Code Issues 1.2k Pull requests 169 Actions Projects 4 Wiki Security Insights

New issue

Jump to bottom

SyntaxNet fails to build with GPU support #248

Closed **nryant** opened this issue on Jun 29, 2016 · 24 comments

Assignees

nryant commented on Jun 29, 2016 • edited

I've been trying for over a day to get SyntaxNet to build with GPU support, and while every attempt passes all tests, invariably the version of TensorFlow that it compiles lacks GPU support:

```
ldd models/syntaxnet/bazel-
bin/syntaxnet/parser_trainer.runfiles/external/org_tensorflow/ensorflow/python/_pywrap_tensorflow

    linux-vdso.so.1 =>  (0x00007ffc2cbd6000)
    libdl.so.2 => /lib/x86_64-linux-gnu/libdl.so.2 (0x00007f1ba0e88000)
    libm.so.6 => /lib/x86_64-linux-gnu/libm.so.6 (0x00007f1ba0b82000)
    libpthread.so.0 => /lib/x86_64-linux-gnu/libpthread.so.0 (0x00007f1ba0964000)
    libstdc++.so.6 => /usr/lib/x86_64-linux-gnu/libstdc++.so.6 (0x00007f1ba05e8000)
    libgcc_s.so.1 => /lib/x86_64-linux-gnu/libgcc_s.so.1 (0x00007f1ba03d1000)
    libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f1ba000c000)
    /lib64/ld-linux-x86-64.so.2 (0x00007f1ba2f7a000)
```

I've done this with both the current version of SyntaxNet (`a4b7bb9`) and also the original release (`32ab5a5`) with the following system setup:

- Ubuntu 14.0.4 LTS
- TITAN X
- CUDA 7.5
- cuDNN v4
- g++ 4.8.4
- bazel 0.2.2b
- Python 2.7.10

NOTE that I've never had trouble compiling TensorFlow separately. Has anyone experienced similar issues recently?

andydavis1 assigned **calberti** on Jul 1, 2016

aselle removed the **triaged** label on Jul 29, 2016

flashxing commented on Aug 1, 2016

I have the same problem with you. Have you solved this?

😄 1

todtom commented on Aug 21, 2016

@nryant Hi, I have the same problem, could you tell me how to solve this?

David-Ba commented on Aug 22, 2016 • edited

Hi, I had the same problem and managed to build SyntaxNet with GPU support with the following steps:

1. Make sure you have the following environment variables set: `CUDA_HOME="[path_to_cuda_top_directory]" LD_LIBRARY_PATH="[path_to_cuda_lib64_directory]:$LD_LIBRARY_PATH" PATH="[path_to_cuda_bin_directory]:$PATH"`
2. Add the line `build --config=cuda` to `tools/bazel.rc`
3. Add the line `cxx_builtin_include_directory: "/usr/local/cuda-7.5/targets/x86_64-linux/include”` to `tensorflow/third_party/gpus/crosstool/CROSSTOOL` (with the cuda part pointing to your Cuda installation)
4. Force Tensorflow to use Cuda by changing the `//conditions:default` part in `syntaxnet/syntaxnet.bzl` from `if_false` to `if_true`
5. Do the same thing for `tensorflow/third_party/gpus/cuda/build_defs.bzl`
6. Build SyntaxNet using this command: `bazel test -c opt --config=cuda --define using_cuda_nvcc=true --define using_gcudacc=true syntaxnet/... util/utf8/...`

Two tests will fail because SyntaxNet cannot find the Cuda dependencies for some reason (cf. test logs). It seems that the LD_LIBRARY_PATH variable is not set in the test environment. When running the *parser_eval* and *parser_trainer* script, however, it should be no problem. Running SyntaxNet on the example in this stage might cause a CUDA_OUT_OF_MEMORY error. A fix for this is available here: #173

Side note: I used Ubuntu 14.04, Cuda 7.5, and cuDNN 4.0.7

@David-Ba I'm not sure why the bazel.rc set crosstool_top to `//third_party/gpus/crosstool` , maybe the first line of tools/bazel.rc need to be modified like `//tensorflow/third_party/gpus/crosstool` and followed your 6 steps and some additional error occured.

command is `~/tools/tensorflow/models/syntaxnet$ bazel test -c opt --config=cuda --define using_cuda_nvcc=true --define using_gcudacc=true syntaxnet/... util/utf8/...`
and show these messages

```
INFO: Found 68 targets and 17 test targets...
INFO: From Compiling
external/org_tensorflow/tensorflow/core/kernels/spacetodepth_op_gpu.cu.cc:
nvcc warning : option '--relaxed-constexpr' has been deprecated and replaced by option '--
expt-relaxed-constexpr'.
nvcc warning : option '--relaxed-constexpr' has been deprecated and replaced by option '--
expt-relaxed-constexpr'.
/usr/include/string.h: In function 'void* __mempcpy_inline(void*, const void*, size_t)':
/usr/include/string.h:652:42: error: 'memcpy' was not declared in this scope
   return (char *) memcpy (__dest, __src, __n) + __n;
                                          ^
ERROR:
/home/hjm/.cache/bazel/_bazel_hjm/1e0c52c2d9671225fb0df00406e3d29b/external/org_tensorflow/tensor
 output
'external/org_tensorflow/tensorflow/core/kernels/_objs/depth_space_ops_gpu/external/org_tensorflo
 was not created.
ERROR:
/home/hjm/.cache/bazel/_bazel_hjm/1e0c52c2d9671225fb0df00406e3d29b/external/org_tensorflow/tensor
 not all outputs were created.
INFO: Elapsed time: 33.099s, Critical Path: 32.81s
//syntaxnet:arc_standard_transitions_test                             NO STATUS
//syntaxnet:beam_reader_ops_test                                      NO STATUS
//syntaxnet:binary_segment_state_test                                 NO STATUS
//syntaxnet:char_properties_test                                      NO STATUS
//syntaxnet:graph_builder_test                                        NO STATUS
//syntaxnet:lexicon_builder_test                                      NO STATUS
//syntaxnet:morphology_label_set_test                                 NO STATUS
//syntaxnet:parser_features_test                                      NO STATUS
//syntaxnet:parser_trainer_test                                       NO STATUS
//syntaxnet:reader_ops_test                                           NO STATUS
//syntaxnet:segmenter_utils_test                                      NO STATUS
//syntaxnet:sentence_features_test                                    NO STATUS
//syntaxnet:shared_store_test                                         NO STATUS
//syntaxnet:tagger_transitions_test                                   NO STATUS
//syntaxnet:text_formats_test                                         NO STATUS
//util/utf8:unicodetext_unittest                                      NO STATUS

 Executed 0 out of 17 tests: 1 fails to build and 16 were skipped.
```

I'm new to tensorflow, I only want to get the parsed tree faster with using gpus . I'm sincerly sorry if there are some silly questions.

I used Ubuntu 16.04, Cuda 7.5, and cuDNN 4.0.7, Geforce GTX TITANX

David-Ba commented on Aug 22, 2016

@todtom Yes, I set `crosstool_top` in `tools/bazel.rc` to `cuda --crosstool_top=@org_tensorflow//third_party/gpus/crosstool` . I forgot to mention that. Also, I am not sure whether this is the way to go. I just looked around the config files and changed them to what I thought is right. However, I have not encountered your error so far. Maybe do a `bazel clean` and then rebuild. It helps sometimes.

todtom commented on Aug 22, 2016

`bazel clean` seems not working for me. Can anyone help me?

calberti commented on Aug 26, 2016

Thanks **@David-Ba** for your detailed answer!
@todtom: the issue running `bazel clean` seems unrelated to GPU support. Can you open a new issue or ask on stack overflow to get more help if needed?

calberti closed this as completed on Aug 26, 2016

Shnurre mentioned this issue on Sep 9, 2016

SyntaxNet fails with CUDA out of memory #173

Closed

Shnurre mentioned this issue on Sep 28, 2016

SintaxNet failes to allocate memory while using GPU #469

Closed

chrhad commented on Oct 22, 2016

I have followed the 6 steps provided by **@David-Ba** as follows:

2. Add the line build --config=cuda to tools/bazel.rc
3. Add the line cxx_builtin_include_directory: "/usr/local/cuda-7.5/targets/x86_64-linux/include” to tensorflow/third_party/gpus/crosstool/CROSSTOOL (with the cuda part pointing to your Cuda installation)
4. Force Tensorflow to use Cuda by changing the //conditions:default part in syntaxnet/syntaxnet.bzl from if_false to if_true
5. Do the same thing for tensorflow/third_party/gpus/cuda/build_defs.bzl
6. Build SyntaxNet using this command: bazel test -c opt --config=cuda --define using_cuda_nvcc=true --define using_gcudacc=true syntaxnet/... util/utf8/...

and set `crosstool_top` in tools/bazel.rc to `build:cuda --crosstool_top=@org_tensorflow//third_party/gpus/crosstool`

Yet, the installation returns error as follows:
ERROR: no such target '@org_tensorflow//third_party/gpus/crosstool:crosstool': target 'crosstool' not declared in package 'third_party/gpus/crosstool' defined by /home/christian/.cache/bazel/_bazel_christian/d9875fd54a23cac839e874ac491a28bb/external/org_tensorflow/third_party/gpus/crosstool/BUILD.

Reverting the `crostool_top` back to `build:cuda --crosstool_top=//third_party/gpus/crosstool` returns the following error:
ERROR: no such package 'third_party/gpus/crosstool': BUILD file not found on package path.

Have I missed anything? My CUDA version is 7.0, with CUDNN version 4.0.7.

hfxunlp commented on Nov 15, 2016

ERROR:no such package 'third_party/gpus/crosstool': BUILD file not found on package path.

a2tm7a commented on Dec 3, 2016 • edited

Same error as **@chrhad** and **@anoidgit**. Can someone help with it

wq343580510 commented on Dec 9, 2016

ERROR:no such package 'third_party/gpus/crosstool': BUILD file not found on package path.
@David-Ba follow the instruction，It seems that many people have encountered this problem。

Closed

TheodoreGalanos commented on Feb 20, 2017

Same here. Tried copy/pasting it in the folder from syntaxnet/tensorflow/third_party but the BUILD file wasn't associated with smth like that (at least my beginner level view of that error).

Is there any updates? It seems like a small issue to my untrained eyes.

ducdauge commented on Feb 23, 2017

Hi guys. I had a similar problem but I might have found a solution here. Quoting it:

> running the same command from the tensorflow serving repository root will fail (with errors) for 2 reasons:
>
> 1. the crosstool in tools/bazel.rc is invalid (AFAIK). change @org_tensorflow//third_party/gpus/crosstool to @local_config_cuda//crosstool:toolchain.
>
> 2. the cuda_configure repository rule will fail (haven't looked in to why exactly), but essentially an bazel clean --expunge && export TF_NEED_CUDA=1 will fix this.
>
> Then, run bazel query 'kind(rule, @local_config_cuda//...)' again and all is well (for me at least); the cuda tool chain should be created in $(bazel info output_base)/external/local_config_cuda/cuda

Afterwards, `bazel test -c opt --config=cuda --define using_cuda_nvcc=true --define using_gcudacc=true syntaxnet/... util/utf8/...` failed just 1 test, but I had some memory issues with the GPU. I solved them adding `config.gpu_options.allow_growth = True` to the relevant files

☺ 👍 2

Vimos commented on Mar 19, 2017 • edited ▾

Using the method offered by **@ducdauge** , I was able to build.

But still met 2 problems.

Problem 1: nccl not found

```
In file included from
external/org_tensorflow/tensorflow/contrib/nccl/kernels/nccl_manager.cc:15:0:
```

Problem 1 is solved via commenting out nccl refered in tensorflow/serving#327

Problem 2: Tests failure

```
At global scope:
cc1plus: warning: unrecognized command line option '-Wno-self-assign'
FAIL: //syntaxnet:reader_ops_test (see
/data/home/vimos/.cache/bazel/_bazel_vimos/8c5df8ecbe273164beccb9b372c94778/execroot/syntaxnet/ba
out/local_linux-opt/testlogs/syntaxnet/reader_ops_test/test.log).
FAIL: //syntaxnet:graph_builder_test (see
/data/home/vimos/.cache/bazel/_bazel_vimos/8c5df8ecbe273164beccb9b372c94778/execroot/syntaxnet/ba
out/local_linux-opt/testlogs/syntaxnet/graph_builder_test/test.log).
```

I am working on these failures right now, it seems that they are memory related issues.

utkrist commented on Mar 24, 2017 • edited

Following is the summary of what worked for me. It is based on prev comments and other sources.

0. Install all the dependencies for syntaxnet

1. Choose non NFS location for bazel temp directory related files. I choose '/tmp/bazeltemp'. Add this line to .bashrc:
   `export TEST_TMPDIR=/tmp/bazeltemp`

2. Install bazel using installer (I choose 0.4.5). Here, bin and bazelrc can be in NFS location
   `$ chmod +x bazel-version-installer-os.sh`
   `$ ./bazel-version-installer-os.sh --bin=$HOME/bin --base=/tmp/bazeltemp/base --bazelrc=$HOME/.bazelrc`

3. Make following edit in configure file of tensorflow: models/syntaxnet/tensorflow/configure
   Replace `bazel clean --expunge` with `bazel clean --expunge_async`

4. `$ ./configure`
   Experiment with different options if you like to

```
Please specify the location of python. [Default is /home/anaconda2/bin/python]:
Please specify optimization flags to use during compilation when bazel option "--config=opt"
is specified [Default is -march=native]:
Do you wish to use jemalloc as the malloc implementation? [Y/n] y
jemalloc enabled
Do you wish to build TensorFlow with Google Cloud Platform support? [y/N] n
No Google Cloud Platform support will be enabled for TensorFlow
Do you wish to build TensorFlow with Hadoop File System support? [y/N] y
Hadoop File System support will be enabled for TensorFlow
Do you wish to build TensorFlow with the XLA just-in-time compiler (experimental)? [y/N] n
```

```
Please input the desired Python library path to use. Default is
[/home/anaconda2/lib/python2.7/site-packages]

Using python library path: /home/anaconda2/lib/python2.7/site-packages
Do you wish to build TensorFlow with OpenCL support? [y/N] n
No OpenCL 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
Please specify which gcc should be used by nvcc as the host compiler. [Default is
/usr/bin/gcc]:
Please specify the CUDA SDK version you want to use, e.g. 7.0. [Leave empty to use system
default]: 8.0
Please specify the Cudnn version you want to use. [Leave empty to use system default]: 5.1
Please specify the location where cuDNN 5.1 library is installed. Refer to README.md for more
details. [Default is /opt/software/cuda/cuda-8.0]:
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.
[Default is: "3.5,5.2"]
```

5. Follow the instructions below:
   a. Make sure you have the following environment variables set in .bashrc

   `CUDA_HOME="[path_to_cuda_top_directory]"`

   `LD_LIBRARY_PATH="[path_to_cuda_lib64_directory] :$LD_LIBRARY_PATH"`

   `PATH="[path_to_cuda_bin_directory]:$PATH"`

   For example my .basrhc has following

```
     export ORACLE_HOME=/opt/software/oracle/product/12.1.0/client
     export PATH=${PATH}:${ORACLE_HOME}/bin
     export PATH=/home/IAIS/uadhikari/anaconda2/bin:$PATH

     export CUDA_HOME=/opt/software/cuda/cuda-8.0
     export CUDA_TOOLKIT_PATH=${CUDA_HOME}
     export
LD_LIBRARY_PATH=${CUDA_HOME}/lib64:${CUDA_HOME}/extras/CUPTI/lib64:$LD_LIBRARY_PATH
     export PATH=${CUDA_HOME}/bin:${PATH}

     export JAVA_HOME=/opt/software/jdk1.8.0_51
     export PATH=/tmp/bazeltemp/bin:$PATH
     export CUDNN_HOME=${CUDA_HOME}
     export TEST_TMPDIR=/tmp/bazeltemp




b. Add the line `build --config=cuda to tools/bazel.rc` (I added as first line in the file)

c. In the file tensorflow/third_party/gpus/crosstool/CROSSTOOL,
replace every `cxx_builtin_include_directory: "%{cuda_include_path}"`
with `cxx_builtin_include_directory: "your/cuda/home/path/include"`
```

```
e. Do the same thing for tensorflow/third_party/gpus/cuda/build_defs.bzl
```

5. `$ bazel clean --expunge_async`

6. Carefully run each of these
   `$ export TF_NEED_CUDA=1`
   `$ export CUDA_TOOLKIT_PATH=$CUDA_HOME`
   `$ export TF_CUDA_VERSION=8.0`
   `$ export TF_CUDNN_VERSION=5.1`
   `$ export CUDNN_INSTALL_PATH=$CUDA_HOME`

7. This has to be run in in models/syntaxnet folder
   `$ bazel test -c opt --config=cuda --define using_cuda_nvcc=true --define using_gcudacc=true syntaxnet/... util/utf8/...`
   If you get error about cross tool or local_config_cuda, goto step 5 and try again

8. If you get error about nccl:
   comment out the dependency for nccl in: tensorflow/tensorflow/contrib/BUILD as mentioned in
   ✓ **bazel GPU build error with fatal error: external/nccl_archive/src/nccl.h: No such file or directory** serving#327
   Goto step 5 and try again

I hope this works for you.

☺ 👍 3

jhowliu commented on Apr 19, 2017

Hi **@utkrist**,

I followed your instructions, but I had some test failed.

The log says message below.

```
exec ${PAGER:-/usr/bin/less} "$0" || exit 1
-----------------------------------------------------------------------------
2017-04-18 10:54:49.948817: F
external/org_tensorflow/tensorflow/core/framework/allocator_registry.cc:42] Check failed:
!CheckForDuplicates(name, priority) Allocator with name: [DefaultCPUAllocator] and priority:
[100] already registered
external/bazel_tools/tools/test/test-setup.sh: line 159:  2453 Aborted
```

Have you ever seen it ?

utkrist commented on Apr 19, 2017

@jhowliu I suggest that you downgrade your bazel to 0.4.2 and try the instructions again. If it still does not work then let me know. I will then make a fresh install in new machine and will try to reproduce your error.

jhowliu commented on Apr 19, 2017 • edited ▾

Hi **@utkrist** ,
Thanks for your suggestions.
It worked when I used bazel 0.4.5 and tensorflow at a7d6015. (It still worked with bazel 0.4.2)

But I got the another trouble about ran of the memory when I use the demo example.
Please take a look the log.
I have tried your instruction given in #173 but still not work.

Thanks again.

utkrist commented on Apr 24, 2017

Hi **@jhowliu**,
Did you mange to solve the problem?

jhowliu commented on Apr 25, 2017

@utkrist
I tried the many version of bazel and lots of instructions i could find but still out of memory.
Maybe my gpu is not enough memory to use the syntaxnet.

so i will use the syntaxnet with cpu until solve the problem.
do you have any ideas ?

sgrvinod mentioned this issue on May 4, 2017

"Segmentation fault (core dumped)" while trying to run Syntaxnet on GPU #1440

Closed

these instructions do no work on MacOS.

udnaan mentioned this issue on May 7, 2017

Syntaxnet fails to compile with XLA enabled on MacOS #1444

Closed

bogatyy mentioned this issue on Jun 2, 2017

input contains unknown fields and/or extensions #1521

Closed

smirnovevgeny commented on Jun 21, 2017 • edited

I've spent a day for reading syntaxnet gpu issues to build docker container.
This container passes sentences through parsey_universal with Russian-SynTagRus model.
`docker pull evgenysmirnov/syntaxnet:cuda`
https://hub.docker.com/r/evgenysmirnov/syntaxnet/

☺ 👍 3 ❤️ 2

zerodarkzone commented on Nov 30, 2017 • edited

I have some tests fails with an OUT_OF_MEMORY error.
I built it with Bazel 0.5.4, Cuda 8.0 and Cudnn 6.0.23
Any update on this error?

lifeiteng commented on Apr 26, 2018

@zerodarkzone try `bazel test --jobs 1 ...`

Assignees

calberti

Labels

None yet

Milestone

No milestone

Development

No branches or pull requests

20 participants