tensorflow / **tensorflow**

GPU: no known devices, despite cuda's deviceQuery returning a "PASS" result #7648

New issue

Closed **oelmekki** opened this issue on 18 Feb · 10 comments

oelmekki commented on 18 Feb • edited

Hi guys,

Sorry, I know there's already been tons of GPU related issues, but I could not find any that seems to be related to my problem.

The main symptom: when running tensorflow, my gpu is not detected (the code being run, and its output).

What differs from usual issues is that cuda seems properly installed and running `./deviceQuery` from cuda samples is successful (output).

I have two graphical cards:

- an old GTX 650 used for my monitors (I don't want to use that one with tensorflow)
- a GTX 1060 that I want to dedicate to tensorflow

I use:

- tensorflow-1.0.0
- cuda-8.0 (ls -l /usr/local/cuda/lib64/libcud*)
- cudnn-5.1.10
- python-2.7.12
- nvidia-drivers-375.26 (this was installed by cuda and replaced my distro driver package)

I've tried:

- adding `/usr/local/cuda/bin/` to `$PATH`
- forcing gpu placement in tensorflow script using `with tf.device('/gpu:1'):` (and `with tf.device('/gpu:0'):` when it failed, for good measure)
- whitelisting the gpu I wanted to use with `CUDA_VISIBLE_DEVICES`, in case the presence of my old unsupported card did cause problems
- running the script with sudo (because why not)

Here are the outputs of nvidia-smi and nvidia-debugdump -l, in case it's useful.

At this point, I feel like I have followed all the breadcrumbs and have no idea what I could try else. I'm not even sure if I'm contemplating a bug or a configuration problem. Any advice about how to debug this would be greatly appreciated. Thanks!

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

4 participants

yaroslavvb commented on 19 Feb • edited Contributor

As a shot in the dark, maybe try TF 1.0 and look at output of verbose logging

```
export TF_CPP_MIN_VLOG_LEVEL=1
```

There's also `export TF_CPP_MIN_VLOG_LEVEL=2` but that can be 100's of MBs of output

vrv commented on 19 Feb Contributor

This is a question that should be posted on StackOverflow -- can you post it there and link us?

vrv closed this on 19 Feb

oelmekki commented on 19 Feb

Sure, I asked it here : http://stackoverflow.com/questions/42326748/tensorflow-on-gpu-no-known-devices-despite-cudas-devicequery-returning-a-pas

oelmekki commented on 19 Feb • edited

@yaroslavvb Thanks for the tip!

Actually, using this does not provide any more info than not using it:

```
$ TF_CPP_MIN_LOG_LEVEL=1 python gpu.py
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't
compiled to use SSE3 instructions, but these are available on your machine and could
speed up CPU computations.
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't
compiled to use SSE4.1 instructions, but these are available on your machine and could
speed up CPU computations.
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't
compiled to use SSE4.2 instructions, but these are available on your machine and could
speed up CPU computations.
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't
compiled to use AVX instructions, but these are available on your machine and could speed
up CPU computations.
Device mapping: no known devices.
MatMul: (MatMul): /job:localhost/replica:0/task:0/cpu:0
b: (Const): /job:localhost/replica:0/task:0/cpu:0
a: (Const): /job:localhost/replica:0/task:0/cpu:0
[[ 22.  28.]
 [ 49.  64.]]
```

(I've also tried to export the variable globally just in case, but it produces the same result)

Funny enough, setting log_level to 2 actually produces a lesser amount of output:

```
$ TF_CPP_MIN_LOG_LEVEL=2 python gpu.py
Device mapping: no known devices.
MatMul: (MatMul): /job:localhost/replica:0/task:0/cpu:0
b: (Const): /job:localhost/replica:0/task:0/cpu:0
a: (Const): /job:localhost/replica:0/task:0/cpu:0
[[ 22.  28.]
 [ 49.  64.]]
```

EDIT: oh, got it. Log level works as the lower the number, the more verbose (and 1 seems to be the default):

```
$ TF_CPP_MIN_LOG_LEVEL=0 python gpu.py
```

```
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't
compiled to use SSE3 instructions, but these are available on your machine and could
speed up CPU computations.
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't
compiled to use SSE4.1 instructions, but these are available on your machine and could
speed up CPU computations.
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't
compiled to use SSE4.2 instructions, but these are available on your machine and could
speed up CPU computations.
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't
compiled to use AVX instructions, but these are available on your machine and could speed
up CPU computations.
Device mapping: no known devices.
I tensorflow/core/common_runtime/direct_session.cc:257] Device mapping:

MatMul: (MatMul): /job:localhost/replica:0/task:0/cpu:0
I tensorflow/core/common_runtime/simple_placer.cc:841] MatMul: (MatMul)/job:localhost
/replica:0/task:0/cpu:0
b: (Const): /job:localhost/replica:0/task:0/cpu:0
I tensorflow/core/common_runtime/simple_placer.cc:841] b: (Const)/job:localhost/replica:0
/task:0/cpu:0
a: (Const): /job:localhost/replica:0/task:0/cpu:0
I tensorflow/core/common_runtime/simple_placer.cc:841] a: (Const)/job:localhost/replica:0
/task:0/cpu:0
[[ 22.  28.]
 [ 49.  64.]]
```

yaroslavvb commented on 19 Feb — Contributor

@oelmekki debug logging was added recently, can you try this with TF 1.0?

oelmekki commented on 19 Feb

This is already what I'm using, actually :)

```
$ python -c "import tensorflow; print(tensorflow.__version__)"
1.0.0
```

yaroslavvb commented on 19 Feb — Contributor

Sorry, it's TF_CPP_MIN_VLOG_LEVEL not TF_CPP_MIN_LOG_LEVEL (there's also TF_CPP_MIN_LOG_LEVEL which will disable TF_CPP_MIN_VLOG_LEVEL if set)

oelmekki commented on 19 Feb

Great, thanks, I have a lot more of debugging info indeed 👍

Since Vijay mentioned this should be discussed on SO instead, I won't paste it here, but let me know if you want to look at it at any point.

Thanks for help!

collawolley commented 22 days ago

hi **@oelmekki** **@yaroslavvb** did you managed to resolve the issue?
I have the same problem able to use GPU before updating tensorflow to V1.3.0. I have also upgraded my Cudnn to V6. My CUDA is v8.0 so I don't seem to understand where the problem is coming from. I can verify that my tensorflow is GPU version because I used tfBInaryUrl for Python2.7-PGU support. Aside this, I have also installed several times with 'pip install tensorflow-gpu' and I still cannot run my codes on gpu. Theano works fine and I could run code with GPU if I use theano.

When I tried to force the computation to be run on GPU , my codes wouldn't run and now, I got this mesage "Device mapping: no known devices.

The frustrating thing is that I was using the GPU before I upgraded to v1.3.0/

I would appreciate any help as regards this problem.
"

oelmekki commented 21 days ago

Hi **@collawolley**,

Indeed, it was solved for me : the problem was that there's a different lib to install for using gpu, `tensorflow-gpu` instead of `tensorflow` (more info here : https://stackoverflow.com/questions/42326748/tensorflow-on-gpu-no-known-devices-despite-cudas-devicequery-returning-a-pas).

That being said, it's been a while since I haven't used tensorflow (I used it to generate word embedding, but now there are a lot already available), so I can't say for sure if this answer is still relevant.

Best wishes!