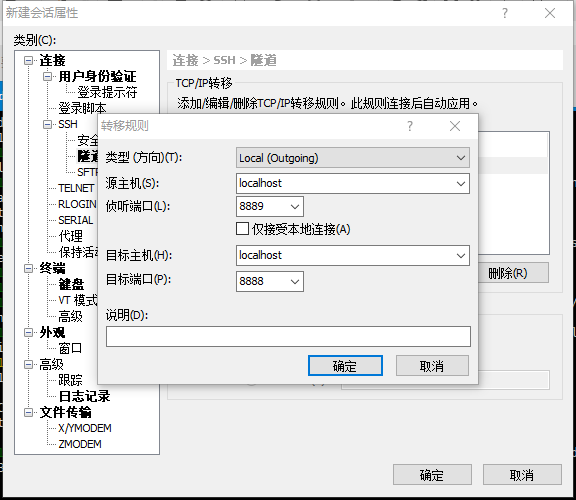
Cmd中！

ssh dl@5r84x0.nidaili.net -p 10339 -L localhost:8889:localhost:8888

其他操作的时候再用xshell吧

Xshell的设置



nidaili\_web

|  |  |
| --- | --- |
|  | [http://htugex.nidaili.net:10331](http://htugex.nidaili.net:10331" \t "_blank) |

注意br\_lan HWaddr 00:e0:70:35:12:63

ssh root@uli2qm.nidaili.net -p 10412 在linux下才正常

nidaili\_10.251 192.168.10.251:22 ei8crz.nidaili.net:10411

ssh root@ ei8crz.nidaili.net -p 10411

nidaili\_8.234 192.168.8.234:22 uli2qm.nidaili.net:10412

ssh root@ uli2qm.nidaili.net -p 10412

ssh [dl@192.168.1.3](mailto:dl@192.168.1.3) –L localhost:8889:localhost:8888

ssh [dl@5r84x0.nidaili.net](mailto:dl@5r84x0.nidaili.net) –L localhost:8889:localhost:8888 -p 10339

用户名 taobao81307

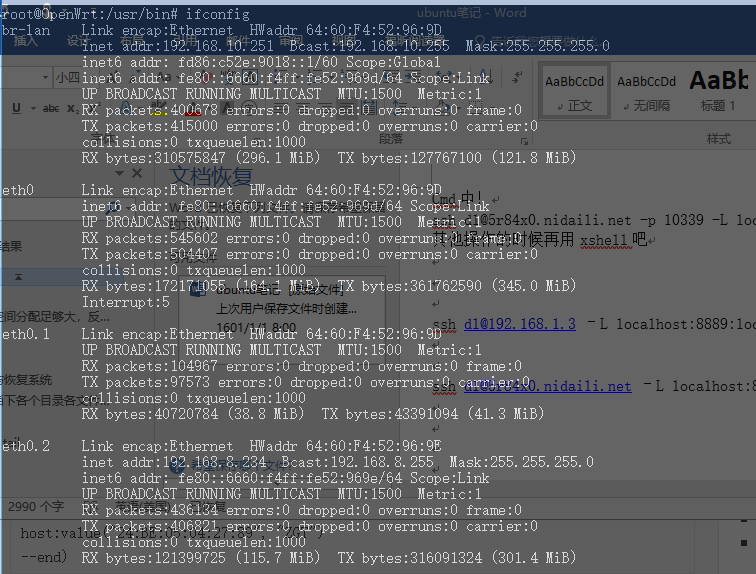
我自己服务器的地址！

enp3s0 Link encap:Ethernet HWaddr 00:e0:70:35:12:63

inet addr:192.168.10.186

这个路由器咋两个ip地址呢。

这两个地址都可以ssh连上。密码admin



# 虚拟机一次性空间分配足够大，反正也不会立刻占用，后期再改的时候，弄不好就开机特别慢。

# 远程开机现在不好用了。

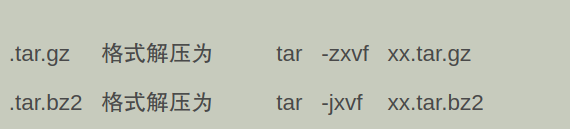
# Linux常用命令

#### echo $PATH

#### 软件安装目录

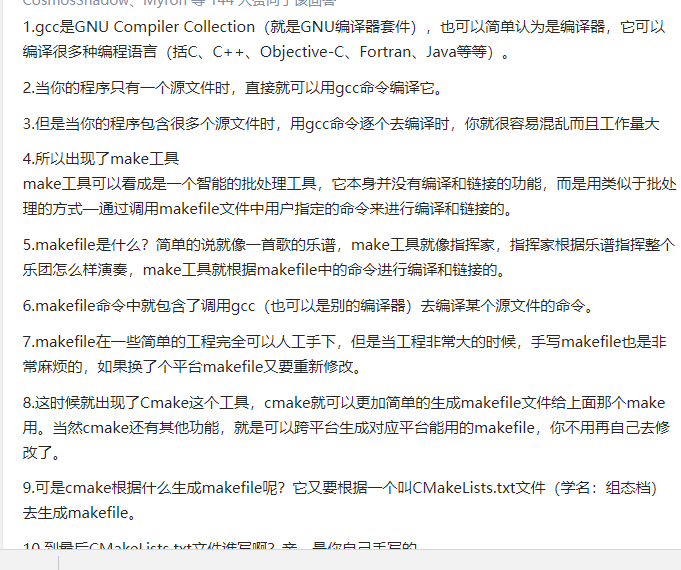


#### 解压 tar –jxvf -zxvf



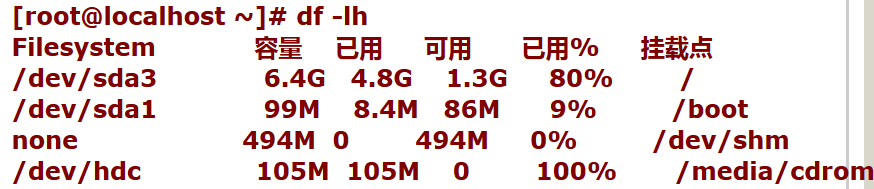
#### Makefile make cmake gcc

Makefile 乐谱，make 指挥家，cmake,生成makefile.

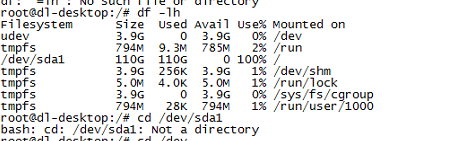




#df命令



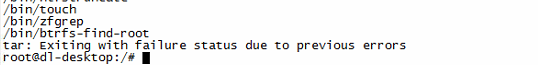
df命令是用来汇报磁盘文件系统的空间使用情况的，参数l表示本地文件系统，h表示以人可识别的方式表示(实际上就是以M/G单位来表示，而不是纯粹的数字，难以理解)。我们看到上面命令的结果是



#### 备份与恢复系统

Sudo tar -cvpzf backup.tgz / --exclude=/proc --exclude=/lost+found --exclude=/backup.tgz --exclude=/mnt --exclude=/sys --exclude=/media --exclude=/tmp --exclude=/home/dl/VirtualBox VMs /

最后有tar报错



网上都说忽略掉

#### 查看当下各个目录各文件大小

Fs –lh [h把文件的大小用K,G显示出来]

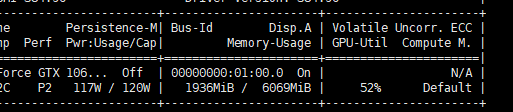
sudo du -h --max-depth=1

$ sudo df -h # 查看磁盘空间占用情况[可选操作]  
$ sudo fdisk -l # 查看磁盘设备信息[可选操作]

#### 监视gpu状态=确保功率和利用率是正常的！

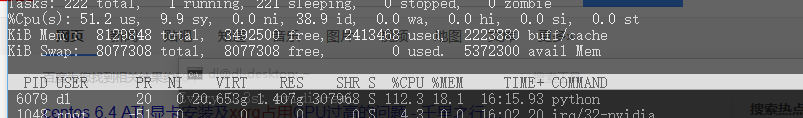
watch -n 0.3 nvidia-smi

watch -n 1 nvidia-smi

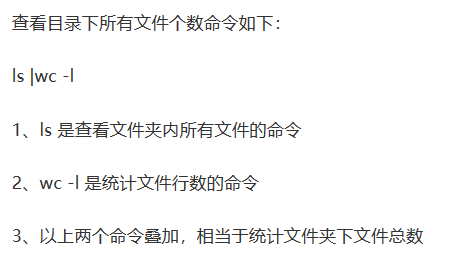


#### 内存监控

Gpu跑起来的时候 不咋耗费内存啊！



#### 查看有多少个文件



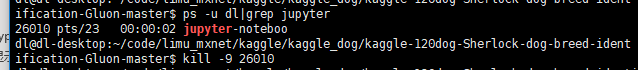
#### 显示 进程

-ps –u dl



#### 杀掉进程

kill -9 来强制终止退出



#### wc

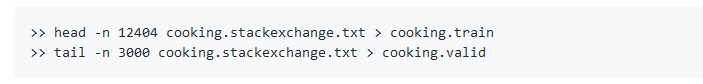
<http://www.cnblogs.com/peida/archive/2012/12/18/2822758.html>



#### Head tail

<http://www.cnblogs.com/peida/archive/2012/11/06/2756278.html>

-n 行数



#### Find

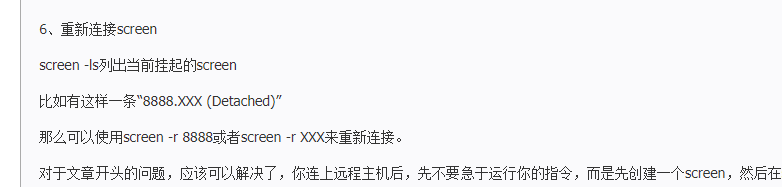
find /etc -name '\*srm\*'　　#使用通配符\*(0或者任意多个)。表示在/etc目录下查找文件名中含有字符串‘srm’的文件

#### locate

<http://www.cnblogs.com/peida/archive/2012/11/12/2765750.html>

#### find . -type f -print|wc –l 查看有多少个文件

#### screen



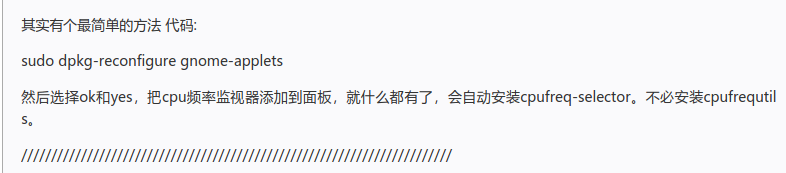
#### 标题栏显示cpu，内存占用

<http://blog.csdn.net/cyntec370/article/details/52973317>

sudo add-apt-repository ppa:fossfreedom/indicator-sysmonitor

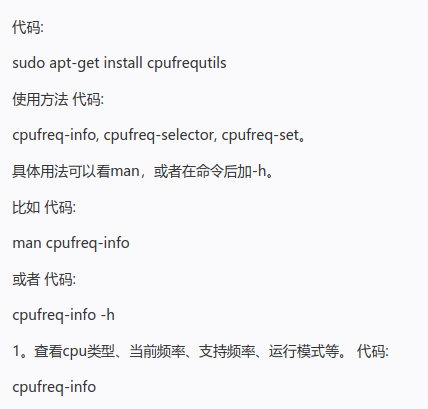
sudo apt-get update

sudo apt-get install indicator-sysmonitor

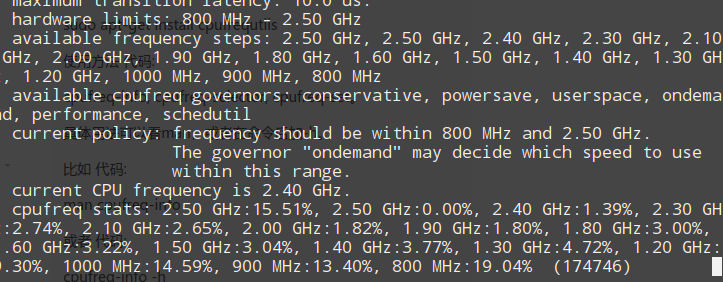


sudo dpkg-reconfigure gnome-applets

#### 显示cpu状态



watch -n 0.15 cpufreq-info

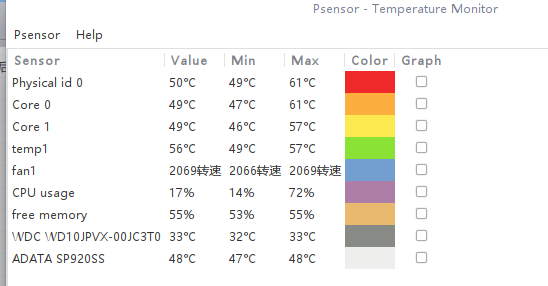


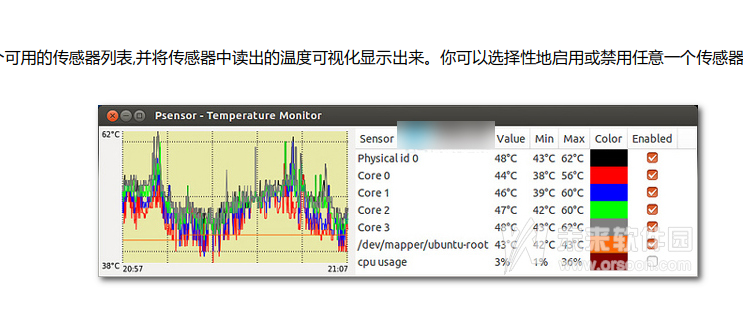
用ps命令 找到jupyter 进程并杀掉。

#### 显示各种温度

<http://www.orsoon.com/news/56755.html>

#pensor，部分报错可能和系统和ubuntu有区别





#### 依赖修复 用于安装软件

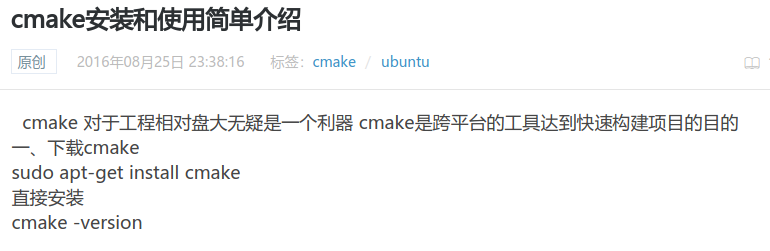
sudo apt-get -f -y install

F fix

#### Gpm复制粘贴快捷键

最后我没有弄

#### Cmake



报错 然后安装了gcc 和c++。

# 环境配置

禁止自动锁屏

### 初始root密码

我设置为了root

1、先用安装时候的用户登录进入系统

2、输入：sudo passwd  按回车

3、输入新密码，重复输入密码，最后提示passwd：password updated sucessfully

此时已完成root密码的设置

4、输入：su root

切换用户到root试试.......

### ssh

sudo apt-get install openssh-server

<http://blog.csdn.net/shengmingqijiquan/article/details/52367657>

还需要启动这个服务

本机secureCRT关闭声音

CRT为每一个回话都进行相同的

ssh -p 22047 [dl@c1838i6293.iok.la](mailto:dl@c1838i6293.iok.la)

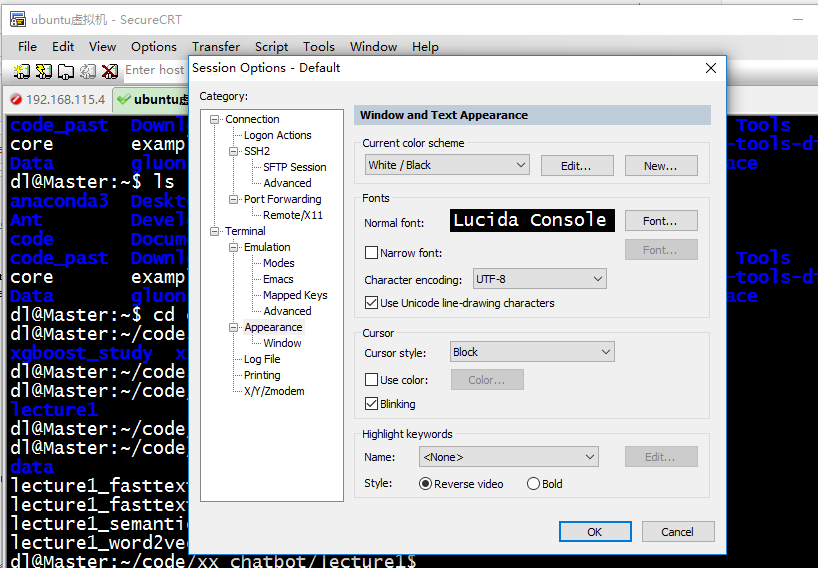
ssh –p 10339 dl@**[5r84x0.nidaili.net](http://192.168.10.251:8080/pages/service/listServices.lsp)**

这样就不用secureCRT了！！

#### SecureCRT 防掉线设置

<http://blog.sina.com.cn/s/blog_40e8378d0100oqhy.html>

### SecureCRT显示中文乱码



### vi

vi输入方向键会变成ABCD

？？但是我现在无法联网啊

ubuntu下 vi输入方向键会变成ABCD，这是ubuntu预装的是vim tiny版本，安装vim full版本即可解决。

先卸载vim-tiny：

$ sudo apt-get remove vim-common

再安装vim full：

$ sudo apt-get install vim

### 加入sudoer

Sudo vim /etc/sudoers, visudo 这个和vi的用法一样

移动光标，到一行root ALL=(ALL) ALL的下一行，输入：your\_user\_name ALL=(ALL) ALL；

your\_user\_name ALL=(ALL) NOPASSWD: ALL。



### 静态ip

我在图形界面安装的

虚拟机中要看vmnet8网卡，这个是NAT方式的

虚拟机的话，虚拟机本身也是要配置的

<https://www.ytyzx.org/index.php?title=%E5%A6%82%E4%BD%95%E9%85%8D%E7%BD%AEVMware%E5%AE%A2%E6%88%B7%E8%99%9A%E6%8B%9F%E6%9C%BA%E4%BD%BF%E7%94%A8NAT%E6%A8%A1%E5%BC%8F%E8%81%94%E7%BD%91%E9%80%9A%E4%BF%A1>

### 远程开机

### 换源

#### 系统源

更新Ubuntu16.04源，用的是[中科大的源](https://lug.ustc.edu.cn/wiki/mirrors/help/ubuntu" \t "_blank)：

cd /etc/apt/  
sudo cp sources.list sources.list.bak  
sudo vi sources.list

把下面的这些源添加到source.list文件头部：

*deb http://mirrors.ustc.edu.cn/ubuntu/ xenial main restricted universe multiverse  
deb http://mirrors.ustc.edu.cn/ubuntu/ xenial-security main restricted universe multiverse  
deb http://mirrors.ustc.edu.cn/ubuntu/ xenial-updates main restricted universe multiverse  
deb http://mirrors.ustc.edu.cn/ubuntu/ xenial-proposed main restricted universe multiverse  
deb http://mirrors.ustc.edu.cn/ubuntu/ xenial-backports main restricted universe multiverse  
deb-src http://mirrors.ustc.edu.cn/ubuntu/ xenial main restricted universe multiverse  
deb-src http://mirrors.ustc.edu.cn/ubuntu/ xenial-security main restricted universe multiverse  
deb-src http://mirrors.ustc.edu.cn/ubuntu/ xenial-updates main restricted universe multiverse  
deb-src http://mirrors.ustc.edu.cn/ubuntu/ xenial-proposed main restricted universe multiverse  
deb-src http://mirrors.ustc.edu.cn/ubuntu/ xenial-backports main restricted universe multiverse*

最后更新源和更新已安装的包：

sudo apt-get update

若报错<http://www.cnblogs.com/EasonJim/p/7343892.html>

sudo apt-get upgrade

#### pip源

修改pip软件源需要修改 *~/.pip/pip.conf*(没有就创建一个)文件，

[global]

index-url = <https://pypi.tuna.tsinghua.edu.cn/simple>

或

[global]

trusted-host = pypi.mirrors.ustc.edu.cn

index-url = https://pypi.mirrors.ustc.edu.cn/simple



##### Pip 常用命令

* List
* Search
* 指定版本==

#### conda 换源

conda config --prepend channels <https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free/>

<http://www.linuxidc.com/Linux/2014-04/100476.htm>

### 安装Anaconda

<http://blog.csdn.net/zhdgk19871218/article/details/46502637>

运行安装向导：

$ bash Anaconda3-4.2.0-Linux-x86\_64.sh





安装完成之后要重启终端，anaconda才能生效

或者

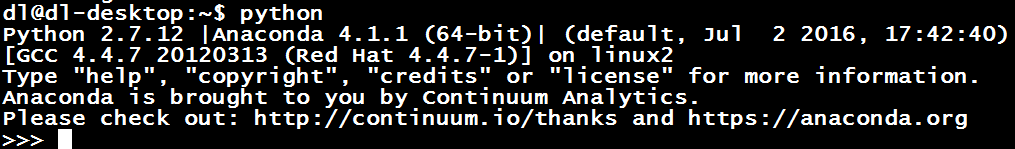


没有anaconda字样，说明还是原来的版本

安装的最后一步！



重启之后



#### conda换源

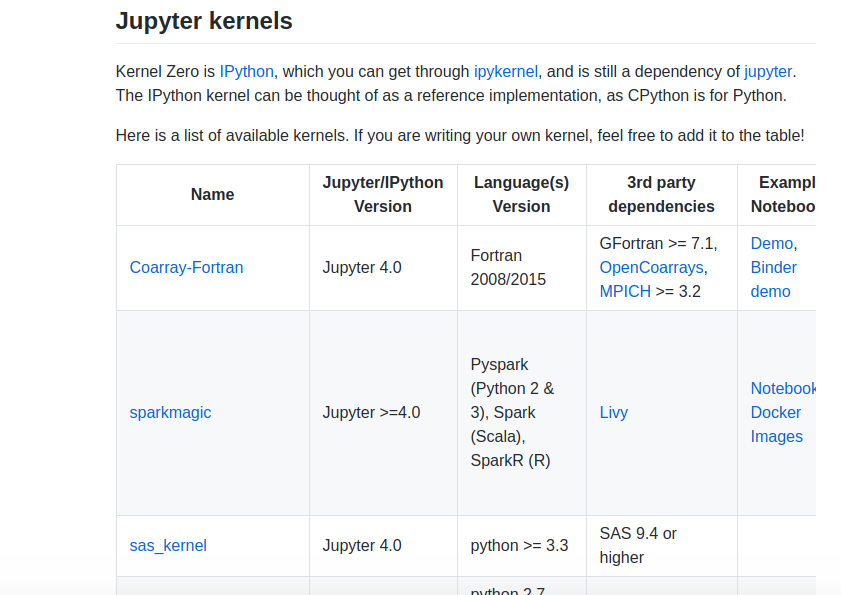
conda config --add channels 'https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free/'

conda config --set show\_channel\_urls yes

#### 2,3共存

### 各种kernel

<https://github.com/jupyter/jupyter/wiki/Jupyter-kernels>



### Cling(C++ kernel)

C的没有装好

<https://github.com/vgvassilev/cling/>

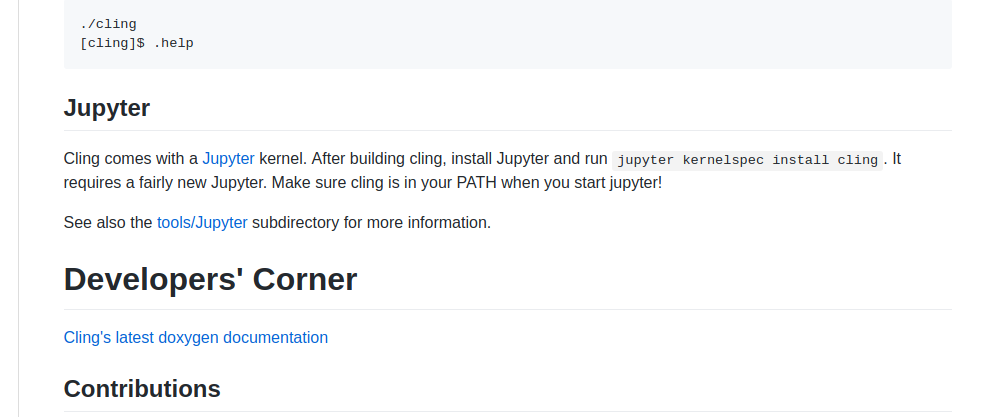
下载二进制那个

<https://root.cern.ch/download/cling/>



这一步不要忘记了

<https://github.com/vgvassilev/cling/tree/master/tools/Jupyter>



sudo /home/wangchen/anaconda3/bin/jupyter-kernelspec install cling-cpp11

sudo /home/wangchen/anaconda3/bin/jupyter-kernelspec install cling-cpp14

sudo /home/wangchen/anaconda3/bin/jupyter-kernelspec install cling-cpp17

### 安装Nvidia驱动

其实cuda中是包含有驱动的，

安装 Nvidia 驱动 375 370 （367.27

sudo add-apt-repository ppa:graphics-drivers/ppa

第一次运行出现如下的警告：

Fresh drivers from upstream, currently shipping Nvidia.

## Current Status

We currently recommend: `nvidia-361`, Nvidia’s current long lived branch.  
For GeForce 8 and 9 series GPUs use `nvidia-340`  
For GeForce 6 and 7 series GPUs use `nvidia-304`

## What we’re working on right now:

– Normal driver updates  
– Investigating how to bring this goodness to distro on a cadence.

## WARNINGS:

This PPA is currently in testing, you should be experienced with packaging before you dive in here. Give us a few days to sort out the kinks.

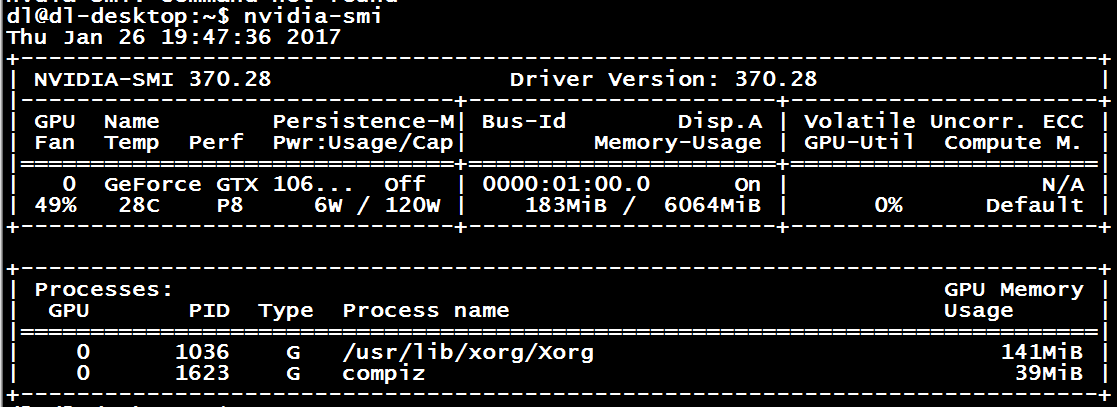
Volunteers welcome! See also: https://github.com/mamarley/nvidia-graphics-drivers/

http://www.ubuntu.com/download/desktop/contribute  
更多信息： https://launchpad.net/~graphics-drivers/+archive/ubuntu/ppa  
按回车继续或者 Ctrl+c 取消添加

回车后继续:

sudo apt-get update  
sudo apt-get install nvidia-375  
sudo apt-get install mesa-common-dev  
sudo apt-get install freeglut3-dev

之后重启系统让[GTX1060](http://www.52nlp.cn/%E6%B7%B1%E5%BA%A6%E5%AD%A6%E4%B9%A0%E4%B8%BB%E6%9C%BA%E6%94%92%E6%9C%BA%E5%B0%8F%E8%AE%B0)显卡驱动生效。



二、cuda安装

这里选择了Ubuntu16.04系统runfile安装方案，千万不要选择deb方案，前方无数坑：

1. 下载cuda\_8.0.27\_linux.run和cudnn-8.0-linux-x64-v5.1.tgz

这里我提供了[百度网盘](http://pan.baidu.com/s/1i515khB)，

### 安装cuda8.0

Cuda可以多个版本共存的！

Cuda8.0.61 含有opencl2.0 cuda8.0.27含有opencl1.2。

Lightgbm要1.2, mxnet需要最新的cuda8才不报错。所以我就安装了两个cuda。61为默认。



下载的“cuda\_8.0.27\_linux.run”有1.4G，按照Nivdia官方给出的方法安装CUDA8：

chmod u+x ./cuda\_8.0.27\_linux.run

sudo sh ./cuda\_8.0.27\_linux.run --tmpdir=/tmp --override

cd 下载/   
//sh cuda\_8.0.27\_linux.run --override

//sudo sh cuda\_8.0.27\_linux.run --tmpdir=/opt/temp/

chmod u+x ./cuda\_8.0.27\_linux.run

sudo ./cuda\_8.0.27\_linux.run --tmpdir=/tmp

你以为你会成功安装吗？并不是，你一定会碰到一个错误，Installation Failed. Using unsupported Compiler. ，这是因为 Ubuntu 16.04 默认的 GCC 5.4 对于 CUDA 8.x来说过于新了，CUDA 安装脚本还不能识别新版本的 GCC。

看了一下安装日志，解决方案也很简单，加一个 --override 选项，

sudo ./cuda\_8.0.27\_linux.run --tmpdir=/tmp --override

作者：灵魂机器  
链接：https://zhuanlan.zhihu.com/p/22635699  
来源：知乎  
著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

这里加了–tmpdir主要是直接运行“sudo sh cuda\_8.0.27\_linux.run”会提示空间不足的错误，其实是全新的电脑主机，硬盘足够大的，google了以下发现加个tmpdir就可以了：

Not enough space on parition mounted at /.  
Need 5091561472 bytes.

Disk space check has failed. Installation cannot continue.

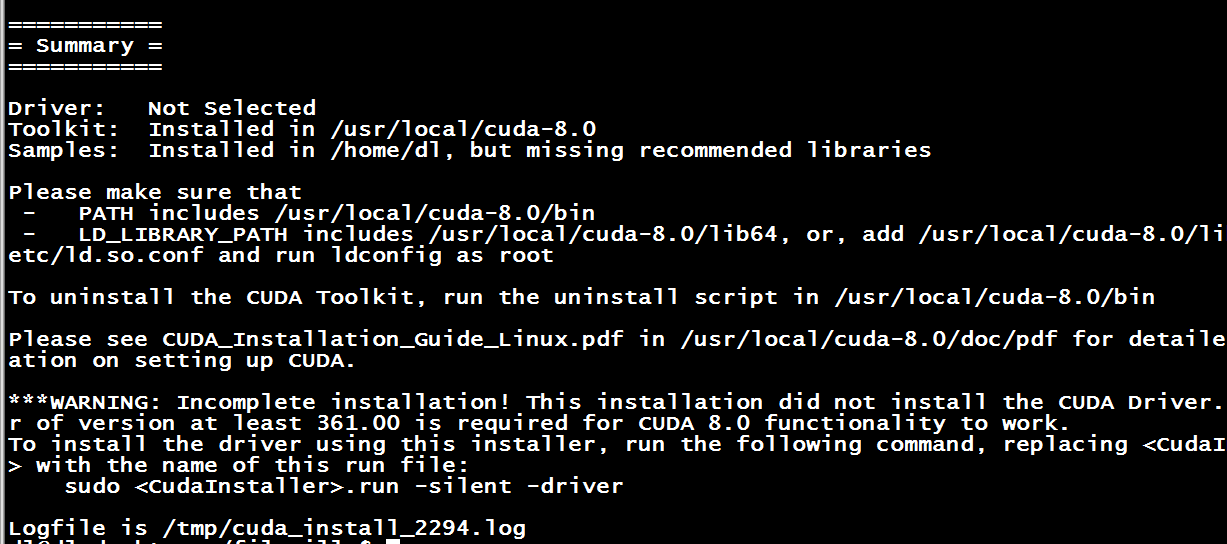
一大段文档要阅读完

执行后会有一系列提示让你确认，非常非常非常非常关键的地方是是否安装361这个低版本的驱动：

Install NVIDIA Accelerated Graphics Driver for Linux-x86\_64 361.62?

答案必须是n，否则之前安装的GTX1080驱动就白费了，而且问题多多。

启动安装程序，一直按空格到最后，输入accept接受条款   
输入n不安装nvidia图像驱动，之前已经安装过了   
输入y安装cuda 8.0工具   
回车确认cuda默认安装路径：/usr/local/cuda-8.0   
输入y用sudo权限运行安装，输入密码   
输入y或者n安装或者不安装指向/usr/local/cuda的符号链接   
输入y安装CUDA 8.0 Samples，以便后面测试   
回车确认CUDA 8.0 Samples默认安装路径：/home/pawn（pawn是我的用户名），该安装路径测试完可以删除



#### 安装完毕后，再声明一下环境变量，并将其写入到 ~/.bashrc 的尾部:

export PATH=/usr/local/cuda-8.0/bin${PATH:+:${PATH}}  
export LD\_LIBRARY\_PATH=/usr/local/cuda-8.0/lib64${LD\_LIBRARY\_PATH:+:${LD\_LIBRARY\_PATH}}

最后再来测试一下CUDA，运行：

nvidia-smi

#### 再来试几个CUDA例子：

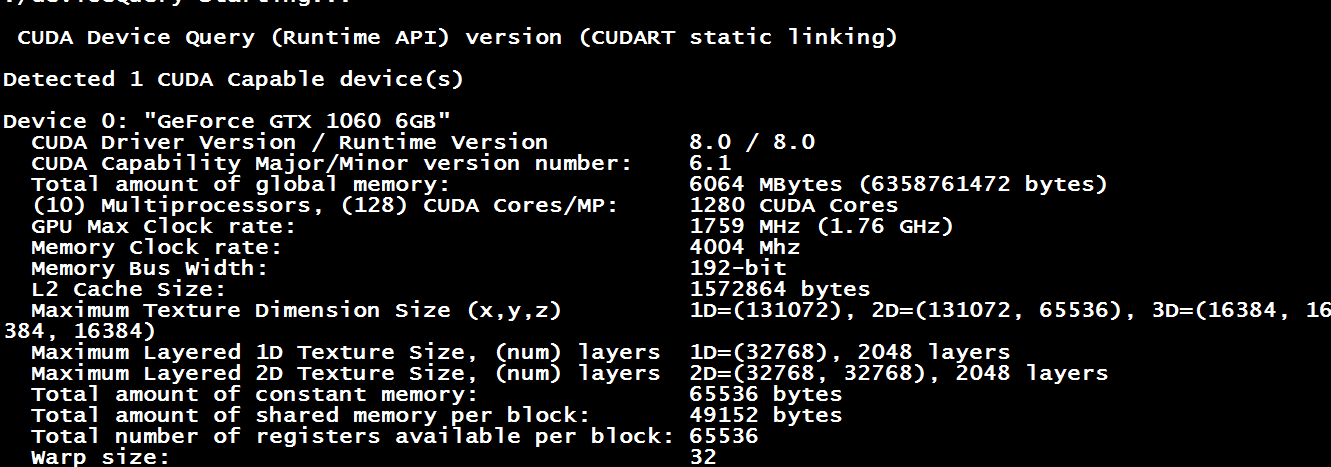
##### 1-devicequery

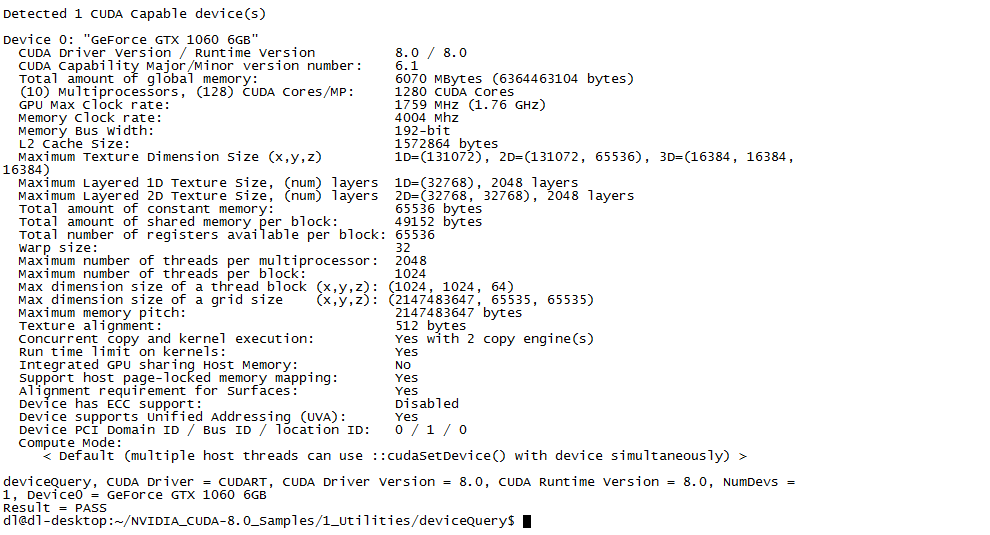
cd ~/NVIDIA\_CUDA-8.0\_Samples/1\_Utilities/deviceQuery

make

./deviceQuery

执行 ./deviceQuery，得到：





##### 2-再测试试一下nobody：

cd ~/NVIDIA\_CUDA-8.0\_Samples/5\_Simulations/nbody/

make

###### 报错

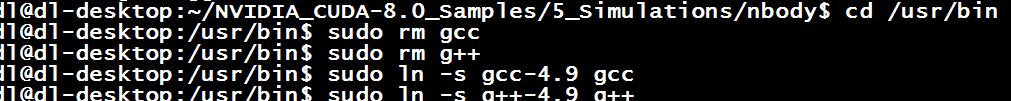
##### unsupported GNU version! gcc versions later than 5.3 are not supported!

以gcc4.9与g++4.9为例

安装低版本gcc与g++：

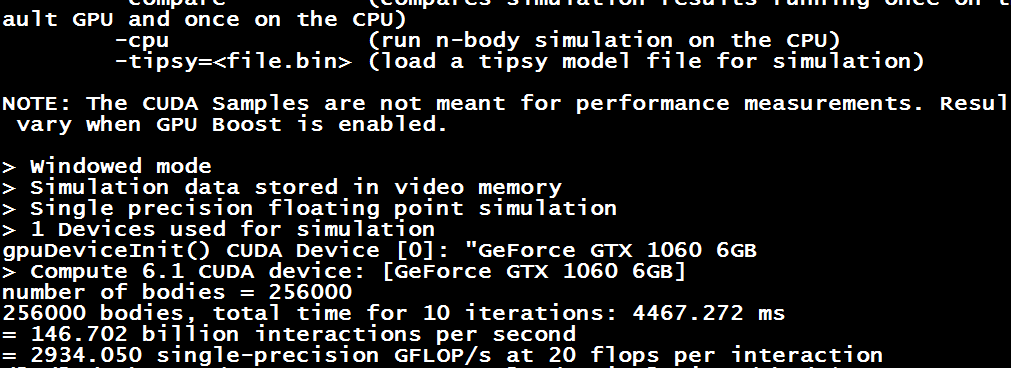
sudo apt-get install gcc-4.9 g++-4.9

之后进入/usr/bin:   
cd /usr/bin  
先删除和gcc5.0关联的gcc:   
sudo rm gcc   
sudo rm g++  
再建个软连接   
sudo ln -s gcc-4.9 gcc   
sudo ln -s g++-4.9 g++



执行：

./nbody -benchmark -numbodies=256000 -device=0

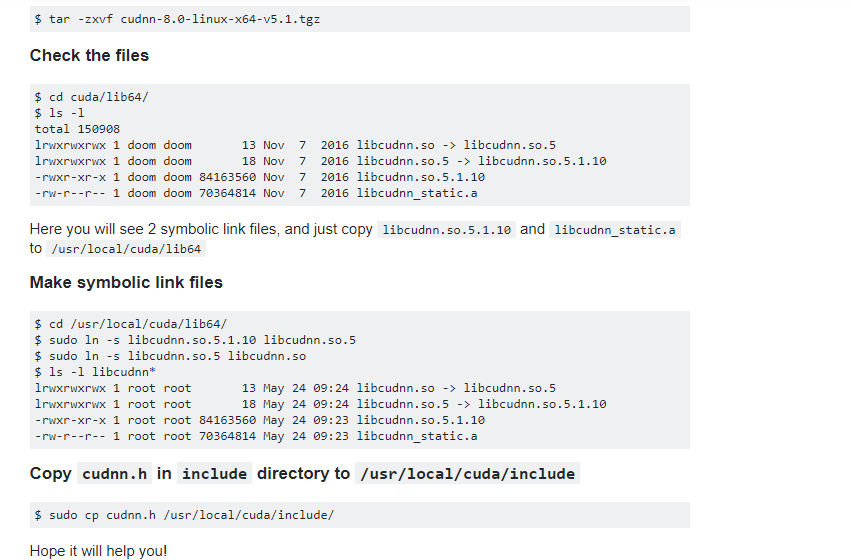


##### 3-至此，说明 CUDA 8.x 安装成功了

### 安装cuDNN 5.1

开始的时候我没装好,后来照着这个重新弄好了

Tf-gpu版里也有



[cuDNN](https://developer.nvidia.com/cudnn" \t "_blank)全称 CUDA Deep Neural Network library，是NVIDIA专门针对深度神经网络设计的一套GPU计算加速库，被广泛用于各种深度学习框架，例如Caffe, TensorFlow, Theano, Torch, CNTK等。

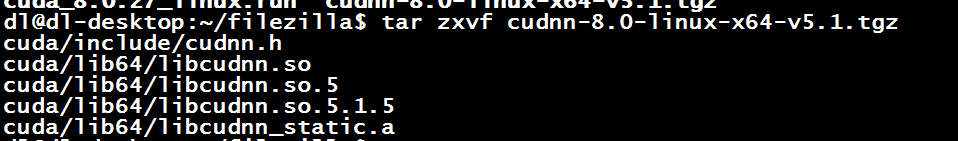
tar -xzvf cudnn-8.0-linux-x64-v5.1-tgz

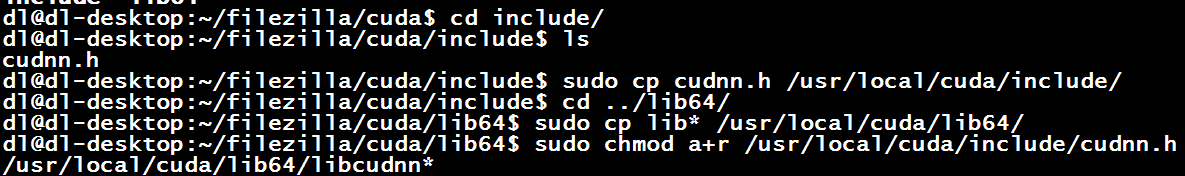
cd cudnn

sudo cp lib\* /usr/local/cuda/lib64/

sudo cp cudnn.h /usr/local/cuda/include/

cd 下载/   
tar zxvf cudnn-8.0-linux-x64-v5.1.tgz   
解压在下载目录下产生一个cuda目录   
cd cuda/include/   
sudo cp cudnn.h /usr/local/cuda/include/ 复制头文件   
cd ../lib64 打开lib64目录   
sudo cp lib\* /usr/local/cuda/lib64/ 复制库文件   
sudo chmod a+r /usr/local/cuda/include/cudnn.h

/usr/local/cuda/lib64/libcudnn\*给所有用户增加这些文件的读权限 



### XGBoost-GPU版本

[http://xgboost.readthedocs.io/en/latest/gpu/index.html#](http://xgboost.readthedocs.io/en/latest/gpu/index.html)

### LightGBM-GPU版本

<https://github.com/huanzhang12/lightgbm-gpu>

参照这个安装

<https://qiita.com/so1_tsuda/items/83a58c93f15afdc34d00>

git clone --recursive https://github.com/Microsoft/LightGBM

cd ./LightGBM

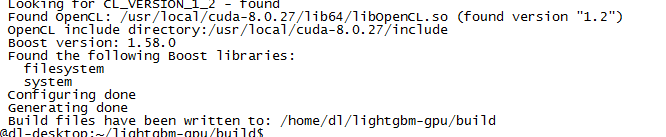
mkdir build; cd build

sudo cmake -DUSE\_GPU**=**1 -DOpenCL\_LIBRARY**=**/usr/local/cuda-8.0.27/lib64/libOpenCL.so -DOpenCL\_INCLUDE\_DIR**=**/usr/local/cuda-8.0.27/include/ ..

sudo make -j**$(**nproc**)**

cd ../python-package/

python setup.py install



Core dumped报错

<https://github.com/Microsoft/LightGBM/issues/526>

<https://github.com/Microsoft/LightGBM/blob/master/docs/GPU-Tutorial.md>

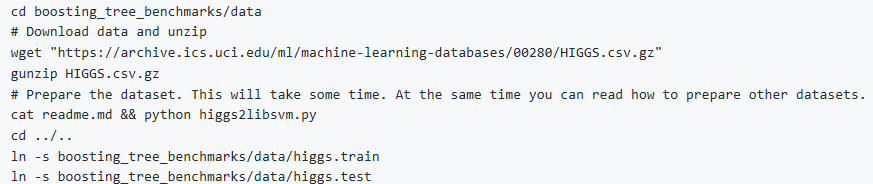
这个是gpu教程，可以参照这个学习

测试gpu的时候，这个数据集也太大了吧！

我要把iter次数改小一点

sudo cmake -DUSE\_GPU=1 -DOpenCL\_LIBRARY=/usr/local/cuda-8.0/lib64/libOpenCL.so -DOpenCL\_INCLUDE\_ （后面这可能有遗漏）

备份系统的时候，我先把这个数据删掉了

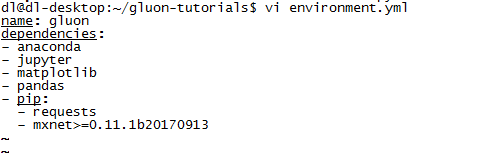


### Mxnet-GPU版本

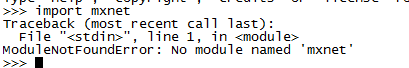
#### 先安装CPU版

<https://zh.gluon.ai/install.html>

官网上用的单独一个环境安装的，我自己把python改成了anaconda



我再自己在本机环境安装



开始的时候不好使，那是肯定的

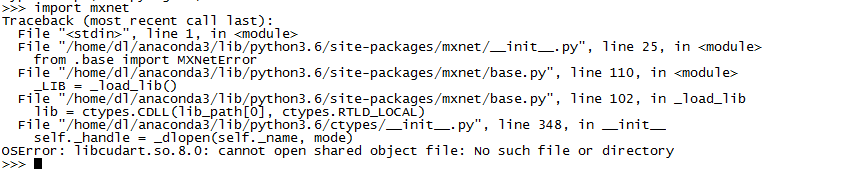
Pip install mxnet

#### 卸载CPU版，安装GPU版本

开始的时候这块总是有报错，后来cuda升级到cuda8.0.61版本就好了

pip install --pre mxnet-cu80 -i https://pypi.douban.com/simple # CUDA 8.0

报错



[https://zh.gluon.ai/install.html#gpu](https://zh.gluon.ai/install.html" \l "gpu)

### Tensorflow-GPU

文档

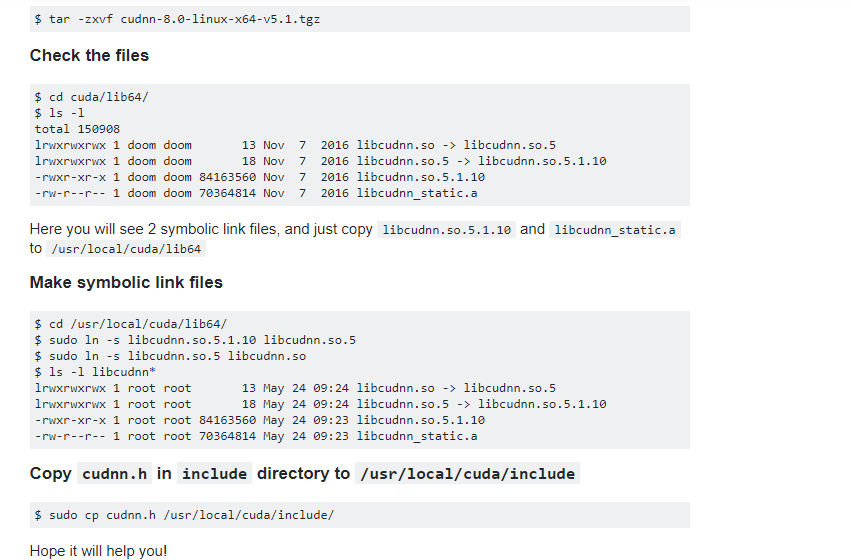
<http://devdocs.io/tensorflow/>

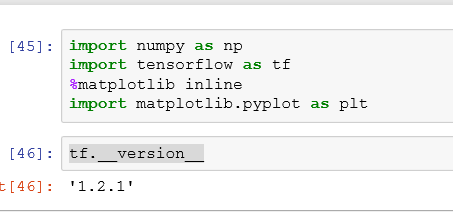
Pip install tensorflow-gpu==1.2

Cudnn报错

<https://stackoverflow.com/questions/42013316/after-building-tensorflow-from-source-seeing-libcudart-so-and-libcudnn-errors>

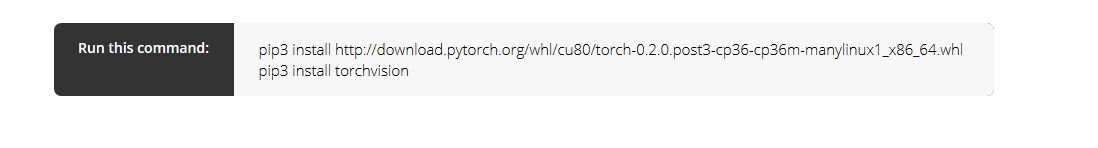
照着这个最后重新安装cudnn





### Pytorch-GPU

官网 pip3改成pip



附一段简单测试CUDA与cuDNN是否工作正常的代码：

# CUDA TEST

import torch

x = torch.Tensor([1.0])

xx = x.cuda()

print(xx)

# CUDNN TEST

from torch.backends import cudnn

print(cudnn.is\_acceptable(xx))

### FastText

Pip install fasttext

### VirtualBox

apt-get install virtualbox

### Teamviewer

<http://blog.csdn.net/zuochao_2013/article/details/53671245>

依赖问题

sudo apt-get -f install

#### Teamvieawer固定密码，无需密码

<https://jingyan.baidu.com/article/870c6fc3252591b03fe4be2b.html>

#### Teamvieawer开机启动

### 在虚拟机win10中安装花生壳，桥接

### Ssh链接jupyter notebook

#### 方法一

<http://yueshi.me/?p=297>

jupyter notebook

ssh [dl@192.168.115.3](mailto:dl@192.168.115.3) –L localhost:8889:localhost:8888

ssh username@host\_address -N -L localhost:8889:localhost:8888

#### 方法三配置法

### Pycharm 并且配置anaconda解析器

### remastersys

<http://blog.csdn.net/t5131828/article/details/53258116>

若有报错

<https://askubuntu.com/questions/517857/dpkg-error-processing-package-linux-image-generic-configure-dependency-pro>

我备份的时候把虚拟机的win10 exclude了和lightgpu那个数据删掉了，防止太大

### 虚拟机和本机共享文件夹-死活没成功

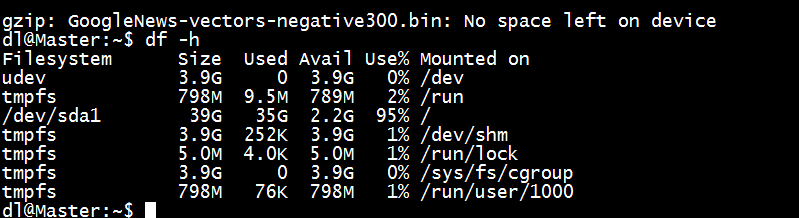
所以就需要gui的时候，就用虚拟机全屏看就好了

### 数据文件夹软连接

Ln -s

Vmware 最小化快捷键

### 虚拟机空间不足



方法一：挂载法

<http://www.xuebuyuan.com/2037759.html>

上面这种我一直 没有弄好

方法二：gparted法，真正的扩展主分区

<http://www.linuxidc.com/Linux/2015-08/121674.htm>

<http://blog.csdn.net/hugewaves/article/details/52276549>

<http://www.linuxidc.com/Linux/2015-08/121674.htm>

### 远程开机，必须有一台同局域网内的机器

<http://blog.163.com/xs_fantasy/blog/static/1015608712011104104926797/>

### jupyter notebook主题

要参照官网上的链接来操作

<http://blog.csdn.net/Techmonster/article/details/73382535>

最后要加 –T 否则没有toolbar

其他类似于全屏的选项

### Jupyter notebook插件

Pip install

 jupyter\_contrib\_nbextensions

win上

<https://github.com/ipython-contrib/jupyter_contrib_nbextensions>

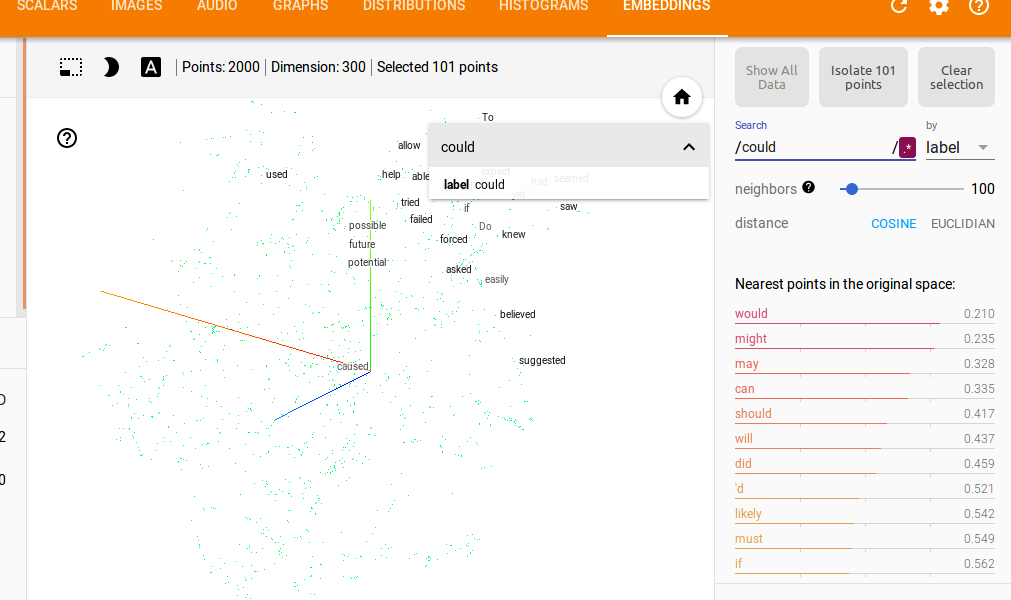
git clone 外加解决一个报错 才安装成功

[http://www.lfd.uci.edu/~gohlke/pythonlibs/#lxml](http://www.lfd.uci.edu/~gohlke/pythonlibs/" \l "lxml) 用的这个解决的报错

### Word2vec可视化

如果提示了 metadata.tsv is not a file 这个错误，   
那么，去 minimalsample 文件夹下会找到一个 projector\_config.pbtxt 文件，把里面的 metadata\_path: 和 image\_path: 改为你的 metadata.tsv 和 mnistdigits.png 所在的绝对路径。

搜索一下才能出词



### 临时添加sys.path

sys.path.append(path)

sys.path.append('..') #用于python import时的临时路径

### win10 cmd复制粘贴

enter 和 右键

### Ubuntu每次启动都显示System program problem detected的解决办法

每次启动Ubuntu都显示System program problem detected，很烦人，不过修改一个文件就可以。

      执行命令：

**[plain]** [view plain](http://blog.csdn.net/invincible7/article/details/18702733" \o "view plain) [copy](http://blog.csdn.net/invincible7/article/details/18702733" \o "copy)

1. sudo gedit /etc/default/apport

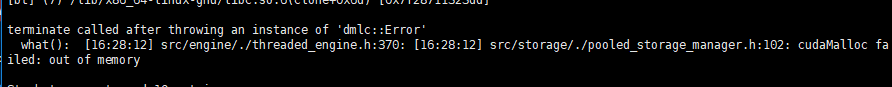
      将enabled=1改为enabled=0保存退出即可

## 终端复制粘贴

快捷键。一般通用的是Ctrl+C与Ctrl+V。不过通用也是有限制的，一般的程序下是没有问题，遇到终端就不行了。其实终端下默认的是 Ctrl+Shift+C，Ctrl+Shift+V，可以自己在编辑项下面自己设置为常用的。

# 报错

## 显存不足



terminate called after throwing an instance of 'std::bad\_alloc'

what(): std::bad\_alloc

动态显示显卡

watch -n 1 -d nvidia-smi