

Build Tensorflow (GPU) R1.14 from source on Ubuntu

Computer specs

Ubuntu 18.04 LTS	
Device name	hairong-desktop-ubuntu
Memory	15.6 GiB
Processor	Intel® Core™ i7-7700 CPU @ 3.60GHz × 8
Graphics	GeForce GTX 1080/PCIe/SSE2
GNOME	3.28.1
OS type	64-bit
Disk	569.6 GB

create a virtual python env using conda (Note: tensorflow is not installed for this environment)

```
$ conda create --name py36_build_tf_gpu python=3.6
$ conda activate py36_build_tf_gpu
```

Create a directory ~/dev/tensorflow_r1.14_gpu to check out tensorflow source code r.1.14

Go to the folder "tensorflow_r1.14_gpu"

(py36_build_tf_gpu) hairong@hairong-desktop-ubuntu:~/dev\$ cd tensorflow_r1.14_gpu

Git clone

(py36_build_tf_gpu) hairong@hairong-desktop-ubuntu:~/dev/tensorflow_r1.14_gpu\$ git clone https://github.com/tensorflow/tensorflow .

(py36_build_tf_gpu) hairong@hairong-desktop-ubuntu:~/dev/tensorflow_r1.14_gpu\$ git checkout r1.14

Copy a screenshot of an area to clipboard Shift+Ctrl+Print

(py36_build_tf_gpu) hairong@hairong-desktop-ubuntu:~/dev/tensorflow_r1.14_gpu\$ gedit configure.py

```
(py36_build_tf_gpu) hairong@hairong-desktop-ubuntu:~/dev/tensorflow_r1.14_gpu$ gedit configure.py
```

```
def check_bazel_version(min_version, max_version):
    """Check installed bazel version is between min_version and max_version.S

    current_bazel_version = check_bazel_version('0.24.1', '0.26.2')
    _TF_CURRENT_BAZEL_VERSION = convert_version_to_int(current_bazel_version)
```

open configure.py to find minimum bazel version which is 0.24.1

Installing Bazel on Ubuntu (IMPORTANT!!! within the terminal of the virtual env py36_build_tf_gpu)
Install Bazel on Ubuntu using one of the following methods:

1. Use the binary installer (recommended)

2. Use our custom APT repository
3. Compile Bazel from source

Here I use the first method which is recommended.

Step 1: Install required packages

(py36_build_tf_gpu) **hairong@hairong-desktop-ubuntu:~/dev/tensorflow_r1.14_gpu\$** python tensorflow/tools/pip_package/setup.py install
(this takes 1 minute, a lot of stuff installed)

```
Using /home/hairong/.local/lib/python3.6/site-packages
Searching for numpy==1.17.2
Best match: numpy 1.17.2
Adding numpy 1.17.2 to easy-install.pth file
Installing f2py script to /home/hairong/anaconda3/envs/py36_build_tf_gpu/bin
Installing f2py3 script to /home/hairong/anaconda3/envs/py36_build_tf_gpu/bin
Installing f2py3.6 script to /home/hairong/anaconda3/envs/py36_build_tf_gpu/bin

Using /home/hairong/.local/lib/python3.6/site-packages
Searching for setuptools==41.2.0
Best match: setuptools 41.2.0
Adding setuptools 41.2.0 to easy-install.pth file
Installing easy_install script to /home/hairong/anaconda3/envs/py36_build_tf_gpu/bin
Installing easy_install-3.6 script to /home/hairong/anaconda3/envs/py36_build_tf_gpu/bin

Using /home/hairong/.local/lib/python3.6/site-packages
Finished processing dependencies for tensorflow==1.14.1
(py36_build_tf_gpu) hairong@hairong-desktop-ubuntu:~/dev/tensorflow_r1.14_gpu$
```

(py36_build_tf_gpu) **hairong@hairong-desktop-ubuntu:~/dev/tensorflow_r1.14_gpu\$** sudo apt-get install pkg-config zip g++ zlib1g-dev unzip

[sudo] password for hairong:

Reading package lists... Done

Building dependency tree

Reading state information... Done

pkg-config is already the newest version (0.29.1-0ubuntu2).

unzip is already the newest version (6.0-21ubuntu1).

zip is already the newest version (3.0-11build1).

zlib1g-dev is already the newest version (1:1.2.11.dfsg-0ubuntu2).

g++ is already the newest version (4:7.4.0-1ubuntu2.3).

The following package was automatically installed and is no longer required:

libnvidia-common-390

Use 'sudo apt autoremove' to remove it.

0 upgraded, 0 newly installed, 0 to remove and 16 not upgraded.

(py36_build_tf_gpu) **hairong@hairong-desktop-ubuntu:~/dev/tensorflow_r1.14_gpu\$**

Step 2: Download Bazel (version 0.24.1) and enter the download folder from the terminal

<https://github.com/bazelbuild/bazel/releases>

 [bazel-0.24.1-installer-linux-x86_64.sh](#)

91.3 MB

Enter /home/hairong/snap/firefox/common/Downloads where bazel is downloaded. Keep stay in py36_build_tf_gpu environment.

(py36_build_tf_gpu) **hairong@hairong-desktop-ubuntu:~/dev/tensorflow_r1.14_gpu\$** cd /home/hairong/snap/firefox/common/Downloads

```
(py36_build_tf_gpu) hairong@hairong-desktop-ubuntu:~/snap/firefox/common/Downloads$
```

Step 3: Run the installer

```
(py36_build_tf_gpu) hairong@hairong-desktop-ubuntu:~/snap/firefox/common/Downloads$ chmod +x  
bazel-0.24.1-installer-linux-x86_64.sh
```

```
(py36_build_tf_gpu)
```

```
hairong@hairong-desktop-ubuntu:~/snap/firefox/common/Downloads$ ./bazel-0.24.1-installer-linux-x86  
_64.sh --user
```

The --user flag installs Bazel to the \$HOME/bin directory on your system and sets the .bazelrc path to \$HOME/.bazelrc.

I used -user so that the previous version of bazel will be overwritten.

Use the --help command to see additional installation options. This also verify that bazel is installed successfully.

Check bazel version

```
(py36_build_tf_gpu) hairong@hairong-desktop-ubuntu:~/snap/firefox/common/Downloads$ bazel  
versionWARNING: --batch mode is deprecated. Please instead explicitly shut down your Bazel server  
using the command "bazel shutdown".
```

```
Build label: 0.24.1
```

```
Build target:
```

```
bazel-out/k8-opt/bin/src/main/java/com/google/devtools/build/lib/bazel/BazelServer_deploy.jar
```

```
Build time: Tue Apr 2 16:29:26 2019 (1554222566)
```

```
Build timestamp: 1554222566
```

```
Build timestamp as int: 1554222566
```

Step 4: Set up your environment

Check path variable

```
(py36_build_tf_gpu) hairong@hairong-desktop-ubuntu:~/snap/firefox/common/Downloads$ echo $PATH  
/home/hairong/anaconda3/envs/py36_build_tf_gpu/bin:/home/hairong/anaconda3/condabin:/home/hairon  
g/.local/bin:/home/hairong/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/ga  
mes:/usr/l  
ocal/games:/snap/bin
```

If you ran the Bazel installer with the --user flag as above, the Bazel executable is installed in your \$HOME/bin directory. It's a good idea to add this directory to your default paths, as follows:

```
export PATH="$PATH:$HOME/bin"
```

You can also add this command to your ~/.bashrc file. (I didn't do this. will consider later if there is any problem coming up)

That's it for installation of bazel. for more details, check the link:

<https://docs.bazel.build/versions/master/install-ubuntu.html#install-with-installer-ubuntu>

Next, go back to the tensorflow directoy to build tensorflow.

```
(py36_build_tf_gpu) hairong@hairong-desktop-ubuntu:~/snap/firefox/common/Downloads$ cd  
~/dev/tensorflow_r1.14_gpu/
```

```
(py36_build_tf_gpu) hairong@hairong-desktop-ubuntu:~/dev/tensorflow_r1.14_gpu$
```

Run configure before build

```
./configure
```

Hairong comments: I turned off or skipped most of the options/extensions But turned on CUDA which is what i want. However, since I didn't installed CUDA, cudnn, ect (only installed Nvidia display driver), I encountered below issue:

```
Do you wish to build TensorFlow with CUDA support? [y/N]: y
```

CUDA support will be enabled for TensorFlow.

```
Do you wish to build TensorFlow with TensorRT support? [y/N]: n
```

No TensorRT support will be enabled for TensorFlow.

Could not find any cuda.h matching version '' in any subdirectory:

''

'include'

'include/cuda'

'include/*-linux-gnu'

'extras/CUPTI/include'

'include/cuda/CUPTI'

of:

'/lib/x86_64-linux-gnu'

'/usr'

'/usr/lib'

'/usr/lib/x86_64-linux-gnu'

'/usr/lib/x86_64-linux-gnu/libfakeroot'

Asking for detailed CUDA configuration...

Please specify the CUDA SDK version you want to use. [Leave empty to default to CUDA 10]:

Now install CUDA and cudnn. What versions should i install?

Previously I successfully installed a virtual python (using Anaconda 3.7). It automatically installed cuda toolkit and cudnn. Let's find out what version they are. So We will install the same version for our build from source.

```
(py36_tf_gpu) hairong@hairong-desktop-ubuntu:~$ conda list
```

```
# packages in environment at /home/hairong/anaconda3/envs/py36_tf_gpu:
```

```
#
```

# Name	Version	Build	Channel
_libgcc_mutex	0.1	main	
_tfflow_select	2.1.0	gpu	
absl-py	0.7.1	py36_0	
astor	0.8.0	py36_0	
blas	1.0	mkl	

bzip2	1.0.8	h7b6447c_0
c-ares	1.15.0	h7b6447c_1001
ca-certificates	2019.5.15	1
cairo	1.14.12	h8948797_3
certifi	2019.6.16	py36_1
cuda-toolkit	10.1.168	0
cudnn	7.6.0	cuda10.1_0
cupi	10.1.168	0

Open a new terminal and got the environment I was working on to build tensorflow

(base) hairong@hairong-desktop-ubuntu:~\$ conda activate py36_build_tf_gpu

(py36_build_tf_gpu) hairong@hairong-desktop-ubuntu:~\$ nvidia-smi

Sat Sep 21 23:31:29 2019

```

+-----+
| NVIDIA-SMI 430.40          Driver Version: 430.40          CUDA Version: 10.1          |
+-----+-----+-----+
| GPU  Name                Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
+-----+-----+-----+-----+
|  0  GeForce GTX 1080        Off      | 00000000:01:00:0  On   |           N/A       |
| 33%   26C    P8      11W / 180W |  387MiB /  8111MiB |      0%      Default |
+-----+-----+-----+

```

```

+-----+
| Processes:                                     GPU Memory |
|  GPU       PID    Type    Process name                     Usage      |
+-----+-----+-----+-----+
|  0         1617     G       /usr/lib/xorg/Xorg                     18MiB |
|  0         1663     G       /usr/bin/gnome-shell                   49MiB |
|  0         1873     G       /usr/lib/xorg/Xorg                    153MiB |
|  0         2015     G       /usr/bin/gnome-shell                   160MiB |
|  0         4981     G       gnome-control-center                    2MiB |
+-----+

```

This shows that the display driver version is the latest already.

Install CUDA with apt

This section shows how to install CUDA 10 (TensorFlow >= 13.0) and CUDA 9 for Ubuntu 16.04 and 18.04. These instructions may work for other Debian-based distros.

 **Caution:** [Secure Boot](#)  complicates installation of the NVIDIA driver and is beyond the scope of these instructions.

Ubuntu 18.04 (CUDA 10)

```
# Add NVIDIA package repositories
$ wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64/cuda-repo-ubuntu1804_10.0.130-1_amd64.deb
$ sudo dpkg -i cuda-repo-ubuntu1804_10.0.130-1_amd64.deb
$ sudo apt-key adv --fetch-keys https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64/7fa2af80ca1e1b2f1a8059b099b06f80
$ sudo apt-get update
$ wget http://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu1804/x86_64/nvidia-machine-learning-repo-ubuntu1804_1.0.0-1_amd64.deb
$ sudo apt install ./nvidia-machine-learning-repo-ubuntu1804_1.0.0-1_amd64.deb
$ sudo apt-get update

# Install NVIDIA driver
$ sudo apt-get install --no-install-recommends nvidia-driver-418
# Reboot. Check that GPUs are visible using the command: nvidia-smi

# Install development and runtime libraries (~4GB)
$ sudo apt-get install --no-install-recommends \
    cuda-10-0 \
    libcudnn7=7.6.2.24-1+cuda10.0 \
    libcudnn7-dev=7.6.2.24-1+cuda10.0

# Install TensorRT. Requires that libcudnn7 is installed above.
$ sudo apt-get install -y --no-install-recommends libnvinfer5=5.1.5-1+cuda10.0 \
    libnvinfer-dev=5.1.5-1+cuda10.0
```

Install CUDA with apt (https://www.tensorflow.org/install/gpu#install_cuda_with_apt)

Ubuntu 18.04 (CUDA 10)

Add NVIDIA package repositories

wget

https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64/cuda-repo-ubuntu1804_10.0.130-1_amd64.deb

(py36_build_tf_gpu) **hairong@hairong-desktop-ubuntu:~\$** wget

https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64/cuda-repo-ubuntu1804_10.0.130-1_amd64.deb



```
sudo dpkg -i cuda-repo-ubuntu1804_10.0.130-1_amd64.deb
(py36_build_tf_gpu) hairong@hairong-desktop-ubuntu:~$ sudo dpkg -i
cuda-repo-ubuntu1804_10.0.130-1_amd64.deb
[sudo] password for hairong:
Selecting previously unselected package cuda-repo-ubuntu1804.
(Reading database ... 134306 files and directories currently installed.)
Preparing to unpack cuda-repo-ubuntu1804_10.0.130-1_amd64.deb ...
Unpacking cuda-repo-ubuntu1804 (10.0.130-1) ...
Setting up cuda-repo-ubuntu1804 (10.0.130-1) ...
```

The public CUDA GPG key does not appear to be installed.

To install the key, run this command:

```
sudo apt-key adv --fetch-keys
```

```
http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64/7fa2af80.pub
```

```
sudo apt-key adv --fetch-keys
```

```
https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86\_64/7fa2af80.pub
```

```
(py36_build_tf_gpu) hairong@hairong-desktop-ubuntu:~$ sudo apt-key adv --fetch-keys
http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64/7fa2af80.pub
Executing: /tmp/apt-key-gpghome.jhypDXMeeA/gpg.1.sh --fetch-keys
http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64/7fa2af80.pub
gpg: requesting key from
'http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64/7fa2af80.pub'
gpg: key F60F4B3D7FA2AF80: public key "cudatools <cudatools@nvidia.com>" imported
gpg: Total number processed: 1
gpg:                imported: 1
```

```
sudo apt-get update
```

```
(py36_build_tf_gpu) hairong@hairong-desktop-ubuntu:~$ sudo apt-get update
Ign:1 http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64 InRelease
Get:2 http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64 Release [564 B]
Get:3 http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64 Release.gpg
[819 B]
Get:4 http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64 Packages [113
kB]
Hit:5 http://ppa.launchpad.net/graphics-drivers/ppa/ubuntu bionic InRelease
Hit:6 http://ca.archive.ubuntu.com/ubuntu bionic InRelease
Fetched 115 kB in 1s (147 kB/s)
Reading package lists... Done
```

```
wget
http://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu1804/x86_64/nvidia-machine-learning-repo-ubuntu1804_1.0.0-1_amd64.deb
(py36_build_tf_gpu) hairong@hairong-desktop-ubuntu:~$ wget
http://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu1804/x86_64/nvidia-machine-learning-repo-ubuntu1804_1.0.0-1_amd64.deb
--2019-09-21 23:54:14--
http://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu1804/x86_64/nvidia-machine-learning-repo-ubuntu1804_1.0.0-1_amd64.deb
Resolving developer.download.nvidia.com (developer.download.nvidia.com)...
2606:2800:21f:3aa:dcf:37b:1ed6:1fb, 192.229.211.70
Connecting to developer.download.nvidia.com
(developer.download.nvidia.com)[2606:2800:21f:3aa:dcf:37b:1ed6:1fb]:80... failed: Network is unreachable.
Connecting to developer.download.nvidia.com (developer.download.nvidia.com)[192.229.211.70]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2926 (2.9K) [application/x-deb]
Saving to: 'nvidia-machine-learning-repo-ubuntu1804_1.0.0-1_amd64.deb'
```

```
nvidia-machine-lear 100%[=====] 2.86K --.-KB/s in 0s
```

```
2019-09-21 23:54:14 (497 MB/s) - 'nvidia-machine-learning-repo-ubuntu1804_1.0.0-1_amd64.deb'
saved [2926/2926]
```



```
sudo apt install ./nvidia-machine-learning-repo-ubuntu1804_1.0.0-1_amd64.deb
(py36_build_tf_gpu) hairong@hairong-desktop-ubuntu:~$ sudo apt
install ./nvidia-machine-learning-repo-ubuntu1804_1.0.0-1_amd64.deb
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'nvidia-machine-learning-repo-ubuntu1804' instead of
'./nvidia-machine-learning-repo-ubuntu1804_1.0.0-1_amd64.deb'
The following package was automatically installed and is no longer required:
  libnvidia-common-390
Use 'sudo apt autoremove' to remove it.
The following NEW packages will be installed:
  nvidia-machine-learning-repo-ubuntu1804
```


0 upgraded, 1 newly installed, 0 to remove and 19 not upgraded.
Need to get 0 B/2,926 B of archives.
After this operation, 37.9 kB of additional disk space will be used.
Get:1 /home/hairong/nvidia-machine-learning-repo-ubuntu1804_1.0.0-1_amd64.deb
nvidia-machine-learning-repo-ubuntu1804 amd64 1.0.0-1 [2,926 B]
Selecting previously unselected package nvidia-machine-learning-repo-ubuntu1804.
(Reading database ... 134309 files and directories currently installed.)
Preparing to unpack ../nvidia-machine-learning-repo-ubuntu1804_1.0.0-1_amd64.deb ...
Unpacking nvidia-machine-learning-repo-ubuntu1804 (1.0.0-1) ...
Setting up nvidia-machine-learning-repo-ubuntu1804 (1.0.0-1) ...

```
sudo apt-get update
(py36_build_tf_gpu) hairong@hairong-desktop-ubuntu:~$ sudo apt-get update
Ign:1 http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64 InRelease
Ign:2 http://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu1804/x86_64 InRelease
Hit:3 http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64 Release
Get:4 http://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu1804/x86_64 Release [564 B]
Get:5 http://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu1804/x86_64 Release.gpg [833 B]
Get:7 http://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu1804/x86_64 Packages [19.8 kB]
Hit:8 http://ppa.launchpad.net/graphics-drivers/ppa/ubuntu bionic InRelease
Hit:9 http://ca.archive.ubuntu.com/ubuntu bionic InRelease
Fetched 21.2 kB in 1s (34.0 kB/s)
Reading package lists... Done
```

Install NVIDIA driver (I chose not to install since I have a newer one installed already)
sudo apt-get install --no-install-recommends nvidia-driver-418

Reboot. Check that GPUs are visible using the command: nvidia-smi

```
(py36_build_tf_gpu) hairong@hairong-desktop-ubuntu:~$ nvidia-smi
Sun Sep 22 00:02:21 2019

+-----+
| NVIDIA-SMI 430.40              Driver Version: 430.40          CUDA Version: 10.1     |
+-----+-----+
| GPU   Name                Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|=====+=====+
| 0   GeForce GTX 1080    Off      | 00000000:01:00.0 On  |          N/A         |
| 32%   26C   P8      8W / 180W | 393MiB / 8111MiB |      0%      Default |
+-----+-----+

+-----+
| Processes:                                     GPU Memory |
|  GPU       PID    Type    Process name                       Usage      |
|=====+=====+
|    0         1617     G   /usr/lib/xorg/Xorg                      18MiB      |
|    0         1663     G   /usr/bin/gnome-shell                   49MiB      |
|    0         1873     G   /usr/lib/xorg/Xorg                      155MiB     |
|    0         2015     G   /usr/bin/gnome-shell                   165MiB     |
|    0         4981     G   gnome-control-center                    2MiB       |
+-----+
```

Install development and runtime libraries (~4GB)

```
sudo apt-get install --no-install-recommends \
    cuda-10-0 \
    libcudnn7=7.6.2.24-1+cuda10.0 \
    libcudnn7-dev=7.6.2.24-1+cuda10.0
```

```
(py36_build_tf_gpu) hairong@hairong-desktop-ubuntu:~$ sudo apt-get install
--no-install-recommends cuda-10-0
libcudnn7=7.6.2.24-1+cuda10.0 libcudnn7-dev=7.6.2.24-1+cuda10.0
```

E: Command line option --no-install-recommends is not understood in combination with the other options

```
(py36_build_tf_gpu) hairong@hairong-desktop-ubuntu:~$ sudo apt-get install cuda-10-0
libcudnn7=7.6.2.24-1+cuda10.0 libcudnn7-dev=7.6.2.24-1+cuda10.0
```

E: Invalid operation install

Hairong's comments:

IMPORTANT!!!

There were problems to run above cmd at once. After a few tests, I was able to separate it into 3 installs as below:

```
sudo apt-get install --no-install-recommends cuda-10-0
```

```
sudo apt-get install --no-install-recommends libcudnn7=7.6.2.24-1+cuda10.0
```

```
sudo apt-get install --no-install-recommends libcudnn7-dev=7.6.2.24-1+cuda10.0
```

Install them separately, one at a time

```
(py36_build_tf_gpu) hairong@hairong-desktop-ubuntu:~$ sudo apt-get install --no-install-recommends  
cuda-10-0
```

...

Running hooks in /etc/ca-certificates/update.d...

done.

done.

```
(py36_build_tf_gpu) hairong@hairong-desktop-ubuntu:~$ sudo apt-get install --no-install-recommends  
libcudnn7=7.6.2.24-1+cuda10.0
```

[sudo] password for hairong:

Reading package lists... Done

Building dependency tree

Reading state information... Done

The following package was automatically installed and is no longer required:

libnvidia-common-390

Use 'sudo apt autoremove' to remove it.

The following NEW packages will be installed:

libcudnn7

0 upgraded, 1 newly installed, 0 to remove and 18 not upgraded.

Need to get 164 MB of archives.

After this operation, 391 MB of additional disk space will be used.

Get:1 http://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu1804/x86_64

libcudnn7 7.6.2.24-1+cuda10.0 [164 MB]

Fetch: 164 MB in 15s (11.1 MB/s)

Selecting previously unselected package libcudnn7.

(Reading database ... 146986 files and directories currently installed.)

Preparing to unpack .../libcudnn7_7.6.2.24-1+cuda10.0_amd64.deb ...

Unpacking libcudnn7 (7.6.2.24-1+cuda10.0) ...

Setting up libcudnn7 (7.6.2.24-1+cuda10.0) ...

Processing triggers for libc-bin (2.27-3ubuntu1) ...

```
(py36_build_tf_gpu) hairong@hairong-desktop-ubuntu:~$ sudo apt-get install --no-install-recommends  
libcudnn7-dev=7.6.2.24-1+cuda10.0
```

The following package was automatically installed and is no longer required:

libnvidia-common-390

Use 'sudo apt autoremove' to remove it.

The following NEW packages will be installed:

libcudnn7-dev

0 upgraded, 1 newly installed, 0 to remove and 19 not upgraded.

Need to get 152 MB of archives.

After this operation, 390 MB of additional disk space will be used.

```
Get:1 http://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu1804/x86_64
libcudnn7-dev 7.6.2.24-1+cuda10.0 [152 MB]
Fetched 152 MB in 14s (10.9 MB/s)
Selecting previously unselected package libcudnn7-dev.
(Reading database ... 146992 files and directories currently installed.)
Preparing to unpack .../libcudnn7-dev_7.6.2.24-1+cuda10.0_amd64.deb ...
Unpacking libcudnn7-dev (7.6.2.24-1+cuda10.0) ...
Setting up libcudnn7-dev (7.6.2.24-1+cuda10.0) ...
update-alternatives: using /usr/include/x86_64-linux-gnu/cudnn_v7.h to provide /usr/include/cudnn.h
(libcudnn) in auto mode
```

Install TensorRT. Requires that libcudnn7 is installed above. (I didn't run this since I won't need TensorRT for now)

```
sudo apt-get install -y --no-install-recommends libnvinfer5=5.1.5-1+cuda10.0 \
    libnvinfer-dev=5.1.5-1+cuda10.0
```

```
(py36_build_tf_gpu) hairong@hairong-desktop-ubuntu:~/dev/tensorflow_r1.14_gpu$ ./configure
```

```
./configure
```

```
(py36_build_tf_gpu) hairong@hairong-desktop-ubuntu:~/dev$ cd tensorflow_r1.14_gpu
```

```
(py36_build_tf_gpu)
```

```
hairong@hairong-desktop-ubuntu:~/dev/tensorflow_r1.14_gpu$ ./configureWARNING: --batch mode is
deprecated. Please instead explicitly shut down your Bazel server using the command "bazel shutdown".
You have bazel 0.24.1 installed.
```

```
Please specify the location of python. [Default is
/home/hairong/anaconda3/envs/py36_build_tf_gpu/bin/python]:
```

Found possible Python library paths:

```
/home/hairong/anaconda3/envs/py36_build_tf_gpu/lib/python3.6/site-packages
Please input the desired Python library path to use. Default is
[/home/hairong/anaconda3/envs/py36_build_tf_gpu/lib/python3.6/site-packages]
```

```
Do you wish to build TensorFlow with XLA JIT support? [Y/n]: n
No XLA JIT support will be enabled for TensorFlow.
```

```
Do you wish to build TensorFlow with OpenCL SYCL support? [y/N]: n
No OpenCL SYCL support will be enabled for TensorFlow.
```

```
Do you wish to build TensorFlow with ROCm support? [y/N]: n
No ROCm support will be enabled for TensorFlow.
```

```
Do you wish to build TensorFlow with CUDA support? [y/N]: y
```

CUDA support will be enabled for TensorFlow.

Do you wish to build TensorFlow with TensorRT support ? [y/N]: n

No TensorRT support will be enabled for TensorFlow.

Found CUDA 10.0 in:

/usr/local/cuda/lib64

/usr/local/cuda/include

Found cuDNN 7 in:

/usr/lib/x86_64-linux-gnu

/usr/include

Please specify a list of comma-separated CUDA compute capabilities you want to build with.

You can find the compute capability of your device at: <https://developer.nvidia.com/cuda-gpus>.

Please note that each additional compute capability significantly increases your build time and binary size, and that TensorFlow only supports compute capabilities ≥ 3.5 [Default is: 6.1]:

Do you want to use clang as CUDA compiler ? [y/N]: n

nvcc will be used as CUDA compiler.

Please specify which gcc should be used by nvcc as the host compiler. [Default is /usr/bin/gcc]:

Do you wish to build TensorFlow with MPI support ? [y/N]: n

No MPI support will be enabled for TensorFlow.

Please specify optimization flags to use during compilation when bazel option "--config=opt" is specified [Default is -march=native -Wno-sign-compare]:

Would you like to interactively configure ./WORKSPACE for Android builds ? [y/N]: n

Not configuring the WORKSPACE for Android builds.

Preconfigured Bazel build configs. You can use any of the below by adding "--config=<>" to your build command. See .bazelrc for more details.

--config=mkl	# Build with MKL support.
--config=monolithic	# Config for mostly static monolithic build.
--config=gdr	# Build with GDR support.
--config=verbs	# Build with libverbs support.
--config=ngraph	# Build with Intel nGraph support.
--config=numa	# Build with NUMA support.

--config=dynamic_kernels # (Experimental) Build kernels into separate shared objects.

Preconfigured Bazel build configs to DISABLE default on features:

--config=noaws # Disable AWS S3 filesystem support.
--config=nogcp # Disable GCP support.
--config=nohdfs # Disable HDFS support.
--config=noignite # Disable Apache Ignite support.
--config=nokafka # Disable Apache Kafka support.
--config=nonccl # Disable NVIDIA NCCL support.

Configuration finished

(py36_build_tf_gpu) hairong@hairong-desktop-ubuntu:~/dev/tensorflow_r1.14_gpu\$

Build (building time: about 1 hour)

Looking through the bazel BUILD files to find targets you will find the right target, and you can build the C++ extension using :

```
bazel build --jobs=6 --verbose_failures //tensorflow:libtensorflow_cc.so
```

Hairong comments: I chose jobs=6 since my computer has 4x2 cpu cores. The build took a while (may be 1 hour?)

(py36_build_tf_gpu) hairong@hairong-desktop-ubuntu:~/dev/tensorflow_r1.14_gpu\$ bazel build --jobs=6 --verbose_failures //tensorflow:libtensorflow_cc.so

.....
.....

INFO: From Compiling tensorflow/core/kernels/reduction_ops_gpu_complex64.cu.cc:
external/com_google_absl/absl/strings/string_view.h(495): warning: expression has no effect

external/com_google_absl/absl/strings/string_view.h(495): warning: expression has no effect

Target //tensorflow:libtensorflow_cc.so up-to-date:

bazel-bin/tensorflow/libtensorflow_cc.so

INFO: Elapsed time: 2903.012s, Critical Path: 160.21s

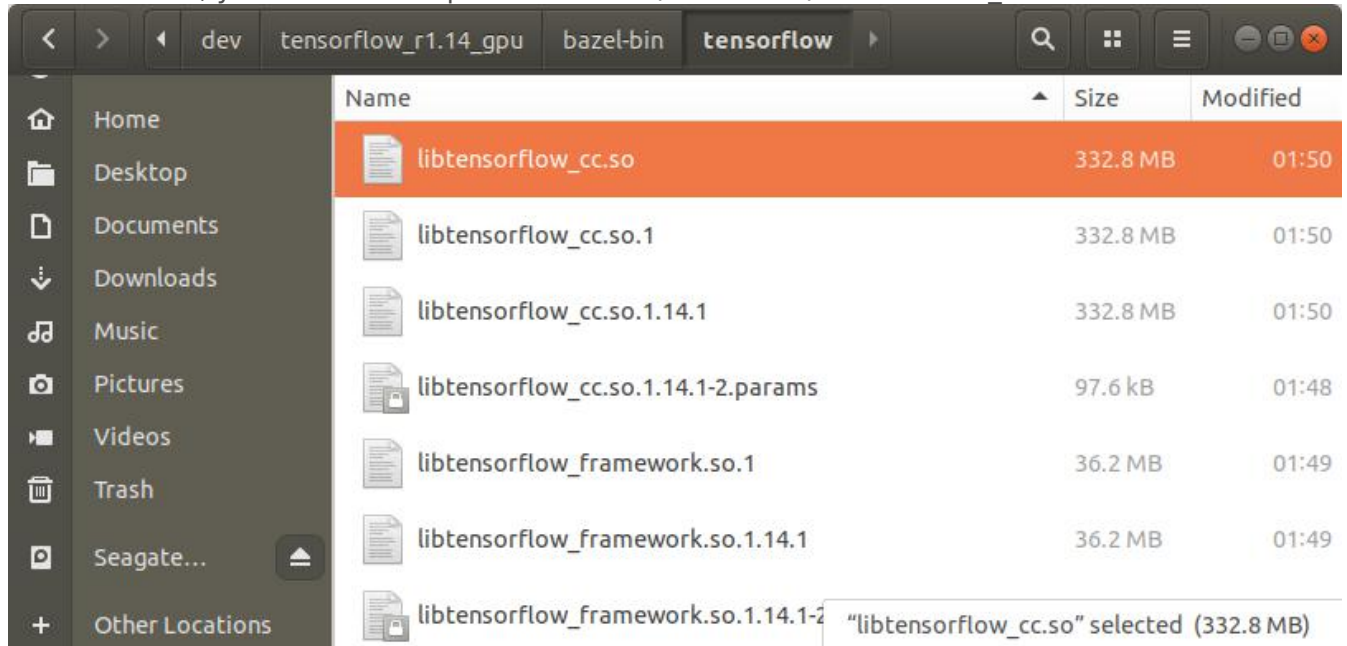
INFO: 6857 processes: 6857 local.

INFO: Build completed successfully, 10410 total actions

(py36_build_tf_gpu) hairong@hairong-desktop-ubuntu:~/dev/tensorflow_r1.14_gpu\$

```
Target //tensorflow:libtensorflow_cc.so up-to-date:
  bazel-bin/tensorflow/libtensorflow_cc.so
INFO: Elapsed time: 2903.012s, Critical Path: 160.21s
INFO: 6857 processes: 6857 local.
INFO: Build completed successfully, 10410 total actions
```

Once it's done, you should end up with `bazel-bin/tensorflow/libtensorflow_cc.so`.



Future work: create a C++ app that can do inference using trained model from Tensorflow

you need to copy them then, either to common lib directory or current one :

```
cd .. & cp -R tensorflow/bazel-bin/tensorflow/libtensorflow_* ./
```

Didn't the copying yet. Will do it later when I need to use the lib.

151 history >cmd_history_build_tfcpu_fromsource_ubuntu18.04.txt

Your `LD_LIBRARY_PATH` doesn't include the path to `libsvmlight.so`.

```
$ export
LD_LIBRARY_PATH=/home/hairong/dev/tensorflow/bazel-bin/tensorflow:$LD_LIBRARY_PATH

./k2tf --graph=./output_graph.pb --labels=./data/flowers/raw-data/labels.txt --input_width=80
--input_height=80 --input_layer=firstConv2D_input --output_layer=k2tfout_0
--image=./data/flowers/raw-data/validation/dandelion/13920113_f03e867ea7_m.jpg
```