linux安装cuda教程

1. 查看是否安装NVIDIA显卡

   `lspci | grep -i vga`

   `lspci | grep -i nvidia` （推荐）

   查看系统型号： `uname -a`

2. 查看显卡信息

   `nvidia-smi` 或者 `nvidia-detector` (输入后，若显示内容，表示已经安装好显卡驱动，若显示无发现，则需要手动安装显卡驱动)

3. 查看可安装驱动

   `ubuntu-drivers devices`

   找到最适合的驱动安装，安装recommended标记的，通常也是数字版本最大的那个。

   查询显卡驱动型号根据显卡信息,通过网址 `https://www.nvidia.com/Download/index.aspx?lang=en-us` 或

   是 `https://www.nvidia.cn/geforce/drivers/`

   英伟达官方的cuda和驱动的对应： `https://docs.nvidia.com/cuda/cuda-toolkit-release-notes/index.html`

   若需要删除以前安装的NVIDIA驱动版本，可用命令 `sudo apt purge nvidia*` 和 `sudo apt autoremove`

   查看gcc版本， `gcc --version`

4. 关闭系统自带驱动nouveau

   nouveau 驱动是 Ubuntu 默认的开源显卡驱动，与 Nvidia 显卡驱动一起使用会导致兼容性问题

   通过 `lsmod | grep nouveau` ，查看nouveau驱动的启用情况，如果有输出表示nouveau驱动正在工作，如果没有内容输出则表示已经禁用了nouveau。

   输出内容如下：

   ```
   nouveau              1863680  9
   video                  49152  1 nouveau
   ttm                   102400  1 nouveau
   mxm_wmi                16384  1 nouveau
   drm_kms_helper        180224  1 nouveau
   drm                   479232  12 drm_kms_helper,ttm,nouveau
   i2c_algo_bit           16384  2 igb,nouveau
   wmi                    28672  4 intel_wmi_thunderbolt,wmi_bmof,mxm_wmi,nouveau
   ```

   有输出，表示nouveau启动了，下面进行nouveau的禁用

   方式一（推荐）：

   在终端输入 `sudo gedit /etc/modprobe.d/blacklist.conf` ,弹出blacklist.conf文件,在blacklist.conf文件末尾加上这两行，并保存：

   ```
   blacklist nouveau
   options nouveau modeset=0
   ```

   方式二：

   ```
   sudo bash -c "echo blacklist nouveau > /etc/modprobe.d/blacklist-nvidia-nouveau.conf"
   sudo bash -c "echo options nouveau modeset=0 >> /etc/modprobe.d/blacklist-nvidia-nouveau.conf"

   cat /etc/modprobe.d/blacklist-nvidia-nouveau.conf
   blacklist nouveau
   options nouveau modeset=0
   ```

   运行完方式一或方式二，然后在终端中输入

   `sudo update-initramfs -u  #应用更改`

   `sudo reboot  #重启`

   `lsmod | grep nouveau #再次查询，应该就无输出了`

   重启，就禁止了ubuntu20.04自带的nouveau显卡驱动了，接下来就可以安装NVIDIA版本的驱动程序了

5. 安装驱动

   方式一：指定版本

   `sudo apt install nvidia-driver-XXX`

方式二：自动安装系统推荐版本

```
sudo ubuntu-drivers autoinstall
```

方式三：系统设置->软件和更新（software and update）->附加驱动->选择NVIDIA驱动->应用更改（推荐）

如果没有遇到报错，说明安装成功，此时调用nvidia-smi指令可能还是看不到显卡信息，不要担心，重启系统之后就能看到了

6. CUDA是查看的是显卡最大支持的CUDA版本号。因此仍然需要手动从官网下载CUDA，且版本号不能高于这个。除了GPU支持最大CUDA版本外，还需要考虑Tensorflow对应的CUDA版本。

   ```
   https://www.tensorflow.org/install/source_windows
   ```

7. 进入NVIDIA官网下载所需的对应版本，网址 `https://developer.nvidia.com/cuda-toolkit-archive`
8. 选择版本号进行安装

   在进行安装cuda之前，需要安装一些相互依赖的库文件

   ```
   sudo apt-get install freeglut3-dev build-essential libx11-dev libxmu-dev libxi-dev libgl1-mesa-glx libglu1-mesa libglu1-mesa-dev
   ```

   接着进行如下方式安装

   - 方式一：选择deb(local)格式的CUDA文件下载Download Installer for Linux Ubuntu 22.04 x86_64,选择Linux→x86_64→Ubuntu→20.04→deb(local)安装步骤同官网，以下代码均来自官网，根据自己选择的版本，复制官网代码安装。

     ```
     wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2004/x86_64/cuda-ubuntu2004.pin
     sudo mv cuda-ubuntu2004.pin /etc/apt/preferences.d/cuda-repository-pin-600
     wget https://developer.download.nvidia.com/compute/cuda/11.3.0/local_installers/cuda-repo-ubuntu2004-11-3-local_11.3.0-465.19.01-1_amd64.deb
     sudo dpkg -i cuda-repo-ubuntu2004-11-3-local_11.3.0-465.19.01-1_amd64.deb
     sudo apt-key add /var/cuda-repo-ubuntu2004-11-3-local/7fa2af80.pub
     sudo apt-get update
     sudo apt-get -y install cuda
     ```

   - 方式二（推荐）：选择runfile(local)格式的CUDA文件下载Download Installer for Linux Ubuntu 22.04 x86_64,选择Linux→x86_64→Ubuntu→20.04→runfile(local)安装步骤同官网，以下代码均来自官网，根据自己选择的版本，复制官网代码安装。

     ```
     #使用wget命令下载CUDA工具包的安装文件,下载CUDA 11.0.2版本的安装文件，这个版本需要配合NVIDIA显卡驱动版本为450.51.05使用
     wget https://developer.download.nvidia.com/compute/cuda/11.0.2/local_installers/cuda_11.0.2_450.51.05_linux.run
     #Linux系统中以超级用户权限运行CUDA安装程序。运行这个命令会启动安装程序，根据提示完成安装过程
     sudo sh cuda_11.0.2_450.51.05_linux.run
     ```

     运行方式二指令后，，会弹出界面，点击Continue，然后再输入accept。

     接着，在弹出的界面中通过Enter键，取消Driver和450.51.05的安装，然后点击Install，等待。

9. 上一步骤方式二，第一句话是下载cuda_xxxx_linux.run脚本，第二句是执行脚本。在弹出的安装界面中选“continue”，如果选了会跳出安装，就说明安装失败，给了失败日志的路径，自己查看原因，一般是gcc版本问题，文章最后给出降级方法，不赘述。在这一步输入accept，然后选择安装项，一般情况我们都安装了显卡驱动，所以这里第一项驱动最好不勾选，其他默认安装。
10. 安装成功后，需要配置环境变量

    ```
    sudo vim ~/.bashrc
    ```

11. 在文件最后一行加入以下内容（注意cuda-11.2这个要根据各自版本将版本号修改）

    ```
    export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/cuda-11.2/lib64
    export PATH=$PATH:/usr/local/cuda-11.2/bin
    export CUDA_HOME=$CUDA_HOME:/usr/local/cuda-11.2
    ```

    or

    ```
    export PATH=/usr/local/cuda-11.2/bin${PATH:+:${PATH}}
    export LD_LIBRARY_PATH=/usr/local/cuda-11.2/lib64\
    {LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}


    export PATH=$PATH:/usr/local/cuda/bin
    export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/cuda/lib64
    export LIBRARY_PATH=$LIBRARY_PATH:/usr/local/cuda/lib64


    export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/cuda/lib64
    export PATH=$PATH:/usr/local/cuda/bin
    export CUDA_HOME=$CUDA_HOME:/usr/local/cuda
    ```

```
# user-add: cuda
export PATH=/usr/local/cuda-11.6/bin:$PATH
export LD_LIBRARY_PATH=/usr/local/cuda-11.6/lib64:$LD_LIBRARY_PATH
export CUDA_HOME=/usr/local/cuda
```

12. 激活命令

    `source ~/.bashrc`

13. 检查一下CUDA是否安装正确：（注意文件夹路径一定要存在，需要跟上面设置环境变量的路径保持一致）
    - 方式一： `cat /usr/local/cuda/version.txt`
    - 方式二： `nvcc -V` or `nvcc --version`
    - 方式三（推荐）：进入/usr/local/cuda-11.2/samples/1_Utilities/deviceQuery

      ```
      sudo make

      ./deviceQuery
      #运行显示内容最后，会有result=pass表示安装成功
      ```

    - 方式三可能出现如下错误：make 中提示如下错误

      ```
      make[1]: Leaving directory '/home/dell/Documents/cuda-samples/Samples/0_Introduction/vectorAddMMAP'
      make[1]: Entering directory '/home/dell/Documents/cuda-samples/Samples/0_Introduction/simpleMPI'
      /opt/anaconda3/bin/mpicxx -I../../../Common    -o simpleMPI_mpi.o -c simpleMPI.cpp
      /opt/anaconda3/bin/mpicxx: line 299: x86_64-conda_cos6-linux-gnu-c++: command not found
      make[1]: *** [Makefile:389: simpleMPI_mpi.o] Error 127
      make[1]: Leaving directory '/home/dell/Documents/cuda-samples/Samples/0_Introduction/simpleMPI'
      make: *** [Makefile:45: Samples/0_Introduction/simpleMPI/Makefile.ph_build] Error 2
      ```

    可能是没安装 gxx_linux-64

    ```
    sudo -i
    conda install gxx_linux-64
    apt install g++-aarch64-linux-gnu
    make clean
    make
    ```

14. 安装cudnn，选择对应cuda版本号的cudnn

    `https://developer.nvidia.com/cudnn` or `https://developer.nvidia.com/rdp/cudnn-archive`

15. 下载对应版本压缩包
    - 解压命令 `tar -zxvf cudnn-11.2-linux-x64-v8.1.1.33.tgz -C .`
      进入到相应目录，运行以下命令

      ```
      cp cuda/lib64/* /usr/local/cuda-11.0/lib64/
      cp cuda/include/* /usr/local/cuda-11.0/include/



      sudo cp cuda/include/cudnn.h /usr/local/cuda-11.2/include
      sudo cp cuda/lib64/libcudnn* /usr/local/cuda-11.2/lib64
      sudo chmod a+r /usr/local/cuda-11.2/include/cudnn.h
      sudo chmod a+r /usr/local/cuda-11.2/lib64/libcudnn*
      ```

    - 下载对应版本压缩包，拷贝文件到指定目录，给予权限

      ```
      sudo cp -d cuda/include/cudnn.h /usr/local/cuda-11.2/include
      sudo cp -d cuda/lib64/libcudnn* /usr/local/cuda-11.2/lib64
      sudo chmod a+r /usr/local/cuda-11.2/include/cudnn.h
      sudo chmod a+r /usr/local/cuda-11.2/lib64/libcudnn*
      ```

16. 查看cuDNN版本

    `cat /usr/local/cuda/include/cudnn_version.h | grep CUDNN_MAJOR -A 2`

    会显示如下内容

```
#define CUDNN_MAJOR 8
#define CUDNN_MINOR 3
#define CUDNN_PATCHLEVEL 3
--
#define CUDNN_VERSION (CUDNN_MAJOR * 1000 + CUDNN_MINOR * 100 + CUDNN_PATCHLEVEL)

#endif /* CUDNN_VERSION_H */
```

17. cuDNN的检测

    进入网址 `https://developer.nvidia.com/rdp/cudnn-download` ,下载三个.deb格式的检测文件

    终端输入如下命令安装下载的三个.deb格式的检测文件

```
dpkg -i libcudnn8-dev_8.0.5.39-1+cuda11.0_amd64.deb
dpkg -i libcudnn8-samples_8.0.5.39-1+cuda11.0_amd64.deb
dpkg -i libcudnn8_8.0.5.39-1+cuda11.0_amd64.deb
```

    通过上面三条指令，cuDNN的测试文件会自动安装在系统的/usr/src/cudnn_samples_v8文件夹下，进入mnistCUDNN下，执行命令make clean && make。

18. 可以执行相关训练文件查看是否有gpu信息输出，或监控一下gpu状态

```
watch -n 1 nvidia-smi
```

19. 卸载cuda

    - 方式一

```
#进入安装目录
cd  /usr/local/cuda-11.2/bin
#执行cuda自带的卸载程序
sudo ./cuda-uninstaller
#输入命令后，弹出如下界面，通过回车键选中三个选项，最后选中Done。执行完下面指令后，上面的cuda文件就删除了。
#最后在终端输入如下指令
sudo rm -rf /usr/local/cuda-11.2
```

    - 方式二（若方式一找不到cuda-uninstaller可以用以下方法）

```
#从系统中删除CUDA软件包
sudo apt-get remove cuda
#删除不再需要的软件包和依赖项
sudo apt autoremove
#删除与CUDA相关的所有软件包
sudo apt-get remove cuda*
#进入/usr/local/目录
cd /usr/local/
#删除CUDA 11.4版本的安装文件夹
sudo rm -r cuda-11.4

#下面命令查看卸载残留和删除，即可正常安装
#列出所有已安装的CUDA软件包。能够找到要删除的软件包的确切名称
sudo dpkg -l |grep cuda
#删除名为cuda-visual-tools-11-4的软件包。请注意，这个命令将完全删除该软件包，包括其配置文件和任何相关的依赖项。
sudo dpkg -P cuda-visual-tools-11-4
#残留都要删除哦，通过删除样式为
sudo dpkg -P 残留文件名
```

20. 更改gcc版本

```
sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-7 9
sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-9 1

sudo update-alternatives --display gcc

sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-7 9
sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-9 1

sudo update-alternatives --display g++```
```