ENVIRONMENT SETUP - C++ EXTENSION FOR PYTORCH

Given below is the exact build used to be able to run c++ extension on my local machine.
Step-by-step procedure for downloading for reference given below.

➔ **STEP 1:**

- Install anaconda3 to get a conda environment - https://www.digitalocean.com/community/tutorials/how-to-install-anaconda-on-ubuntu-18-04-quickstart
- Install the required cuda-toolkit and cudnn depending on the nvidia driver version.
  Nvidia driver version - 460.32.03
  cuda toolkit - 10.2.89
  cudnn - 7.6.5
- Install the required dependencies according to - https://github.com/pytorch/pytorch#from-source
- Install nvcc_linux-64 from conda-forge channel to be able to track cudatoolkit and cudnn outside the conda env.

  Reference websites:
  - https://developer.nvidia.com/cuda-10.2-download-archive?target_os=Linux
  - https://developer.nvidia.com/rdp/cudnn-archive
  - https://anaconda.org/conda-forge/nvcc_linux-64

➔ **STEP 2:**

- Set the appropriate environment variables in the bash terminal.
  - TORCH_CUDA_ARCH_LIST="3.5 5.2 6.0 6.1 7.0+PTX"
  - TORCH_NVCC_FLAGS="-Xfatbin -compress-all"
  - CMAKE_PREFIX_PATH=${CONDA_PREFIX:-"$(dirname $(which conda))/../"}
  - LD_LIBRARY_PATH=/usr/local/cuda-10.2/targets/x86_64-linux/include

  Reference websites:

  - https://en.wikipedia.org/wiki/CUDA - Under GPUs supported section to determine TORCH_CUDA_ARCH_LIST value.
  - https://michhar.github.io/how-i-built-pytorch-gpu/

➔ **STEP 3:**

- Build PyTorch from source -> clone the repo and build pytorch
- Build torchvision from source -> clone from pytorch/vision repo and build

Reference Websites:
- https://discuss.pytorch.org/t/issue-building-torchvision-from-source/98053/4
- https://github.com/pytorch/vision
- https://github.com/pytorch/pytorch#from-source

➔ **Some errors that can be encountered:**

◆ Missing library files error. (eg) *.h or *.hpp directory could not be found in **D**

Soln:

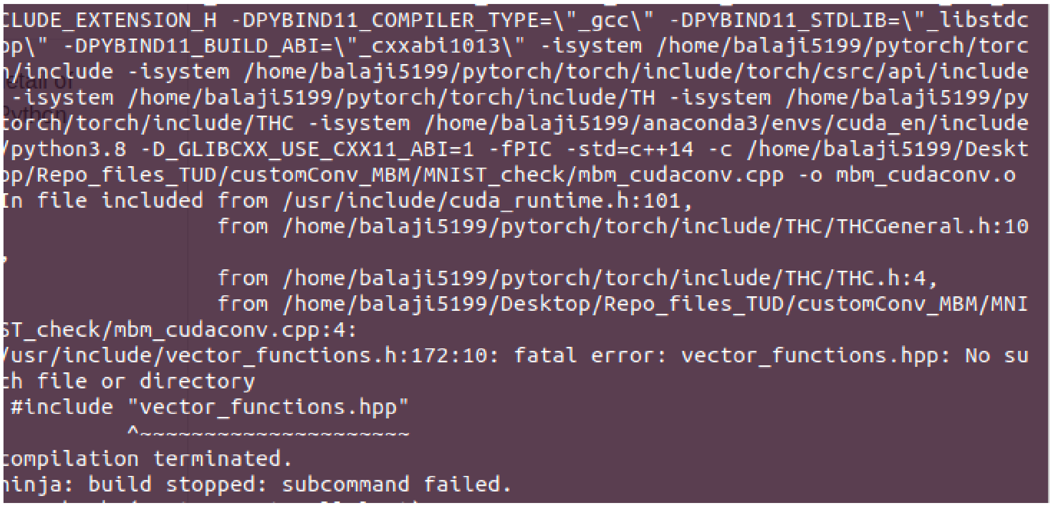
1) sudo find / -iname "*<name of the missing file>.h" - outputs the dir **'S'** where the header file is present

2) make a symlink -> sudo ln -s **S D**

where,

S is source dir

D is the dir where symlink is to be done



```
CLUDE_EXTENSION_H -DPYBIND11_COMPILER_TYPE=\"_gcc\" -DPYBIND11_STDLIB=\"_libstdc
pp\" -DPYBIND11_BUILD_ABI=\"_cxxabi1013\" -isystem /home/balaji5199/pytorch/torc
h/include -isystem /home/balaji5199/pytorch/torch/include/torch/csrc/api/include
 -isystem /home/balaji5199/pytorch/torch/include/TH -isystem /home/balaji5199/py
torch/torch/include/THC -isystem /home/balaji5199/anaconda3/envs/cuda_en/include
/python3.8 -D_GLIBCXX_USE_CXX11_ABI=1 -fPIC -std=c++14 -c /home/balaji5199/Deskt
op/Repo_files_TUD/customConv_MBM/MNIST_check/mbm_cudaconv.cpp -o mbm_cudaconv.o
In file included from /usr/include/cuda_runtime.h:101,
                 from /home/balaji5199/pytorch/torch/include/THC/THCGeneral.h:10
,
                 from /home/balaji5199/pytorch/torch/include/THC/THC.h:4,
                 from /home/balaji5199/Desktop/Repo_files_TUD/customConv_MBM/MNI
ST_check/mbm_cudaconv.cpp:4:
/usr/include/vector_functions.h:172:10: fatal error: vector_functions.hpp: No su
ch file or directory
 #include "vector_functions.hpp"
          ^~~~~~~~~~~~~~~~~~~~~~
compilation terminated.
ninja: build stopped: subcommand failed.
```

◆ CUDA error - "all CUDA-capable devices are busy or unavailable"

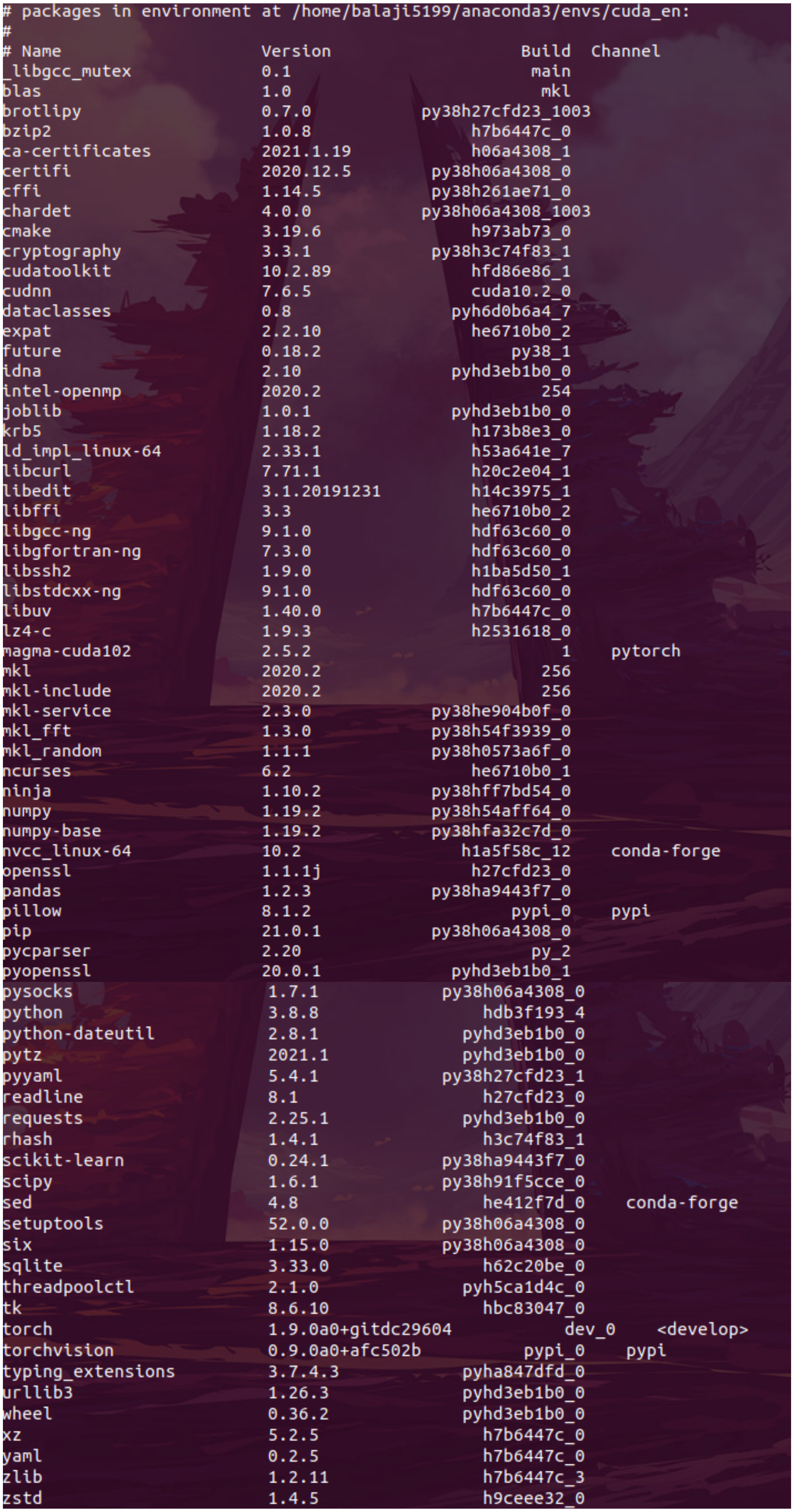
Soln: sudo reboot -> rebooting the pc worked for me.

➔ Further work:

◆ Now I will start working on the c++ extension for simulation

◆ I shall use DenseNet-161 architecture as the encoder part for simulations. Then we shall decide if we can move forward with MobileNetv1/ MobileNetv2, etc.

➔ Extensive list of dependencies in conda environment for your reference:



```
# packages in environment at /home/balaji5199/anaconda3/envs/cuda_en:
#
# Name                    Version                    Build  Channel
_libgcc_mutex             0.1                         main
blas                      1.0                          mkl
brotlipy                  0.7.0           py38h27cfd23_1003
bzip2                     1.0.8                 h7b6447c_0
ca-certificates           2021.1.19             h06a4308_1
certifi                   2020.12.5         py38h06a4308_0
cffi                      1.14.5            py38h261ae71_0
chardet                   4.0.0           py38h06a4308_1003
cmake                     3.19.6                h973ab73_0
cryptography              3.3.1             py38h3c74f83_1
cudatoolkit               10.2.89               hfd86e86_1
cudnn                     7.6.5                 cuda10.2_0
dataclasses               0.8                 pyh6d0b6a4_7
expat                     2.2.10                he6710b0_2
future                    0.18.2                    py38_1
idna                      2.10                pyhd3eb1b0_0
intel-openmp              2020.2                       254
joblib                    1.0.1               pyhd3eb1b0_0
krb5                      1.18.2                h173b8e3_0
ld_impl_linux-64          2.33.1                h53a641e_7
libcurl                   7.71.1                h20c2e04_1
libedit                   3.1.20191231          h14c3975_1
libffi                    3.3                   he6710b0_2
libgcc-ng                 9.1.0                 hdf63c60_0
libgfortran-ng            7.3.0                 hdf63c60_0
libssh2                   1.9.0                 h1ba5d50_1
libstdcxx-ng              9.1.0                 hdf63c60_0
libuv                     1.40.0                h7b6447c_0
lz4-c                     1.9.3                 h2531618_0
magma-cuda102             2.5.2                          1    pytorch
mkl                       2020.2                       256
mkl-include               2020.2                       256
mkl-service               2.3.0             py38he904b0f_0
mkl_fft                   1.3.0             py38h54f3939_0
mkl_random                1.1.1             py38h0573a6f_0
ncurses                   6.2                   he6710b0_1
ninja                     1.10.2            py38hff7bd54_0
numpy                     1.19.2            py38h54aff64_0
numpy-base                1.19.2            py38hfa32c7d_0
nvcc_linux-64             10.2                 h1a5f58c_12    conda-forge
openssl                   1.1.1j                h27cfd23_0
pandas                    1.2.3             py38ha9443f7_0
pillow                    8.1.2                     pypi_0    pypi
pip                       21.0.1            py38h06a4308_0
pycparser                 2.20                        py_2
pyopenssl                 20.0.1              pyhd3eb1b0_1
pysocks                   1.7.1             py38h06a4308_0
python                    3.8.8                 hdb3f193_4
python-dateutil           2.8.1               pyhd3eb1b0_0
pytz                      2021.1              pyhd3eb1b0_0
pyyaml                    5.4.1             py38h27cfd23_1
readline                  8.1                   h27cfd23_0
requests                  2.25.1              pyhd3eb1b0_0
rhash                     1.4.1                 h3c74f83_1
scikit-learn              0.24.1            py38ha9443f7_0
scipy                     1.6.1             py38h91f5cce_0
sed                       4.8                   he412f7d_0    conda-forge
setuptools                52.0.0            py38h06a4308_0
six                       1.15.0            py38h06a4308_0
sqlite                    3.33.0                h62c20be_0
threadpoolctl             2.1.0               pyh5ca1d4c_0
tk                        8.6.10                hbc83047_0
torch                     1.9.0a0+gitdc29604            dev_0    <develop>
torchvision               0.9.0a0+afc502b           pypi_0    pypi
typing_extensions         3.7.4.3             pyha847dfd_0
urllib3                   1.26.3              pyhd3eb1b0_0
wheel                     0.36.2              pyhd3eb1b0_0
xz                        5.2.5                 h7b6447c_0
yaml                      0.2.5                 h7b6447c_0
zlib                      1.2.11                h7b6447c_3
zstd                      1.4.5                 h9ceee32_0
```