

Ubuntu 16.04安装CUDA与cuDNN库

Ubuntu 16.04安装CUDA与cuDNN库

[概述](#)

[环境准备](#)

[安装步骤](#)

[安装显卡驱动](#)

[安装CUDA](#)

[C程序测试CUDA的安装是否成功](#)

[安装cuDNN](#)

[参考资料](#)

概述

CUDA是nvidia公司为自己GPU专门推出的并程序设计语言，语法结构与C语言类似，CUDA能够让我们充分地利用GPU的计算能力，提高程序性能，但是并程序设计与计算机硬件有极强的关联性。

环境准备

本文将在安装了nvidia tesla P40 GPU的高性能HP服务器中进行。下面给出了详细的系统配置。

项目	版本
操作系统	Ubuntu 16.04.5
gcc	gcc 5.4.0
CUDA	cuda 9.2
cuDNN	cuDNN for CUDA 9.2
内核版本	4.15.0-34-generic

注意：低版本的内核不支持安装CUDA，因此要尽量使用Ubuntu 16.04之后的操作系统。

安装步骤

安装显卡驱动

```
1 sudo apt-get update    #更新系统
2 sudo apt-get install build-essential
3 sudo apt-get update
4 sudo apt-get upgrade    #保持当前大版本，将小版本升级到最新
5 lspci |grep -i nvidia    #查看系统中安装的显卡版本，因为tesla是没有视频输出功能的GPU因此使用该命令查看
6 nvidia-smi    #查看显卡的信息，如下面两张图图，我们能够清楚的看到，系统中安装了两张显卡，分别是Quadro P400 （视频接口卡）和P40 （并行计算卡），在我们后续的实验中，将使用P40来做并行计算
7 uname -r    #查看内核版本
8 sudo apt-get install linux-headers-$(uname -r)    #安装内核头文件
9
10 ubuntu-drivers devices    #查看GPU版本以及对应的GPU驱动
11 sudo service lightdm stop    #在安装驱动时，关闭图形界面
12 sudo ubuntu-drivers autoinstall    #安装上述命令中查看到的所有驱动
13 sudo service lightdm restart
14 sudo apt-get update
15 sudo apt-get upgrade
16 sudo apt-get install mesa-utils    #安装查看是否安装了正确的GPU驱动
17 glxinfo | grep rendering    #如果返回信息为YES，则说明已经正确的安装了显卡驱动，但是如果为NO，则说明驱动没有正确的安装
18 nvidia-smi -q -i 1    #查看第一块GPU的详细信息，我们能够看到究竟第1块GPU的型号，这对于在运行Python程序时指定在某块具体的GPU上运行有极其重要的影响
19 nvidia-smi -q -i 0    #查看第一块GPU的详细信息，我们能够看到究竟第0块GPU的型号，这对于在运行Python程序时指定在某块具体的GPU上运行有极其重要的影响
```

更新GPU驱动之前，驱动的版本为：384.130

```
graph@graph-HP-Z8-G4-Workstation:~$ nvidia-smi
Tue Sep 11 14:16:06 2018
```

NVIDIA-SMI 384.130		Driver Version: 384.130					
GPU	Name	Persistence-M	Bus-Id	Disp.A	Volatile	Uncorr.	ECC
Fan	Temp	Perf	Pwr:Usage/Cap	Memory-Usage	GPU-Util	Compute	M.
0	Quadro P400	Off	00000000:2D:00.0	On			N/A
34%	45C	P8	ERR! / N/A	110MiB / 1990MiB	0%		Default
1	Tesla P40	Off	00000000:8D:00.0	Off			Off
N/A	35C	P8	10W / 250W	0MiB / 24445MiB	0%		Default

Processes:				GPU Memory
GPU	PID	Type	Process name	Usage
0	1432	G	/usr/lib/xorg/Xorg	108MiB

更新GPU驱动之后，驱动版本为：396.44

```
graph@graph-HP-Z8-G4-Workstation:~$ nvidia-smi
Tue Sep 11 14:32:36 2018
```

NVIDIA-SMI 396.44		Driver Version: 396.44					
GPU	Name	Persistence-M	Bus-Id	Disp.A	Volatile	Uncorr.	ECC
Fan	Temp	Perf	Pwr:Usage/Cap	Memory-Usage	GPU-Util	Compute	M.
0	Quadro P400	Off	00000000:2D:00.0	On			N/A
35%	48C	P8	N/A / N/A	110MiB / 1991MiB	0%		Default
1	Tesla P40	Off	00000000:8D:00.0	Off			Off
N/A	41C	P8	11W / 250W	0MiB / 24451MiB	0%		Default

Processes:				GPU Memory
GPU	PID	Type	Process name	Usage
0	1430	G	/usr/lib/xorg/Xorg	107MiB

安装CUDA

在官网中选择后执行下面的命令: [https://developer.nvidia.com/cuda-downloads?](https://developer.nvidia.com/cuda-downloads?target_os=Linux&target_arch=x86_64&target_distro=Ubuntu&target_version=1604&target_type=debnetwork)

[target_os=Linux&target_arch=x86_64&target_distro=Ubuntu&target_version=1604&target_type=debnetwork](https://developer.nvidia.com/cuda-downloads?target_os=Linux&target_arch=x86_64&target_distro=Ubuntu&target_version=1604&target_type=debnetwork)

- 1 `sudo dpkg -i cuda-repo-ubuntu1604_9.2.148-1_amd64.deb`
- 2 `sudo apt-key adv --fetch-keys`
http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86_64/7fa2af80.pub

```
3 sudo apt-get update
4 sudo apt-get install cuda
```

安装完成之后，我们能够看到在/usr/local下面有对应版本的cuda文件夹及其软连接，此时我们需要在环境变量~/.bashrc中添加 `PATH=$PATH:/usr/local/cuda/bin` 和 `LD_LIBRARY_PATH=/usr/local/cuda/lib64`。

C程序测试CUDA的安装是否成功

在上一个阶段，我们已经成功的安装CUDA，但是还不知道安装是否成功，当然在安装的过程中，没有报错已经能够表明安装成功，但是仍然不够，下面，我们cuda/sample下自带的code示例测试是否成功安装。

```
1 cd /usr/local/cuda/samples/1_Uutilities/deviceQuery
2 sudo make
3 sudo ./deviceQuery      #在输出中我们能够看到GPU的型号，像下面图示，说明成功安装CUDA支持
4 nvcc --version          #nvcc是CUDA程序的编译器，如果能够正确的返回nvcc的版本，也充分说明CUDA已经成功安装
```

```
graph@graph-HP-Z8-G4-Workstation:/usr/local/cuda/samples/1_Uutilities/deviceQuery$ sudo ./deviceQuery
./deviceQuery Starting...

  CUDA Device Query (Runtime API) version (CUDA static linking)

Detected 2 CUDA Capable device(s)
Device 0: "Tesla P40"
  CUDA Driver Version / Runtime Version          9.2 / 9.2
  CUDA Capability Major/Minor version number:    6.1
  Total amount of global memory:                 24452 MBytes (25639649280 bytes)
  (30) Multiprocessors, (128) CUDA Cores/MP:     3840 CUDA Cores
  GPU Max Clock rate:                           1531 Mhz (1.53 GHz)
  Memory Clock rate:                             3615 Mhz
  Memory Bus Width:                              384-bit
  L2 Cache Size:                                 3145728 bytes
  Maximum Texture Dimension Size (x,y,z)         1D=(131072), 2D=(131072, 65536), 3D=(16384, 16384, 16384)
  Maximum Layered 1D Texture Size, (num) layers  1D=(32768), 2048 layers
  Maximum Layered 2D Texture Size, (num) layers  2D=(32768, 32768), 2048 layers
  Total amount of constant memory:               65536 bytes
  Total amount of shared memory per block:       49152 bytes
  Total number of registers available per block: 65536
  Warp size:                                     32
  Maximum number of threads per multiprocessor:  2048
```

安装cuDNN

我们能够在 <https://developer.nvidia.com/cuda-gpus> 上查找到关于Tesla p40的compute Capability为6.1，因此，为了加速神经网络的计算，应该安装cuDNN库来加速神经网络的计算过程。

```
1 #下载的安装包为: cudnn-9.2-linux-x64-v7.2.1.38.tgz
2 tar -xvf cudnn-9.2-linux-x64-v7.2.1.38.tgz #解压后在下载目录下能够看
   到一个名为cuda的文件夹
3 cd cuda
4 sudo cp lib64/lib* /usr/local/cuda/lib64
5 sudo include/cudnn.h /usr/local/cuda/include
6 cd /usr/local/cuda/lib64/
7 sudo chmod +r libcudnn.so.7.2.1
8 sudo ln -sf libcudnn.so.7.2.1 libcudnn.so.7
9 sudo ln -sf libcudnn.so.7 libcudnn.so
10 sudo ldconfig
```

至此完成cuDNN库的安装，它在计算能力3以上的计算机中，能够加速神经网络的计算。

参考资料

1. <https://askubuntu.com/questions/700208/ubuntu-14-04-direct-rendering-not-working-with-nvidia>
2. https://blog.csdn.net/A_Z666666/article/details/72853346
3. Python程序指定使用某个GPU执行程序
<https://blog.csdn.net/ccbrid/article/details/79761674>
4. 详细介绍了CUDA的初次使用步骤
https://blog.csdn.net/xsc_c/article/details/24250345