

AMBER16+CUDA Installation in Ubuntu 18.04 (BIONIC)

Author: Senthilkumar Kailasam

Date: Feb 2018 * Here I will go through the steps involved in installing amber16 + Ambertools 17 in Ubuntu 18.04 LTS with cuda 9.0 and nvidia-driver-390

Note: If you don't have a GPU card you can skip the steps involving CUDA and NVIDIA-DRIVER

1. Install all prerequisites/dependency

- Assuming its a fresh installation of Ubuntu 18.04 LTS

```
### commands to install packages in ubuntu box as a root user.
```

```
apt-get install aptitude
aptitude install vim csh flex patch gfortran g++ make xorg-dev bison libbz2-dev
```

```
### Parallel MPI packages
```

```
aptitude install openmpi-bin libopenmpi-dev
```

```
### Python packages
```

```
aptitude install python-numpy python-matplotlib python-scipy ipython python-notebook cython python-dev
```

2. GNU GCC 6 Installation

- Ubuntu 18.04 LTS comes with GCC 7 and AMBER16 requires GCC 6.

```
### GCC6 installation
```

```
aptitude install gcc-6 g++-6
```

```
#### create Symbolic Links
```

```
ln -s gcc-6 gcc
ln -s gcc-ar-6 gcc-ar
ln -s gcc-nm-6 gcc-nm
ln -s gcc-ranlib-6 gcc-ranlib
ln -s g++-6 g++
ln -s /usr/bin/gfortran-6 /usr/bin/gfortran
```

3. NETCDF installation

- Download parallel-netcdf-1.9.0

```
### Install netcdf
```

```
## go to netcdf folder followed by make and make install
cd parallel-netcdf-1.9.0/
```

```
make
```

```
make install
```

4. CUDA 9.0 Installation

- Assuming that the NVIDIA DRIVER installation went well! If not see that instruction at Step 6
- Download cuda_9.0.176_384.81_linux.run from NVIDIA website.

On the terminal , install cuda

```
./cuda_9.0.176_384.81_linux.run
```

add lines to ld.so.conf

```
echo "include /usr/local/cuda-9.0/lib64" >> /etc/ld.so.conf
```

run ldconfig

```
ldconfig
```

Add PATH and environment updates to ~/.bashrc file and source

```
export PATH="/usr/software/anaconda2/bin:$PATH"
```

```
test -f /usr/software/amber16/amber.sh && source /usr/software/amber16/amber.sh
```

```
export PATH=/usr/local/cuda-9.0/bin${PATH:+:${PATH}}export
```

```
LD_LIBRARY_PATH=/usr/local/cuda/lib64${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```

```
source ~/.bashrc
```

5. Compiling AMBER 16 and AMBERTOOLS 17

- Download AMBER with license purchase at ambermd.org
- Download AmberTools17 from ambermd.org

Steps on the terminal

Steps to compile the serial version assuming my files are in tmp folder.

```
tar xvvf /tmp/Amber16.tar.bz2
```

```
tar xvvf /tmp/AmberTools17.tar.bz2
```

```
cd /usr/software/amber16
```

```
./configure gnu
```

```
make install
```

Steps to compile the parallel version with NETCDF capabilities

```
./configure -mpi --with-pnetcdf /usr/software/parallel-netcdf gnu
```

```
make install
```

Steps to compile serial cuda enabled amber

```
./configure -cuda gnu
```

```
make install

#### Steps to compile parallel+netcdf+cuda enabled amber
./configure -mpi -cuda --with-pnetcdf /usr/soft/parallel-netcdf gnu

make install
```

6. NVIDIA DRIVER INSTALLATION

I followed the instruction given at <https://www.mvps.net/docs/install-nvidia-drivers-ubuntu-18-04-lts-bionic-beaver-linux/>

Clean the system of other Nvidia drivers

```
aptitude purge nvidia*
```

check the latest driver version for our Nvidia GPU

using link <https://www.nvidia.com/object/unix.html>

Add the Nvidia graphic card PPA

```
add-apt-repository ppa:graphics-drivers
```

Prepare the system for the installation

```
aptitude update
```

Install the Nvidia GPU driver

```
aptitude install nvidia-driver-390
```

```
reboot
```

Verify Nvidia Driver installation

```
lsmod | grep nvidia
```

or

```
nvidia-smi
```