Tutorial for CUDA 7.5 Installation
Based on CentOS 7

Xue Shengke [*]

December 28, 2016

1 Prepare CentOS 7 distribution, with Python 2.7.* already installed

Internet connection available is also required.

2 Install NVIDIA graphic driver for Linux system

NVIDIA graphic driver for Linux system will be installed individually before the CUDA installation, so we download the driver according to its actual type and proper operating system (Linux, 64bit / 32bit).
Taking GT 640 as an instance, we download NVIDIA-Linux-x86_64-361.42.run. Open a terminal (better as root),

```
# uname -r
3.10.0-327.el7.x86_64 ; the distribution varies from person to person
    , so remember it
# yum install gcc kernel-devel kernel-headers
```

Add two lines in both files as follows,

```
# vim /etc/modprobe.d/blacklist.conf
# vim /lib/modprobe.d/dist-blacklist.conf
...
blacklist nouveau
options nouveau modeset=0
```

Remake initramfs image,

```
# cp /boot/initramfs-$(uname -r).img /boot/initramfs-$(uname -r).img.
    bak
# dracut /boot/initramfs-$(uname -r).img $(uname -r)
# rm /boot/initramfs-$(uname -r).img.bak
```

Convert to text mode (runlevel 3), i.e., manipulate in command line window.

[*] Contact e-mail: xueshengke@zju.edu.cn

```
# systemctl set-default multi-user.target
# init 3
# reboot    ; this command is not necessary
```

Here, we find that the screen turns into command line with low resolution apparently. Switch to the directory where the driver file (NVIDIA-Linux-x86_64-361.42.run) exists.

```
# cd /to/your/directory/
# ./NVIDIA-Linux-x86_64-361.42.run --kernel-source-path=/usr/src/
   kernels/3.10.0-327.el7.x86_64  -k $(uname -r)
```

Steps as follows:

```
LICENSE ... -accept
Building kernel modules ...
Install NVIDIA's 32-bit compatibility libraries. -yes
...
WARNING: Unable to perform the runtime configuration check for 32-bit
    library ... Assuming successful installation. -ok
Would you like to run the nvidia-xconfig utility to automatically
   update you X configuration file ... -yes -ok
```

Convert back to graphical mode (runlevel 5),

```
# systemctl set-default graphical.target
# init 5
```

At this time, we will see the display shows graphical window with very high definition resolution. Now check graphic driver has been successfully installed in our system.

```
# lspci |grep -i vga
01:00.0 VGA compatible controller: NVIDIA Corporation GK208 [GeForce
   GT 640 Rev. 2] (rev al)
```

3 Download RUN version of CUDA 7.5 toolkit

Download it from its official website. Choose Linux - x86_64 - CentOS - 7 - runfile(local) - Download(1.1 GB).

4 Install full version of CUDA 7.5 toolkit (Toolkit, Samples)

Fisrt, we choose 'no' for Graphics Driver. Caution ! Do not choose 'yes' to install OpenGL, for which may conflicts with GNOME, causing the crash of desktop.

```
# ./cuda_7.5.18_linux.run
...
Install NVIDIA Accelerated Graphics Driver ... -n
Install OpenGL ... -n
Install CUDA 7.5 Toolkit ... -y
Toolkit location /usr/local/cuda-7.5 ... [Enter]
Install a symbolic link at ... -y
Install CUDA 7.5 Samples ... -y
Enter CUDA Samples Location ... [Enter]
...
Finished
Driver : Not Selected
Toolkit : Installed in /usr/local/cuda-7.5
Samples : Installed in /root, but missing recommended libraries
***WARNING: Incomplete installation!
```

Remember the location of directory of NVIDIA_CUDA-7.5_Samples, for we need to check whether our CUDA works below.

5 Add path to environment variables

Edit /etc/profile to update PATH and LD_LIBRARY_PATH. Modify /etc/profile like this,

```
...
export PATH=/usr/local/cuda-7.5/bin:$PATH
export LD_LIBRARY_PATH=/usr/local/cuda-7.5/lib64:$LD_LIBRARY_PATH
# source /etc/profile
```

6 Upgrade build compilers

Update G++/GCC compiler.

```
# yum install build-essential
# gcc -v
# make -v
```

7 Compile CUDA samples and check CUDA driver

```
# cd /to/your/path/NVIDIA_CUDA-7.5_Samples/
# make
```

After a long time compilation, none of errors occur, which suggests CUDA has been successfully installed.

```
# nvcc -V
nvcc : NVIDIA (R) cuda compiler driver

# nvidia-smi
A table containing NVIDIA-SMI: 361.42 Driver Version: 361.42 appears.
GPU Name ...
Processes ...
```

8 Install recommended packages and libraries

Note that pip, nose and OpenBLAS are installed separately through .tar.gz files. They may be downloaded from website such as https://github.com.

```
# yum install numpy, scipy, python-devel, git
# tar -zxf nose-1.3.7.tar.gz
# cd nose-1.3.7/
# python setup.py install

# tar -zxf pip-8.1.2.tar.gz
# cd pip-8.1.2/
# python setup.py install

# tar -zxf  OpenBLAS-0.2.18.tar.gz
# cd OpenBLAS-0.2.18/
# make
```

9 Install Theano

Only a small library the Theano is, but it requires the network with an appropriate status to download it.

```
# pip install Theano
```

10 Make /root/.theanorc file

```
# vim /root/.theanorc
...
[global]
device=gpu
floatX=float32
root=/usr/local/cuda-7.5
[nvcc]
fastmath = True
[blas]
ldflags = -lopenblas
```

Save this file in path /root, and do not move it to other places.

11 Test Theano library

Run testing_theano.py file (accompanied with this tutorial) using python.

```
# sudo python testing_theano.py
...
Using gpu device 0: GeForce GT 640 ...
Looping 1000 times took 0.351630 seconds
```

```
Result is CudaNdarray ...
Numpy result is [...
Used the gpu
```

From the information printed above, we see python is able to use GPU to run the program now.

All complete ! So far We have successfully deployed CUDA 7.5 toolkit in our CentOS 7.