Project: tensorflow

2017-12-23T13:58:25Z kirk86 107comments

ImportError: libcublas.so.9.0: cannot open shared object file: No such file or directory

Web url: https://github.com/tensorflow/tensorflow/issues/15604

API url: https://api.github.com/repos/tensorflow/tensorflow/issues/15604

I installed tf-nightly build and I get the following error on import of tensorflow.

`ImportError: libcublas.so.9.0: cannot open shared object file: No such file or directory`.

If I check for cuda 9, I get the following:

```

ldconfig -v

/usr/local/cuda-8.0/targets/x86\_64-linux/lib:

libnvgraph.so.8.0 -> libnvgraph.so.8.0.61

libnppicom.so.8.0 -> libnppicom.so.8.0.61

libnppial.so.8.0 -> libnppial.so.8.0.61

libcufftw.so.8.0 -> libcufftw.so.8.0.61

libcufft.so.8.0 -> libcufft.so.8.0.61

libnppif.so.8.0 -> libnppif.so.8.0.61

libcublas.so.8.0 -> libcublas.so.8.0.88

libnvblas.so.8.0 -> libnvblas.so.8.0.88

libnppi.so.8.0 -> libnppi.so.8.0.61

libcusolver.so.8.0 -> libcusolver.so.8.0.61

libnppidei.so.8.0 -> libnppidei.so.8.0.61

libnvrtc-builtins.so.8.0 -> libnvrtc-builtins.so.8.0.61

libnvrtc.so.8.0 -> libnvrtc.so.8.0.61

libnpps.so.8.0 -> libnpps.so.8.0.61

libcuinj64.so.8.0 -> libcuinj64.so.8.0.61

libnppig.so.8.0 -> libnppig.so.8.0.61

libOpenCL.so.1 -> libOpenCL.so.1.0.0

libnppicc.so.8.0 -> libnppicc.so.8.0.61

libnppist.so.8.0 -> libnppist.so.8.0.61

libnppisu.so.8.0 -> libnppisu.so.8.0.61

libnppim.so.8.0 -> libnppim.so.8.0.61

libcurand.so.8.0 -> libcurand.so.8.0.61

libcudart.so.8.0 -> libcudart.so.8.0.61

libnvToolsExt.so.1 -> libnvToolsExt.so.1.0.0

libnppitc.so.8.0 -> libnppitc.so.8.0.61

libnppc.so.8.0 -> libnppc.so.8.0.61

libcusparse.so.8.0 -> libcusparse.so.8.0.61

/usr/local/cuda-9.1/targets/x86\_64-linux/lib:

libnppicc.so.9.1 -> libnppicc.so.9.1.85

libnppisu.so.9.1 -> libnppisu.so.9.1.85

libcufftw.so.9.1 -> libcufftw.so.9.1.85

libcufft.so.9.1 -> libcufft.so.9.1.85

libnppial.so.9.1 -> libnppial.so.9.1.85

libnppist.so.9.1 -> libnppist.so.9.1.85

libcublas.so.9.1 -> libcublas.so.9.1.85

libnvblas.so.9.1 -> libnvblas.so.9.1.85

libnppitc.so.9.1 -> libnppitc.so.9.1.85

libcusolver.so.9.1 -> libcusolver.so.9.1.85

libnvrtc.so.9.1 -> libnvrtc.so.9.1.85

libnvrtc-builtins.so.9.1 -> libnvrtc-builtins.so.9.1.85

libnppidei.so.9.1 -> libnppidei.so.9.1.85

libOpenCL.so.1 -> libOpenCL.so.1.0.0

libnppig.so.9.1 -> libnppig.so.9.1.85

libnppc.so.9.1 -> libnppc.so.9.1.85

libcudart.so.9.1 -> libcudart.so.9.1.85

libnvToolsExt.so.1 -> libnvToolsExt.so.1.0.0

libnvgraph.so.9.1 -> libnvgraph.so.9.1.85

libnppif.so.9.1 -> libnppif.so.9.1.85

libcusparse.so.9.1 -> libcusparse.so.9.1.85

libaccinj64.so.9.1 -> libaccinj64.so.9.1.85

libcuinj64.so.9.1 -> libcuinj64.so.9.1.85

libnppim.so.9.1 -> libnppim.so.9.1.85

libnppicom.so.9.1 -> libnppicom.so.9.1.85

libnpps.so.9.1 -> libnpps.so.9.1.85

libcurand.so.9.1 -> libcurand.so.9.1.85

```

I that due to a name mismatch. `libcublas.so.9.0 =! libcublas.so.9.1`? And if so how can we overcome this?

-------------------------------------------------------------------------

2017-12-23T14:43:12Z Timonzimm

I think this is due to the fact that you have CUDA 9.1 and not 9.0, I am facing exactly the same issue.

-------------------------------------------------------------------------

2017-12-23T15:31:18Z kirk86

@Timonzimm I know and I think the whole issue is this f\*\* naming libcublas.so.xxx that nvidia puts. This inherently is mismatch on linux systems whenever that number changes, so since it can not find the exact matches then it thinks the file doesn't exist and throws the error.

-------------------------------------------------------------------------

2017-12-26T12:25:52Z burui11087

I think you should use symbol link from ''cuda/'' to ''cuda/9.1",or your cuda version is too new to tensorflow master branch

-------------------------------------------------------------------------

2017-12-27T21:32:54Z kirk86

@burui11087 I completely forgot about symlinking. Thanks for reminding me.

-------------------------------------------------------------------------

2017-12-29T22:12:20Z asimshankar

Seems like this is resolved (and the root cause was a version mismatch of CUDA - 9.0 vs 9.1)?

Closing this out since I understand it to be resolved, but please let me know if I'm mistaken.

FYI @gunan @av8ramit (who are working on the upcoming 1.5 release)

-------------------------------------------------------------------------

2018-01-03T11:20:52Z yangfengKAUST

I also occur the exactly same problem with kirk86. For me, I installed cuda toolkit 8.0, and cudnn 5.1.

Then I did what you guys said above, all of them does not work.

-------------------------------------------------------------------------

2018-01-03T21:39:35Z gunan

For using nightlies, you have to have CUDA 9.0 and cudnn 7 installed.

@yangfengKAUST with the current version of cuda and cudnn installed TF is just complaining that it cannot find the versions it is expecting.

-------------------------------------------------------------------------

2018-01-09T02:57:46Z zhugejun

@Timonzimm I am facing the same issue. Have you figured it out?

-------------------------------------------------------------------------

2018-01-17T22:15:22Z gbolcer

I have 8.0, 9.0, 9.1 installed + cudnn versions which seem specific to each. The sym linking didn't work from the 9.1 libs. I suspect that sometimes the symlink in the LD\_LIBRARY\_PATH doesn't work either when I switch versions on the /usr/local/cuda link. I ended up just doing it the low tech way to get the libraries loaded into my java program until I can figure out a cleaner way to handle the paths inside of Eclipse.

try {

System.load("/usr/local/cuda/lib64/libcublas.so.9.0");

System.load("/usr/local/cuda/lib64/libcusolver.so.9.0");

System.load("/usr/local/cuda/lib64/libcudart.so.9.0");

System.load("/usr/local/cuda/lib64/libcufft.so.9.0");

System.load("/usr/local/cuda/lib64/libcurand.so.9.0");

System.load("/home/greg/Desktop/platform/tensorbuilder/jni/libtensorflow\_jni.so");

} catch (UnsatisfiedLinkError e) {

System.err.println("Native code library failed to load.\n" + e);

System.exit(1);

}

.

-------------------------------------------------------------------------

2018-01-18T18:25:15Z AwasthiMaddy

@asimshankar Would like to know that in your above comment you mean that we should downgrade cuda to 9.0 and tensorflow 1.5 doesn't work with cuda 9.1 ?

You have closed this issue but its not clear what is the correct action that we should take!

Note: I also have cuda 9.1 installed instead of cuda 9.0.

-------------------------------------------------------------------------

2018-01-18T18:53:49Z gbolcer

Just FYI, I have both installed. Building from scratch will work w/ either, but the nightly binaries use 9.0.

-------------------------------------------------------------------------

2018-01-18T19:01:05Z asimshankar

@AwasthiMaddy - Yes TensorFlow 1.5 release binaries are built for CUDA 9.

-------------------------------------------------------------------------

2018-01-30T02:47:28Z aipeteryao

Have you solved it ? This problem is caused tensorflow-gpu-1.5 required cuda 9.0 ,so you should install tensorflow-gpu-1.4. And rember uninstall tensorflow-gpu-1.5. Please use this"pip install --upgrade tensorflow-gpu==1.4"

-------------------------------------------------------------------------

2018-02-02T09:22:51Z nitinagarwal

@aipeteryao - Thank you.

-------------------------------------------------------------------------

2018-02-02T16:41:40Z bwesen

Someone needs to fix the https://www.tensorflow.org/install/install\_linux page if this is true, I just followed its instructions exactly, and tells you to install CUDA 8.0 (specifically, not "latest CUDA").

Then as soon as you're done, you get this error (it is looking for cublas 9.0, which, from what I can read here, would not have worked either, as CUDA 9.1 is the default you get from NVIDIA).

Either the webpage instructions should work with the default latest of everything, or it should tell you explicitly to install tensorflow-gpu-1.4 (for example) and not tensorflow-gpu..

-------------------------------------------------------------------------

2018-02-02T23:50:32Z yazabaza

Seconding bwesons's comment. I have CUDA 8.0 and Tensorflow 1.3. I followed the current install instructions for TF 1.5 (GPU, ubuntu, virtualenv) and it breaks as described above. Reverting to TF 1.3 until this is resolved.

-------------------------------------------------------------------------

2018-02-03T00:40:49Z AustinSong

@aipeteryao This fixed it, thanks! I ended up uninstalling the latest version and installing 1.4, in my virtualenv.

```

pip3 uninstall tensorflow-gpu

pip3 install --upgrade tensorflow-gpu==1.4

```

The install page for Ubuntu should be updated: https://www.tensorflow.org/install/install\_linux

Since TensorFlow 1.5 is expecting Cuda 9.0 ( NOT 9.1 ), as well as cuDNN 7

-------------------------------------------------------------------------

2018-02-03T05:40:22Z aipeteryao

In fact, we should view the official document of tensorflow ,it give tensorflow‚Äòs envirment(include python,gcc,cuda,cudnn,an so on).

-------------------------------------------------------------------------

2018-02-05T14:17:29Z aipeteryao

@bwesen yes,you were right .My computer installed CUDA 8.0,cudnn 6.0 ,tensorflow 1.4.

-------------------------------------------------------------------------

2018-02-08T15:29:53Z robhawkins

I think this issue should still be open. @bwesen's [comment](https://github.com/tensorflow/tensorflow/issues/15604#issuecomment-362637994) is correct. [The docs](http://archive.is/S1P6z) tell you to install Cuda 8.0 and use `pip install --upgrade tensorflow-gpu`. Right now that gives you tensorflow 1.5 which does not work with Cuda 8.0

pinging @asimshankar

-------------------------------------------------------------------------

2018-02-12T11:29:51Z mkaze

I have the same issue (with cuda 9.1 + tensorflow 1.5). I think to resolve it, one option is that to downgrade cuda to 9.0. The other option would be to downgrade both cuda to 8.0 and tensorflow to 1.4. If you have already installed cuda 8.0, you only need to modify `LD\_LIBRARY\_PATH` (and `CUDA\_HOME`) environment variable to point to cuda 8.0 directory (i.e. `/usr/local/cuda-8.0`).

-------------------------------------------------------------------------

2018-02-16T04:50:38Z DylanDmitri

I'm getting this issue (Cuda 9.1.85, cuDNN 7.05)

Tried with tensorflow 1.5, it broke. Uninstalled, installed 1.4 with ```pip3 install --upgrade tensorflow-gpu==1.4```, still broke.

-------------------------------------------------------------------------

2018-02-16T05:15:52Z AustinSong

@DylanDmitri 1.5 expects Cuda 9.0, not 9.1

Have you tried with Cuda 9.0 drivers?

-------------------------------------------------------------------------

2018-02-16T05:16:42Z robhawkins

@DylanDmitri @mkaze You need Cuda 9.0.

Also, for anyone having trouble installing requirements, I suggest double checking your cuDNN installation. The .deb file didn't work for me because it did not copy files to the right place. I had to use the .tgz file and manually copy files according to nVidia's directions in order to get a working installation.

-------------------------------------------------------------------------

2018-02-25T06:21:54Z abrahamrhoffman

Why not just install cuda-9-0?

- Go here: https://developer.nvidia.com/cuda-90-download-archive

- Then, for me: Download deb (network)

```

sudo dpkg -i¬†cuda-repo-ubuntu1604\_9.0.176-1\_amd64.deb

sudo apt-key adv --fetch-keys \

http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86\_64/7fa2af80.pub

sudo apt-get update

sudo apt-get install cuda-9-0

```

-------------------------------------------------------------------------

2018-02-25T17:28:37Z kirk86

> Why not just install cuda-9-0?

@abrahamrhoffman That's easy for anyone who has sudo privileges but what about people on a shared system like a cluster environment with simple user privileges. In those cases even if you ask from the sys admin to install any libraries most probably the answer is gonna be NO! Since they are afraid that might interfere with other users' settings and environments.

@abrahamrhoffman Would you also mind providing a justification on the down vote?

-------------------------------------------------------------------------

2018-02-26T02:33:58Z rjurney

I installed cuda-9.0 and still it does not work. This is really irritating.

-------------------------------------------------------------------------

2018-02-26T02:43:52Z abrahamrhoffman

Please make sure to set your PATH variable appropriately, such as described here: https://stackoverflow.com/questions/39287744/ubuntu-16-04-nvidia-toolkit-8-0-rc-darknet-compilation-error-expected-a/41290056#41290056

```

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

export LD\_LIBRARY\_PATH=/usr/local/cuda9.0/lib64${LD\_LIBRARY\_PATH:+:${LD\_LIBRARY\_PATH}}

```

-------------------------------------------------------------------------

2018-02-26T14:35:23Z TensorTom

cuda 9.1 is the current version. I want TF to use it. How to?

-------------------------------------------------------------------------

2018-02-26T14:40:21Z TensorTom

Even tf-nightly-gpu is not looking for cuda 9.1. Meh

-------------------------------------------------------------------------

2018-02-26T15:05:36Z TensorTom

I tried symbolic links from all the 9.0 filenames to all the 9.1 filenames and it didn't work. In the end, TF knows the true version. The repo doesn't even have 9.0 anymore so I'm afraid I'll break my nvidia stuff if I remove 9.1 and then manually install 9.0.

-------------------------------------------------------------------------

2018-02-26T15:15:36Z TensorTom

I fix him for now by:

Download deb (network) from: https://developer.nvidia.com/cuda-90-download-archive?target\_os=Linux&target\_arch=x86\_64&target\_distro=Ubuntu&target\_version=1604&target\_type=debnetwork

Then: `dpkg -i cuda-repo-ubuntu1604\_9.0.176-1\_amd64.deb`

Then: aptitude update

Then: aptitude install cuda-9-0

-------------------------------------------------------------------------

2018-02-27T13:06:48Z xiezhongzhao

First I've installed tensorflow 1.5, it broke, and I get the following error:

`ImportError: libcublas.so.9.0: cannot open shared object file: No such file or directory`

then I uninstalled, installed 1.4 with pip install --upgrade tensorflow-gpu==1.4, it did't work, and I get the following error:

`ImportError: libcublas.so.8.0: cannot open shared object file: No such file or directory`

-------------------------------------------------------------------------

2018-02-27T14:00:50Z mkaze

@xiezhongzhao What version of Cuda are you using? For tensorflow 1.5 you must have installed the Cuda 9.0 and for tensorflow 1.4 you must use cuda 8.0. If the the tensorflow version and cuda version are compatible, then check the environment variables i.e. `CUDA\_HOME `and `LD\_LIBRARY\_PATH`.

-------------------------------------------------------------------------

2018-02-27T14:09:57Z xiezhongzhao

@mkaze I used Cuda9.1

-------------------------------------------------------------------------

2018-02-27T14:24:24Z mkaze

@xiezhongzhao Install Cuda 9.0 and you should be fine. Tensorflow 1.5 does not work with Cuda 9.1.

-------------------------------------------------------------------------

2018-02-28T03:07:37Z xiezhongzhao

@mkaze Thank you very much

-------------------------------------------------------------------------

2018-03-02T05:52:49Z andrewjtimmons

I am also getting this issue and struggling to resolve it.

```

$ pip3 install tensorflow-gpu

$ python3

>>> import tensorflow

Traceback (most recent call last):

File "/home/ubuntu/.local/lib/python3.5/site-packages/tensorflow/python/pywrap\_tensorflow.py", line 58, in <module>

from tensorflow.python.pywrap\_tensorflow\_internal import \*

File "/home/ubuntu/.local/lib/python3.5/site-packages/tensorflow/python/pywrap\_tensorflow\_internal.py", line 28, in <module>

\_pywrap\_tensorflow\_internal = swig\_import\_helper()

File "/home/ubuntu/.local/lib/python3.5/site-packages/tensorflow/python/pywrap\_tensorflow\_internal.py", line 24, in swig\_import\_helper

\_mod = imp.load\_module('\_pywrap\_tensorflow\_internal', fp, pathname, description)

File "/usr/lib/python3.5/imp.py", line 242, in load\_module

return load\_dynamic(name, filename, file)

File "/usr/lib/python3.5/imp.py", line 342, in load\_dynamic

return \_load(spec)

ImportError: libcublas.so.9.0: cannot open shared object file: No such file or directory

```

I installed following these instructions

https://www.tensorflow.org/install/install\_linux#nvidia\_requirements\_to\_run\_tensorflow\_with\_gpu\_support

I believe I installed the right versions from nvidia. filenames were

`cuda-repo-ubuntu1604\_9.0.176-1\_amd64.deb`

and `cudnn-9.0-linux-x64-v7.tgz` (version 7.0.5)

I set the path as per those instructions on the tensorflow docs and also tried the instructions that `abrahamrhoffman` gave above.

When I run ldconfig -v I get some 9.0 libs, but do not see libcublas.so.9.0

```

/usr/local/cuda-9.0/targets/x86\_64-linux/lib:

libaccinj64.so.9.0 -> libaccinj64.so.9.0.176

libOpenCL.so.1 -> libOpenCL.so.1.0.0

libnvToolsExt.so.1 -> libnvToolsExt.so.1.0.0

libcuinj64.so.9.0 -> libcuinj64.so.9.0.176

libcudart.so.9.0 -> libcudart.so.9.0.176

/usr/local/cuda-9.1/targets/x86\_64-linux/lib:

libnppif.so.9.1 -> libnppif.so.9.1.85

libcusparse.so.9.1 -> libcusparse.so.9.1.85

libcusolver.so.9.1 -> libcusolver.so.9.1.85

libnpps.so.9.1 -> libnpps.so.9.1.85

libnppial.so.9.1 -> libnppial.so.9.1.85

libnvgraph.so.9.1 -> libnvgraph.so.9.1.85

libcuinj64.so.9.1 -> libcuinj64.so.9.1.85

libaccinj64.so.9.1 -> libaccinj64.so.9.1.85

libnppicc.so.9.1 -> libnppicc.so.9.1.85

libcudart.so.9.1 -> libcudart.so.9.1.85

libnppc.so.9.1 -> libnppc.so.9.1.85

libnppicom.so.9.1 -> libnppicom.so.9.1.85

libOpenCL.so.1 -> libOpenCL.so.1.0.0

libnppig.so.9.1 -> libnppig.so.9.1.85

libnvToolsExt.so.1 -> libnvToolsExt.so.1.0.0

libnppidei.so.9.1 -> libnppidei.so.9.1.85

libcurand.so.9.1 -> libcurand.so.9.1.85

libnvblas.so.9.1 -> libnvblas.so.9.1.128

libnvrtc.so.9.1 -> libnvrtc.so.9.1.85

libnppitc.so.9.1 -> libnppitc.so.9.1.85

libnppist.so.9.1 -> libnppist.so.9.1.85

libcublas.so.9.1 -> libcublas.so.9.1.128

libnppim.so.9.1 -> libnppim.so.9.1.85

/sbin/ldconfig.real: /usr/local/cuda-9.1/targets/x86\_64-linux/lib/libcudnn.so.7 is not a symbolic link

libcudnn.so.7 -> libcudnn.so.7.0.5

libcufftw.so.9.1 -> libcufftw.so.9.1.85

libcufft.so.9.1 -> libcufft.so.9.1.85

libnppisu.so.9.1 -> libnppisu.so.9.1.85

libnvrtc-builtins.so.9.1 -> libnvrtc-builtins.so.9.1.85

```

I did not install 9.1, at least not intentionally. This is on a amazon ec2 instance with stock ubuntu 16.04.

nvidia-smi also returns a gpu, this is a g3.4xlarge instance

any guidance is greatly appreciated.

-------------------------------------------------------------------------

2018-03-02T06:43:47Z entropy43

Per the CUDNN guide at:

http://docs.nvidia.com/deeplearning/sdk/cudnn-install/index.html

You need to copy the unpacked files (from the directory you ran `$ tar -xzvf cudnn-9.0-linux-x64-v7.tgz` or similar) into `/usr/local/cuda` subdirectories:

`$ sudo cp cuda/include/cudnn.h /usr/local/cuda/include`

`$ sudo cp cuda/lib64/libcudnn\* /usr/local/cuda/lib64`

`$ sudo chmod a+r /usr/local/cuda/include/cudnn.h`

`/usr/local/cuda/lib64/libcudnn\*`

-------------------------------------------------------------------------

2018-03-02T16:20:45Z andrewjtimmons

Thank you for the reply @entropy43.

I should have been more specific. I did those two `cp` and the `chmod` commands after `tar`. When I look in the folder cuda folder for where I ran the tar command like `ls cuda/lib64` I see

```libcudnn.so libcudnn.so.7 libcudnn.so.7.0.5 libcudnn\_static.a```

I tried this section as well from the nvidia [doc](http://docs.nvidia.com/deeplearning/sdk/cudnn-install/index.html)

`2.3.2. Installing from a Debian File`

and the tests pass here in `2.4. Verifying`

Any advice to get that lib is appreciated

-------------------------------------------------------------------------

2018-03-06T16:17:24Z yazabaza

Could someone please summarize where this currently stands? I am using TF1.3 + cuda 8 with no problem. I'd like to upgrade but it seems like the install process for newer versions is completely broken. Advice?

-------------------------------------------------------------------------

2018-03-06T18:45:13Z dartdog

Similar to what yazabazra is asking above:

TF1.6 Ubuntu 16.04

nvcc -V

nvcc: NVIDIA (R) Cuda compiler driver

Copyright (c) 2005-2017 NVIDIA Corporation

Built on Fri\_Nov\_\_3\_21:07:56\_CDT\_2017

Cuda compilation tools, release 9.1, V9.1.85

Which requited a Nvidia display driver 390+

Critical to see: https://devtalk.nvidia.com/default/topic/1000340/cuda-setup-and-installation/-quot-nvidia-smi-has-failed-because-it-couldn-t-communicate-with-the-nvidia-driver-quot-ubuntu-16-04/post/5243047/#5243047

whelp to add to it all,, After a major amount of hassle I got the Nvidia updated to the newest release see above, as the TF doc indicated that there were bugs in an earlier release..

Now I'm getting the :

ImportError: libcublas.so.9.0: cannot open shared object file: No such file or directory

Failed to load the native TensorFlow runtime.

Which appears to be a mismatch between 9.0 (TF wants) vs 9.1 Which is most current Nvidia.

It would seem better to run with 9.1 but I'd rather avoid building TF from source and it seems that may not fix it anyhow..

Can this combo be made to work with a binary package?

TF 1.6 Cuda 9.1 ??

-------------------------------------------------------------------------

2018-03-06T19:26:39Z dartdog

Further note and caution to those looking here.. after upgrading my Nvidia stuff my older versions of TF in separate conda env's no longer work as the older TF wants : ImportError: libcublas.so.8.0: cannot open shared object file: No such file or directory

so 9.1 won't cut it.. how about specifying greater than?? vs specific versions? just a suggestion.. In the meantime I'm dead in the water..

-------------------------------------------------------------------------

2018-03-06T19:40:54Z dartdog

And this is why availability of a binary that supports 9.1 would be nice: (from the TF1.6 release notes)

Using XLA:GPU with CUDA 9 and CUDA 9.1 results in garbage results and/or

CUDA\_ILLEGAL\_ADDRESS failures.

Google discovered in mid-December 2017 that the PTX-to-SASS compiler in CUDA 9

and CUDA 9.1 sometimes does not properly compute the carry bit when

decomposing 64-bit address calculations with large offsets (e.g. load [x + large\_constant]) into 32-bit arithmetic in SASS.

As a result, these versions of ptxas miscompile most XLA programs which use

more than 4GB of temp memory. This results in garbage results and/or

CUDA\_ERROR\_ILLEGAL\_ADDRESS failures.

A fix in CUDA 9.1.121 is expected in late February 2018. We do not expect a

fix for CUDA 9.0.x. Until the fix is available, the only workaround is to

downgrade to CUDA 8.0.x

or disable XLA:GPU.

Maybe one of the nightlies does it?

-------------------------------------------------------------------------

2018-03-06T20:17:05Z dartdog

Another solution? can one install multiple revisions of Cuda since TF seems to search for specific Rev's? If so, any advice as to how to?

So Ideally I'd be able to to run TF 1.4(which currently requires Cuda 8.0) in one conda environment and TF 1.6 (which currently requires Cuda 9.0) in another?

-------------------------------------------------------------------------

2018-03-06T20:56:12Z dartdog

So I just added sudo apt-get -y install cuda-toolkit-9.0 and I'm up and running with TF1.6

-------------------------------------------------------------------------

2018-03-08T09:47:48Z alexattia

@dartdog after installing cuda-toolkit-9.0, did you face the issue `ImportError: libcudnn.so.7: cannot open shared object file: No such file or directory` ?

-------------------------------------------------------------------------

2018-03-09T16:29:12Z abrahamrhoffman

@dartdog

```

sudo apt-get install cuda-7-0

vim ~/.bashrc

export PATH=/usr/local/cuda-7.0/bin${PATH:+:${PATH}}

export LD\_LIBRARY\_PATH=/usr/local/cuda7.0/lib64${LD\_LIBRARY\_PATH:+:${LD\_LIBRARY\_PATH}}

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

export LD\_LIBRARY\_PATH=/usr/local/cuda9.0/lib64${LD\_LIBRARY\_PATH:+:${LD\_LIBRARY\_PATH}}

```

-------------------------------------------------------------------------

2018-03-11T12:20:30Z cubetastic33

This is definitely supposed to be included in the tensorflow documentation, as said by @bwesen. It should also be included in the errors list. Is it possible for us do this in anyway? I think I am going to open a new issue, requesting to either add support for cuda 9.1, or mention that you need cuda 9.0 specifically in the docs, and also include this in the errors list

-------------------------------------------------------------------------

2018-03-11T12:47:27Z cubetastic33

Ok, guys. I have now opened a new issue at https://github.com/tensorflow/tensorflow/issues/17629.

-------------------------------------------------------------------------

2018-03-11T16:54:54Z mldm4

I was handling with this issue as well. What worked for me with tensorflow-gpu 1.6:

- I downloaded the toolkit from the [archive](https://developer.nvidia.com/cuda-toolkit-archive) as 9.0 but it got installed as 9.1 (I do not know why...)

- Still not found libcublas.so.9.0

- Run: `sudo apt-get install cuda-libraries-9-0` as suggested at the end of the installation instructions.

- The issue seems to have been solved.

-------------------------------------------------------------------------

2018-03-11T17:01:54Z qysnn

If you want to have tensorflow work with your CUDA version, you need to first uninstall it then compile it from source and specify the CUDA version while running ./configure

Detailed information can be found here: [https://www.tensorflow.org/install/install\_sources](url)

-------------------------------------------------------------------------

2018-03-11T17:32:49Z pascalwhoop

I am trying this (which builds tensorflow manually)[link](http://www.python36.com/install-tensorflow141-gpu/)

Might take a while longer but you can define the minor versions this way.

-------------------------------------------------------------------------

2018-03-12T02:13:09Z cubetastic33

@mldm4 actually, the command `sudo apt-get install cuda` probably installed 9.1 for you because you also had that in your system. I had the same problem, and I did `sudo apt-get install cuda-9-0` to install a specific version (I had also downloaded from the archive).

I think the commad you did (`sudo apt-get install cuda-libraries-9-0`) also downloads cuda 9.0.

This issue is not solved, as I mentioned at issue 17629, I know the problem is that TensorFlow expects version 9.0, while I had 9.1. The issue is to mention this more clearly in the docs and include it in the common installation problems in the bottom of the docs, or update tensorflow to accept Cuda 9.1.

-------------------------------------------------------------------------

2018-03-12T20:11:11Z RoboEvangelist

so no solution yet?

-------------------------------------------------------------------------

2018-03-12T21:12:30Z abrahamrhoffman

@thread :

\*\*\*\*Please read through the posts carefully! The answer is posted.\*\*\*\*

It is your job to read the thread, and discover the solution; not simply scroll to the end.

-------------------------------------------------------------------------

2018-03-12T21:51:27Z RoboEvangelist

@abrahamrhoffman that's rude.

I just changed my batchrc from cuda-9.1 to just cuda. Then my tensorflow is able of finding the libcublas.so.9.0

-------------------------------------------------------------------------

2018-03-16T15:14:27Z DanlanChen

just fyi: nvidia website for downloading cuda-9.0 is actually downloading cuda-9.1. https://developer.nvidia.com/cuda-90-download-archive?target\_os=Linux&target\_arch=x86\_64&target\_distro=Ubuntu&target\_version=1604&target\_type=deblocal

-------------------------------------------------------------------------

2018-03-16T15:34:12Z cubetastic33

@DanlanChen That is probably because you also have 9.1 installed. To install 9.0, in the steps to download, do `sudo apt-get install cuda-9-0` instead of `sudo apt-get install cuda`.

-------------------------------------------------------------------------

2018-03-16T15:35:22Z DanlanChen

@cubetastic33 I downgraded my tensorflow version and use cuda8.0, thank you anyway.

-------------------------------------------------------------------------

2018-03-16T15:39:52Z cubetastic33

@DanlanChen but then, I guess it is preffered to use the latest version. So, if you ever want to upgrade, you now know what to do! :smile:

-------------------------------------------------------------------------

2018-03-21T02:09:58Z magick93

Im facing the same issue, but I am trying to run tensorflow using nvidia-docker. I have cuda-9-0 installed on the host, but when I try to run my docker container I get

`ImportError: libcuda.so.1: cannot open shared object file: No such file or directory`

-------------------------------------------------------------------------

2018-03-21T19:39:00Z Oktai15

@magick93 and all that turn up here!

LISTEN! Anything you need is downgrade your cuda 9.1 -> cuda 9.0. That's it! Just do it (if you downloaded cuda 9.1 before that you can execute following command in your terminal):

`sudo apt-get install cuda-9-0` and remove cuda 9.1 by rm -rf.

Btw, don't forget to change $PATH in your `~/.bashrc` (9.1 -> 9.0).

-------------------------------------------------------------------------

2018-03-21T22:49:14Z magick93

Hi @Oktai15

> LISTEN! Anything you need is downgrade your cuda 9.1 -> cuda 9.0. That's it! Just do it

Yes, I have done this - many times.

```

sudo apt-get install cuda-9-0

[sudo] password for anton:

Reading package lists... Done

Building dependency tree

Reading state information... Done

cuda-9-0 is already the newest version (9.0.176-1).

0 upgraded, 0 newly installed, 0 to remove and 7 not upgraded.

```

```

$ ldconfig -p | grep cuda

libnvrtc.so.9.0 (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libnvrtc.so.9.0

libnvrtc.so (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libnvrtc.so

libnvrtc-builtins.so.9.0 (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libnvrtc-builtins.so.9.0

libnvrtc-builtins.so (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libnvrtc-builtins.so

libnvgraph.so.9.0 (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libnvgraph.so.9.0

libnvgraph.so (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libnvgraph.so

libnvblas.so.9.0 (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libnvblas.so.9.0

libnvblas.so (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libnvblas.so

libnvToolsExt.so.1 (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libnvToolsExt.so.1

libnvToolsExt.so (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libnvToolsExt.so

libnpps.so.9.0 (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libnpps.so.9.0

libnpps.so (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libnpps.so

libnppitc.so.9.0 (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libnppitc.so.9.0

libnppitc.so (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libnppitc.so

libnppisu.so.9.0 (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libnppisu.so.9.0

libnppisu.so (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libnppisu.so

libnppist.so.9.0 (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libnppist.so.9.0

libnppist.so (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libnppist.so

libnppim.so.9.0 (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libnppim.so.9.0

libnppim.so (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libnppim.so

libnppig.so.9.0 (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libnppig.so.9.0

libnppig.so (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libnppig.so

libnppif.so.9.0 (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libnppif.so.9.0

libnppif.so (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libnppif.so

libnppidei.so.9.0 (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libnppidei.so.9.0

libnppidei.so (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libnppidei.so

libnppicom.so.9.0 (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libnppicom.so.9.0

libnppicom.so (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libnppicom.so

libnppicc.so.9.0 (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libnppicc.so.9.0

libnppicc.so (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libnppicc.so

libnppial.so.9.0 (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libnppial.so.9.0

libnppial.so (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libnppial.so

libnppc.so.9.0 (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libnppc.so.9.0

libnppc.so (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libnppc.so

libicudata.so.55 (libc6,x86-64) => /usr/lib/x86\_64-linux-gnu/libicudata.so.55

libcusparse.so.9.0 (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libcusparse.so.9.0

libcusparse.so (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libcusparse.so

libcusolver.so.9.0 (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libcusolver.so.9.0

libcusolver.so (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libcusolver.so

libcurand.so.9.0 (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libcurand.so.9.0

libcurand.so (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libcurand.so

libcuinj64.so.9.0 (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libcuinj64.so.9.0

libcuinj64.so (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libcuinj64.so

libcufftw.so.9.0 (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libcufftw.so.9.0

libcufftw.so (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libcufftw.so

libcufft.so.9.0 (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libcufft.so.9.0

libcufft.so (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libcufft.so

libcudnn.so.7 (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libcudnn.so.7

libcudart.so.9.0 (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libcudart.so.9.0

libcudart.so.7.5 (libc6,x86-64) => /usr/lib/x86\_64-linux-gnu/libcudart.so.7.5

libcudart.so (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libcudart.so

libcudart.so (libc6,x86-64) => /usr/lib/x86\_64-linux-gnu/libcudart.so

libcuda.so.1 (libc6,x86-64) => /usr/lib/x86\_64-linux-gnu/libcuda.so.1

libcuda.so (libc6,x86-64) => /usr/lib/x86\_64-linux-gnu/libcuda.so

libcublas.so.9.0 (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libcublas.so.9.0

libcublas.so (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libcublas.so

libaccinj64.so.9.0 (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libaccinj64.so.9.0

libaccinj64.so (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libaccinj64.so

libOpenCL.so.1 (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libOpenCL.so.1

libOpenCL.so (libc6,x86-64) => /usr/local/cuda-9.0/targets/x86\_64-linux/lib/libOpenCL.so

```

Yet I cant even run this simple tensorflow script, as it results I get `ImportError: No module named tensorflow.python.client`

```

from tensorflow.python.client import device\_lib

def get\_available\_gpus():

local\_device\_protos = device\_lib.list\_local\_devices()

return [x.name for x in local\_device\_protos if x.device\_type == 'GPU']

get\_available\_gpus()

```

-------------------------------------------------------------------------

2018-03-23T02:49:07Z ammarhameed

I had the same issue everybody is having here, installing 9.0 did help but then I got other issues.

Then I followed [this](https://github.com/williamFalcon/tensorflow-gpu-install-ubuntu-16.04) and it finally worked for me.

-------------------------------------------------------------------------

2018-03-23T04:11:28Z cubetastic33

@Oktai15 doesn't `rm -rf` delete your system? Please be more clear here, because people might try it without going into the required directory, and end up emptying their home folder.

-------------------------------------------------------------------------

2018-03-23T04:12:41Z cubetastic33

@magick93 your issue seems to be something else, not the CUDA version.

-------------------------------------------------------------------------

2018-03-23T14:51:07Z Abduoit

I had same issue, I think I solved by some changing, this combination works for me

Ubuntu 16, cuda 9.0, cudnn 7.0, python 3,5, tensorflow 1.6

If you install new cuda while you still have previous version please make sure to specify the path like this

```

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

export LD\_LIBRARY\_PATH=/usr/local/cuda-9.0/lib64${LD\_LIBRARY\_PATH:+:${LD\_LIBRARY\_PATH}}

```

-------------------------------------------------------------------------

2018-03-24T14:01:51Z cubetastic33

@Abduoit your Ubuntu version doesn't really matter. The thing is, TensorFlow 1.6 expects CUDA to be version 9.0, and cuDNN to be version 7.0.4 (yes, the 0.4 \_does\_ matter)

-------------------------------------------------------------------------

2018-03-30T20:33:34Z SAGGSOC

Traceback (most recent call last):

File "utils.py", line 15, in <module>

import tensorflow as tf

File "/home/sagar/miniconda2/lib/python2.7/site-packages/tensorflow/\_\_init\_\_.py", line 24, in <module>

from tensorflow.python import \*

File "/home/sagar/miniconda2/lib/python2.7/site-packages/tensorflow/python/\_\_init\_\_.py", line 49, in <module>

from tensorflow.python import pywrap\_tensorflow

File "/home/sagar/miniconda2/lib/python2.7/site-packages/tensorflow/python/pywrap\_tensorflow.py", line 74, in <module>

raise ImportError(msg)

ImportError: Traceback (most recent call last):

File "/home/sagar/miniconda2/lib/python2.7/site-packages/tensorflow/python/pywrap\_tensorflow.py", line 58, in <module>

from tensorflow.python.pywrap\_tensorflow\_internal import \*

File "/home/sagar/miniconda2/lib/python2.7/site-packages/tensorflow/python/pywrap\_tensorflow\_internal.py", line 28, in <module>

\_pywrap\_tensorflow\_internal = swig\_import\_helper()

File "/home/sagar/miniconda2/lib/python2.7/site-packages/tensorflow/python/pywrap\_tensorflow\_internal.py", line 24, in swig\_import\_helper

\_mod = imp.load\_module('\_pywrap\_tensorflow\_internal', fp, pathname, description)

ImportError: libcublas.so.9.0: cannot open shared object file: No such file or directory

Failed to load the native TensorFlow runtime.

See https://www.tensorflow.org/install/install\_sources#common\_installation\_problems

sudo apt-get install cuda-7-0

vim ~/.bashrc

export PATH=/usr/local/cuda-7.0/bin${PATH:+:${PATH}}

export LD\_LIBRARY\_PATH=/usr/local/cuda7.0/lib64${LD\_LIBRARY\_PATH:+:${LD\_LIBRARY\_PATH}}

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

export LD\_LIBRARY\_PATH=/usr/local/cuda9.0/lib64${LD\_LIBRARY\_PATH:+:${LD\_LIBRARY\_PA

for some common reasons and solutions. Include the entire stack trace

above this error message when asking for help.

-------------------------------------------------------------------------

2018-03-31T02:11:32Z DaniloBarros

As @pascalwhoop mentioned, I followed the instructions in here [http://www.python36.com/install-tensorflow141-gpu/](http://www.python36.com/install-tensorflow141-gpu/) to build Tensorflow from source.

Whenever it said cuda 9.0 I changed to 9.1, and cudnn 7.0 I put 7.1.2.

Worked fine, so far!

-------------------------------------------------------------------------

2018-03-31T02:53:13Z cubetastic33

@SAGGSOC why are you installing cuda 7.0? You need CUDA 9.0 and CuDNN 7.0.4

-------------------------------------------------------------------------

2018-03-31T07:38:51Z pascalwhoop

Everyone who's been having issues: who's up for turning this into a solid docker image that's shared with the community. Rather do a 6gb image pull once that works than DLing 5 versions of CuDNN before stuff works..

https://github.com/pascalwhoop/tf\_openailab\_gpu\_docker

I started a while back but stopped because of shifting project focuses. But I think it's worth pursuing. Keeps the whole trouble of finding a the right combination of 17 moving parts away from most ppl.

-------------------------------------------------------------------------

2018-03-31T15:46:49Z kirk86

Just to clarify a few things for anyone who might stumble on this post. I have in my system installed cuda-8.0, cuda-9.0, cuda-9.1. You don't have to remove anything to make it work with tensorflow. Instead if you are missing cuda-9.0 from your system, as other have already pointed then you'll need to install it that is a prerequisite for tensorflow to work properly.

If you have cuda-9.0 installed on you system and tensorflow is complaining about `libcublas.so.9.0` again as others have said expose that during runtime through your `LD\_LIBRARY\_PATH` environment variable in your `.bashrc` make it point to `/usr/local/cuda-9.0/lib64`. This should be working even for tensorflow 1.7.

What I have tried and failed to accomplish is build from source. For some reason bazel always exits with an error. If you try to build with cuda-9.0/cuda-9.1 and cudnn7 it complains about gcc7. Using gcc5 compilation seems to be working fine but then at the end I always get an error and the build is unsuccessful.

My question is if anyone has managed to compile from source with cuda-9.1/cuda-9.0 without problems?

-------------------------------------------------------------------------

2018-04-01T05:32:46Z etemiz

This worked for me:

Download CUDA Toolkit 9.0 from NVidia previous releases section.

Then:

```

sudo dpkg -i¬†cuda-repo-ubuntu1604-9-0-local\_9.0.176-1\_amd64.deb

sudo apt-key add /var/cuda-repo-9-0-local/7fa2af80.pub

sudo apt-get update

sudo apt-get install cuda-9.0

```

Notice 9.0 at the last line above.

`export LD\_LIBRARY\_PATH=/usr/local/cuda-9.0/lib64

`

My setup

Tensorflow 1.7

cuDNN 7.1.2

Ubuntu 16.04

-------------------------------------------------------------------------

2018-04-04T21:48:29Z AndrewVGr

etemiz,

I've got the same setup and it works for me too.

Thank you for your post :)

-------------------------------------------------------------------------

2018-04-06T12:26:33Z alexdominguez09

I had teh same situation. I had cuda 9.1, and tensorflow would not find libraries for cuda 9.0.

I have installed cuda 9.0 with command: sudo apt-get install cuda-libraries-9-0

That solved my problem.

-------------------------------------------------------------------------

2018-04-10T15:07:19Z sjsy

With cuda 8 and 9.0 installed, setting `LD\_LIBRARY\_PATH` in `.bashrc` and `.profile` not work. So, I set

`LD\_LIBRARY\_PATH=/usr/local/cuda-9.0/lib64` in pycharm Environment variable field and it works.

-------------------------------------------------------------------------

2018-04-12T04:11:12Z buysilver

the above worked for installing on Ubuntu Server 17. namely,

- installing cuda-9.0 (NOT 9.1)

- cuDNN v7.1.2 (Mar 21, 2018) for CUDA 9.0

- everything else according to the official tf installation instructions

much easier than compiling.

-------------------------------------------------------------------------

2018-04-14T03:22:29Z rpetrenko

be careful conda users. i hit the same problem and was scratching my head for two days, until finally i discovered that local copy of libcudnn.so was used by conda, under:

/miniconda3/lib/libcudnn.so which pointed to libcudnn.so.7 which pointed to libcudnn.so.7.0.5

i don't remember who and how placed it there but pretty much it overloaded the system default libcudnn.so.7.1.2 !!!

once removed, everything works like a charm:

tensorflow 1.7 or 1.8-nightly, cuda-9.1, cudnn-7.1.2 on ubuntu 16.04

-------------------------------------------------------------------------

2018-04-29T19:00:34Z Suananda

Somehow, I solved this by installing:

cuda 9.1 (from package manager),

cudnn 7.1 for 9.1

and from anaconda:

by using this default command 'conda install -c anaconda tensorflow-gpu'

cudatoolkit 9.0,

tensorflow 1.7,

tensorflow-gpu 1.7

I used Antergos linux, GTX 1060 in my PC. It worked as well in my notebook (Xubuntu 18.04, GT 840m). In my notebook i used :

cuda 9.1 (from nvidia ppa), cudnn 7.1 for 9.1 (from nvidia web), and the rest was the same

-------------------------------------------------------------------------

2018-04-30T21:35:39Z jzhou316

Thanks @Suananda! It works like magic.

-------------------------------------------------------------------------

2018-05-04T17:44:50Z athlonshi

If you have old version of CUDA, the library link may point to the old library even you install the newer CUDA especially if you install it manually. Try delete your old installation, and then sudo ldconfig to update the dynamic links.

-------------------------------------------------------------------------

2018-05-07T07:24:56Z neelkadia

No solution yet!?

-------------------------------------------------------------------------

2018-05-07T15:34:32Z kirk86

For anyone that might stumble on this I have released a community wheel of latest tensorflow 1.8.0-rc1 built with cuda 9.1. You can find it [here](https://github.com/yaroslavvb/tensorflow-community-wheels/issues/64)!

-------------------------------------------------------------------------

2018-05-14T03:32:42Z NYcleaner

I have find the reason is ldconf, ldconfig is a dynamic link library management command whose purpose is to allow the dynamic link library to be usedby the system.

The default ldconf only search /lib and /usr/lib, as well as the library file under the directory listed in the configuration file /etc/ld. so. conf.

so all of this is caused by the dynamic library of CUDA in the installed CUDA path such as : /path/cuda-9.0/lib64 or /path/cuda-9.0/lib. (for eample my CUDA is installed in /usr/local/cuda-9.0)

1. if you install the CUDA manual, then after install, you should add the path of cuda/lib64 to /etc/ld.so.conf file

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

then

`sudo ldconfig`

of course , you can add the path manual, like:

`vim /etc/ld.so.conf`

then add the path '/usr/local/cuda-9.0/lib64' at the end.

`sudo ldconfig`

after the operation, reopen the ipython or pycharm ,

import tensorflow as tf

wow, you will enjoy it!

2. if you install the CUDA by command such as 'dpkg -i cuda-repo-ubuntu1604\_9.0.176-1\_amd64.deb' or others, it may add the cuda lib path to the /etc/ld.so.conf automatically . but to be on the safe side, check the /etc/ld.so.conf and see if the path add to it .

-------------------------------------------------------------------------

2018-05-16T14:08:40Z sebma

@NYcleaner

On Ubuntu, there is a `/etc/ld.so.conf.d/cuda-9-1.conf` file containing :

`/usr/local/cuda-9.1/targets/x86\_64-linux/lib`

Is this enough or do I need to add the directory `/usr/local/cuda-9.1/lib64/` to it ?

-------------------------------------------------------------------------

2018-05-21T01:22:02Z NYcleaner

@sebma

you should add the `/usr/local/cuda-9.1/lib64/` path to itÔºå the \*\*\*.so files are in the lib64

-------------------------------------------------------------------------

2018-06-01T18:43:29Z evandrix

```

sudo bash -c "echo /usr/local/cuda/lib64/ >/etc/ld.so.conf.d/cuda.conf"

sudo ldconfig

```

from https://gist.github.com/zhanwenchen/e520767a409325d9961072f666815bb8

-------------------------------------------------------------------------

2018-06-04T13:35:10Z kirk86

@mashu Well the other option is that the community provides pre-built [wheels](https://github.com/yaroslavvb/tensorflow-community-wheels). If you read 2-3 threads above you'll also see that mentioned again.

-------------------------------------------------------------------------

2018-06-06T02:53:54Z wl1136

@Suananda Thanks, it works for me in a tensorflow conda environment.

Suggest the official guys to modify the installation guide "https://www.tensorflow.org/install/install\_linux#InstallingAnaconda", step 4 of Anaconda installing, from "pip install --ignore-installed --upgrade tfBinaryURL" to "conda install -c anaconda tensorflow-gpu"

-------------------------------------------------------------------------

2018-06-06T06:13:57Z jinfagang

Softlink seems not solve this issue:

```

‚ûú cuda ls lib64

libaccinj64.so libcufftw\_static.a libnppial.so.9.2 libnppig\_static.a libnvblas.so.9.2

libaccinj64.so.9.2 libcuinj64.so libnppial.so.9.2.88 libnppim.so libnvblas.so.9.2.88

libaccinj64.so.9.2.88 libcuinj64.so.9.2 libnppial\_static.a libnppim.so.9.2 libnvgraph.so

libcublas\_device.a libcuinj64.so.9.2.88 libnppicc.so libnppim.so.9.2.88 libnvgraph.so.9.2

libcublas.so libculibos.a libnppicc.so.9.2 libnppim\_static.a libnvgraph.so.9.2.88

libcublas.so.9.0 libcurand.so libnppicc.so.9.2.88 libnppist.so libnvgraph\_static.a

libcublas.so.9.2 libcurand.so.9.2 libnppicc\_static.a libnppist.so.9.2 libnvrtc-builtins.so

libcublas.so.9.2.88 libcurand.so.9.2.88 libnppicom.so libnppist.so.9.2.88 libnvrtc-builtins.so.9.2

libcublas\_static.a libcurand\_static.a libnppicom.so.9.2 libnppist\_static.a libnvrtc-builtins.so.9.2.88

libcudadevrt.a libcusolver.so libnppicom.so.9.2.88 libnppisu.so libnvrtc.so

libcudart.so libcusolver.so.9.2 libnppicom\_static.a libnppisu.so.9.2 libnvrtc.so.9.2

libcudart.so.9.2 libcusolver.so.9.2.88 libnppidei.so libnppisu.so.9.2.88 libnvrtc.so.9.2.88

libcudart.so.9.2.88 libcusolver\_static.a libnppidei.so.9.2 libnppisu\_static.a libnvToolsExt.so

libcudart\_static.a libcusparse.so libnppidei.so.9.2.88 libnppitc.so libnvToolsExt.so.1

libcufft.so libcusparse.so.9.2 libnppidei\_static.a libnppitc.so.9.2 libnvToolsExt.so.1.0.0

libcufft.so.9.2 libcusparse.so.9.2.88 libnppif.so libnppitc.so.9.2.88 libOpenCL.so

libcufft.so.9.2.88 libcusparse\_static.a libnppif.so.9.2 libnppitc\_static.a libOpenCL.so.1

libcufft\_static.a libnppc.so libnppif.so.9.2.88 libnpps.so libOpenCL.so.1.0

libcufft\_static\_nocallback.a libnppc.so.9.2 libnppif\_static.a libnpps.so.9.2 libOpenCL.so.1.0.0

libcufftw.so libnppc.so.9.2.88 libnppig.so libnpps.so.9.2.88 stubs

libcufftw.so.9.2 libnppc\_static.a libnppig.so.9.2 libnpps\_static.a

libcufftw.so.9.2.88 libnppial.so libnppig.so.9.2.88 libnvblas.so

```

Still got:

```

Traceback (most recent call last):

File "/usr/local/lib/python3.6/dist-packages/tensorflow/python/pywrap\_tensorflow.py", line 58, in <module>

from tensorflow.python.pywrap\_tensorflow\_internal import \*

File "/usr/local/lib/python3.6/dist-packages/tensorflow/python/pywrap\_tensorflow\_internal.py", line 28, in <module>

\_pywrap\_tensorflow\_internal = swig\_import\_helper()

File "/usr/local/lib/python3.6/dist-packages/tensorflow/python/pywrap\_tensorflow\_internal.py", line 24, in swig\_import\_helper

\_mod = imp.load\_module('\_pywrap\_tensorflow\_internal', fp, pathname, description)

File "/usr/lib/python3.6/imp.py", line 243, in load\_module

return load\_dynamic(name, filename, file)

File "/usr/lib/python3.6/imp.py", line 343, in load\_dynamic

return \_load(spec)

ImportError: libcublas.so.9.0: cannot open shared object file: No such file or directory

During handling of the above exception, another exception occurred:

```

-------------------------------------------------------------------------

2018-06-07T08:25:57Z YoungKang1222

when I run my code on the linux environment directly, everything is OK. But when I run on the local pycharm through the remote interpreter, I encounter the problem: `ImportError: libcublas.so.9.0: cannot open shared object file: No such file or directory. "Failed to load the native TensorFlow runtime."`

-------------------------------------------------------------------------

2018-06-12T10:58:18Z Ohmat-Robotics

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

export CUDA\_HOME=${CUDA\_HOME}:/usr/local/cuda:/usr/local/cuda-9.0

export LD\_LIBRARY\_PATH=${LD\_LIBRARY\_PATH}:/usr/local/cuda-9.0/lib64

export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:/usr/local/cuda/extras/CUPTI/lib64

if use pycharm - add it to interpreter

-------------------------------------------------------------------------

2018-06-16T15:44:43Z dongzhuoyao

what about f\*\* I just only want to use tensorflow1.8 and cuda9.1?

-------------------------------------------------------------------------

2018-06-17T12:23:59Z kirk86

@dongzhuoyao So what's the problem? look at my comment 6 threads above and you'll find your solution there!

-------------------------------------------------------------------------

2018-06-19T19:08:04Z mebble

I guess the problem has cropped up again with `cuda 9.2` and `tensorflow-gpu 1.8`. In a virtualenv, I get:

```

>>> import tensorflow as tf

ImportError: libcublas.so.9.0: cannot open shared object file: No such file or directory

```

My `/usr/local/cuda-9.2/lib64` folder has the following libcublas:

```

$ ls /usr/local/cuda-9.2/lib64 | grep libcublas

libcublas\_device.a

libcublas.so

libcublas.so.9.2

libcublas.so.9.2.113

libcublas.so.9.2.88

libcublas\_static.a

```

Versions:

```

Ubuntu 16.04

cuda 9.2

cudnn 7.1.4

tensorflow-gpu 1.8.0

```

We'll need an update to tensorflow-gpu to use cuda 9.2.

Also, if I were to downgrade to cuda 9.0, would I have to first remove cuda 9.2 or just install 9.0 straight away? Would I have conflicting installations?

-------------------------------------------------------------------------

2018-06-19T19:53:06Z kirk86

@mebble Here you go: [link](https://we.tl/b7yAL9rYrO). That's tf 1.8 wheel for cuda 9.2.

> if I were to downgrade to cuda 9.0, would I have to first remove cuda 9.2

Don't downgrade. Install whatever other version you want they'll get installed at `/usr/local/cuda-x.x`

-------------------------------------------------------------------------

2018-06-20T06:25:38Z mebble

Thanks! I forgot to mention that im using `python 3.5.2` and pip `10.0.1`. I think the wheel is for python 3.6 so the install doesn't work. Do you have one for 3.5 as well?

-------------------------------------------------------------------------

2018-06-20T09:10:39Z pavan-08

@kirk86 after installing the whl you gave it throws a similar error for libmpi.so.40

I'm on CentOS and K80 GPU, cuda 9.2 and cudnn v7.1

-------------------------------------------------------------------------

2018-06-20T09:50:33Z mashu

### Suggestion

As far as I know you can have sub-packages xxx,yyy,zzz etc.. and install them as follow

```

pip install mainpackage[xxx]

```

This way different co-existing back-ends can be provided. Tensorflow can be build with different options, so at least a couple of cuda-toolkit builds could be provided this way.

The whole point of package is to save time of building, but package build for very specific set of libraries that installs fine, but does not work is counter-productive. It would be better off not to have such package in the first place.

-------------------------------------------------------------------------

2018-06-20T11:19:20Z kirk86

@mebble just make a conda virtual environment for python 3.6. Make sure that you also have installed on your system openmpi.

-------------------------------------------------------------------------

2018-06-20T11:21:13Z kirk86

@pavan-08 Install openmpi on your system also nccl 2.x whatever is the latest from nvidia. I've compiled tf with most of the packages and libraries, so it can be used hdfs, kafta, aws, etc. That's why is asking libmpi.so because it's from openmpi library.

-------------------------------------------------------------------------

2018-06-28T15:21:48Z justforkix25

With cuda 9.2 and tensorflow-gpu 1.8 I cannot build tensorflow

declared output 'external/local\_config\_cuda/cuda/cuda/lib/libcudnn.so.7' is a dangling symbolic link

The symlink exists

Versions:

Ubuntu 17.10

cuda 9.2

cudnn 7.1.4

tensorflow-gpu 1.8.0

-------------------------------------------------------------------------

2018-06-29T10:29:38Z jcRisch

Cuda is only compatible with Nvidia graphics card. If you still have the error after downgrading to tf-gpu 1.4; maybe you don't have a Nvidia !

You can check your graphic card with the command : `lspci|grep VGA`

-------------------------------------------------------------------------

2018-07-04T01:56:00Z Jackiexiao

This works for me (tensorflow-gpu==1.8.0 and cuda version is 9.0, install in anaconda)

```

export LD\_LIBRARY\_PATH=LD\_LIBRARY\_PATH:/usr/local/cuda-9.0/lib64/

```

suggestion from: https://stackoverflow.com/questions/48428415/importerror-libcublas-so-9-0-cannot-open-shared-object-file

-------------------------------------------------------------------------

2018-07-16T03:05:59Z johntiger1

@Jackiexiao Yea, that's what I was going to say. I'm using a workstation, so for me all I needed to do was use CUDA 9.0 instead of 9.2 (since multiple versions of CUDA were installed)

-------------------------------------------------------------------------

```

export LD\_LIBRARY\_PATH=LD\_LIBRARY\_PATH:/usr/local/cuda-9.0/lib64/

```

suggestion from: https://stackoverflow.com/questions/48428415/importerror-libcublas-so-9-0-cannot-open-shared-object-file

-------------------------------------------------------------------------

2018-07-16T03:05:59Z johntiger1

@Jackiexiao Yea, that's what I was going to say. I'm using a workstation, so for me all I needed to do was use CUDA 9.0 instead of 9.2 (since multiple versions of CUDA were installed)

-------------------------------------------------------------------------