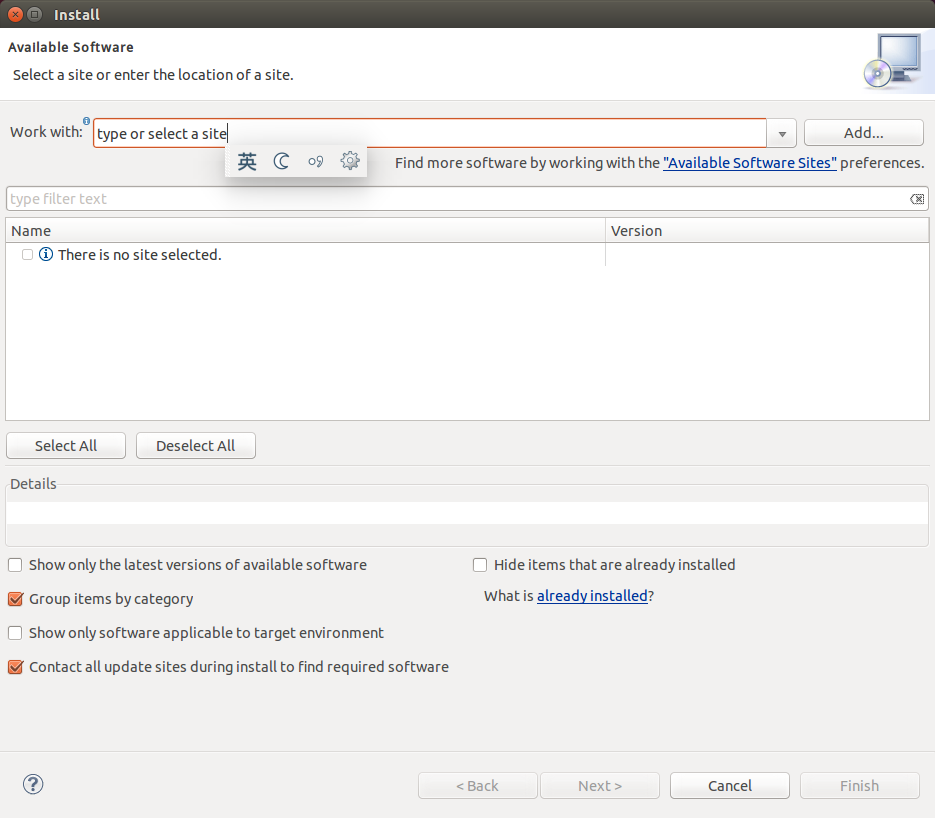
点下一步，下一步安装

中间会问你是否信任Aptana、Pydev什么的，信任就可以了

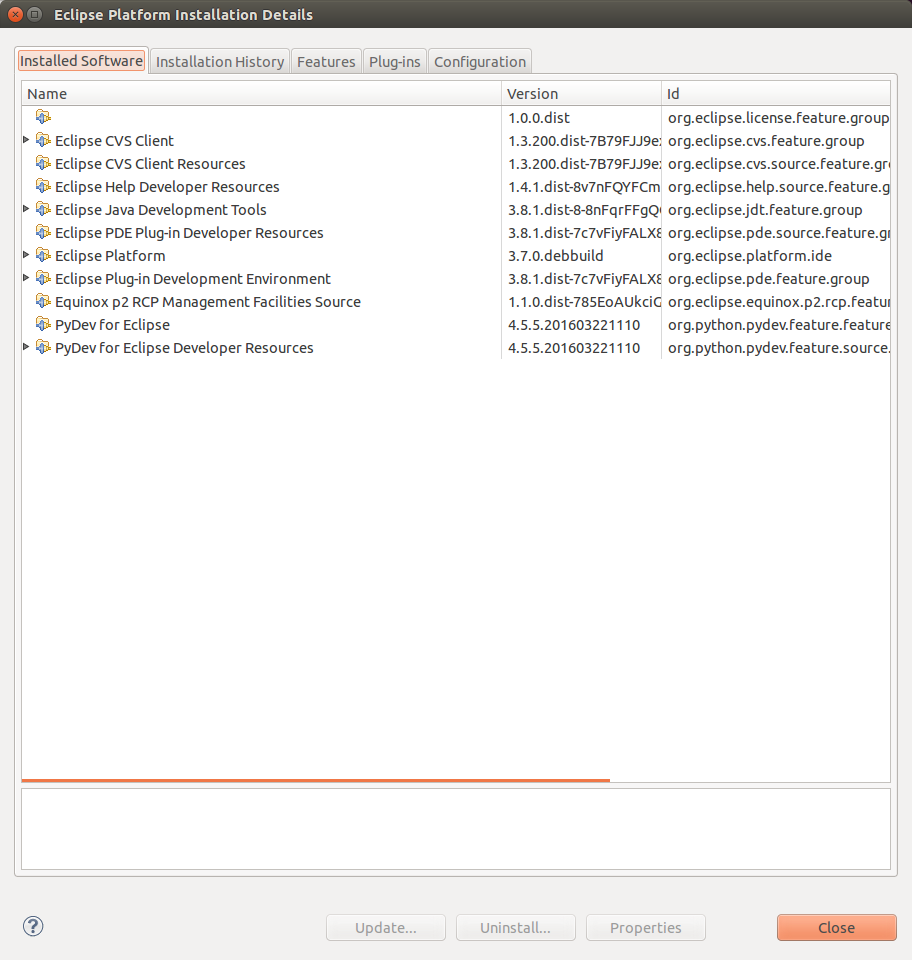
最后重启Eclips

若上述pydev安装不正确,则请先删除历史安装pydev版本,方法如下:

点下图中“already installed”



从下图中可以卸载已安装的pydev



(3) 如果已安装高版本pydev，有可能出现eclipse无法识别pydev，因为eclipse版本不够高，因此手动安装pydev，方法如下:

将PyDev 4.5.5zip解压到eclipse相应目录：/usr/share/eclipse/dropins/,

可以看到该目录底下多了两个文件夹features和plugins

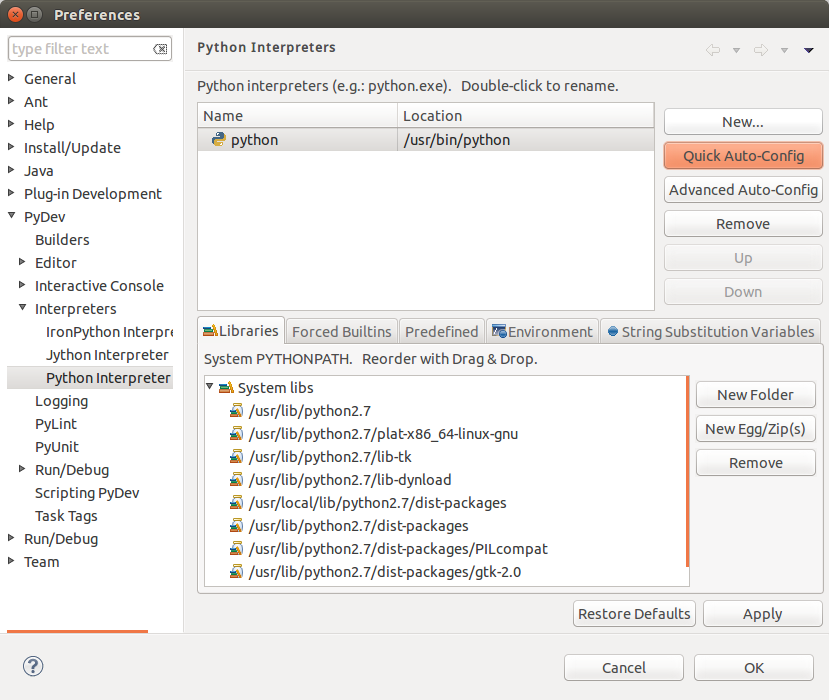
重启eclipse即可看到preference中的pydev选项。

(4) 配置pydev解释器

安装好pydev后， 需要配置Python解释器。

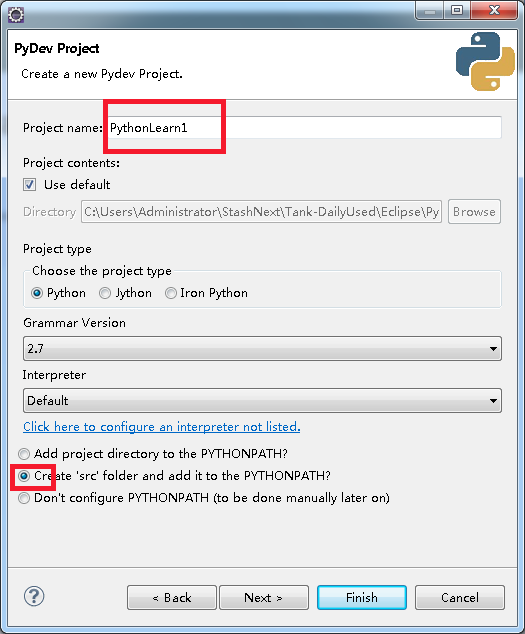
在Eclipse菜单栏中，点击Windows ->Preferences.

在对话框中，点击pyDev->Interpreter - Python.  点击Quick Auto-Config即可配置python 解释器,



(5) 开始写代码

启动Eclipse,  创建一个新的项目,   File->New->Projects...   选择PyDev->PyDevProject 输入项目名称.



新建 pyDev Package.    就可以写代码了。

(6) 在eclipse中python编译时可能遇到以下问题:

1) 问题: 在运行程序import lmdb时,出现错误:No module named lmdb

解决方法: 估计你用 pip install lmdb 时默认装到 python3 下了，然后你 import lmdb 是在 python2 环境下，所以会有这样的错误，建议用 pip2 install lmdb 试试

2) can not find module skimage.io

此时只要按照以下命令操作即可：  
$ sudo apt-get install python-numpy python-scipy python-matplotlib python-sklearn python-skimage python-h5py python-protobuf python-leveldb python-networkx python-nose python-pandas python-gflags Cython ipython

$ sudo apt-get update

然后重新编译pycaffe:

在caffe-master目录下：  
$ make pycaffe

3) ImportError: No module named setproctitle

$ sudo apt-get install python-pip

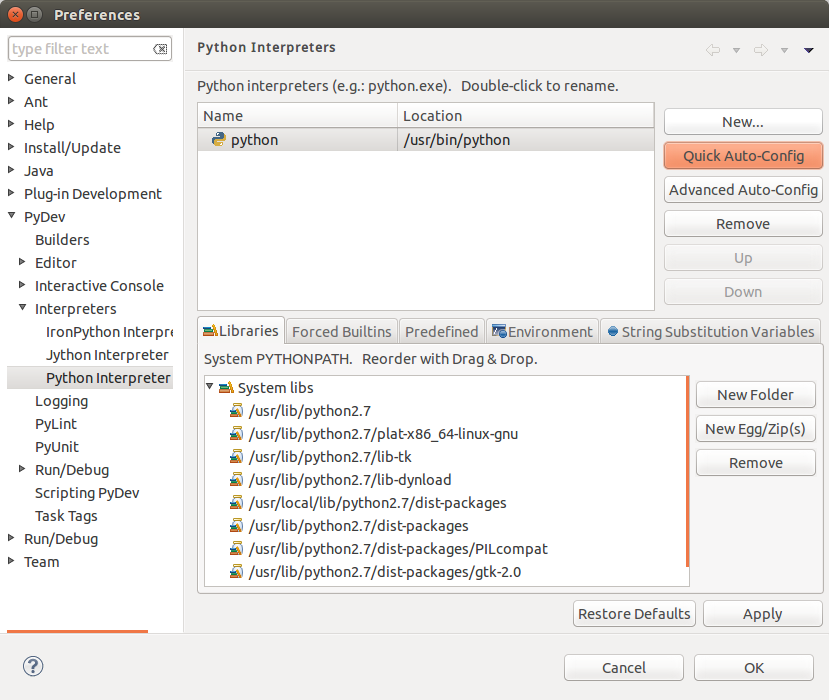
$ pip install setproctitle

4)ImportError: cannot import name symbol\_database,

5) No module named google.protobuf.internal

这说明你没有添加google.protobuf.的位置,eclipse软件设置中添加以下路径,如下图:

/usr/lib/python2.7/dist-packages/



6) can’t import symbol\_database 等类似错误TypeError: \_\_init\_\_() got an unexpected keyword argument ‘oneofs’ 等错错误

这些错误是google.protobuf 版本问题所导致的。

解决办法：下载protobuf-2.6.0

切换到/usr/lib/python2.7/dist-packages/ 目录下  先将google 文件夹备份

$ sudo mv google google\_back

将其protobuf-2.6.0/python 目录下的google文件拷贝到/usr/lib/python2.7/dist-packages/目录下即可。

这里需要将你刚备份的文件google\_back/protobuf/ 目录 descriptor\_pb2.py  拷贝到google/protobuf/的目录下。

**17、**若机器显示不正常,则重新安装nvidia驱动

1）卸载已装的nvidia

$ sudo apt-get remove --purge nvidia-\*

2）安装新的nvidia，Ctrl+Alt+F1进入命令行，在命令行界面执行以下操作，在安装nvidia驱动前关掉lightdm

$ sudo service lightdm stop

$ sudo apt-get install nvidia-352

$ sudo apt-get install nvidia-settings nvidia-prime

$ sudo prime-select intel

最后一条指令是将用于显示的显卡手动设定为intel，如果不是intel有可能出现黑屏情况，无法进入桌面

在编译caffe有可能遇到错误，“CUDA driver version is insufficient for CUDA runtime version”，说明nvidia驱动版本与cuda驱动版本不匹配，此时需要更新nvidia驱动

$ sudo apt-get install nvidia-352-dev

$ sudo apt-get install nvidia-352-updates

$ sudo apt-get install nvidia-352-updates-dev