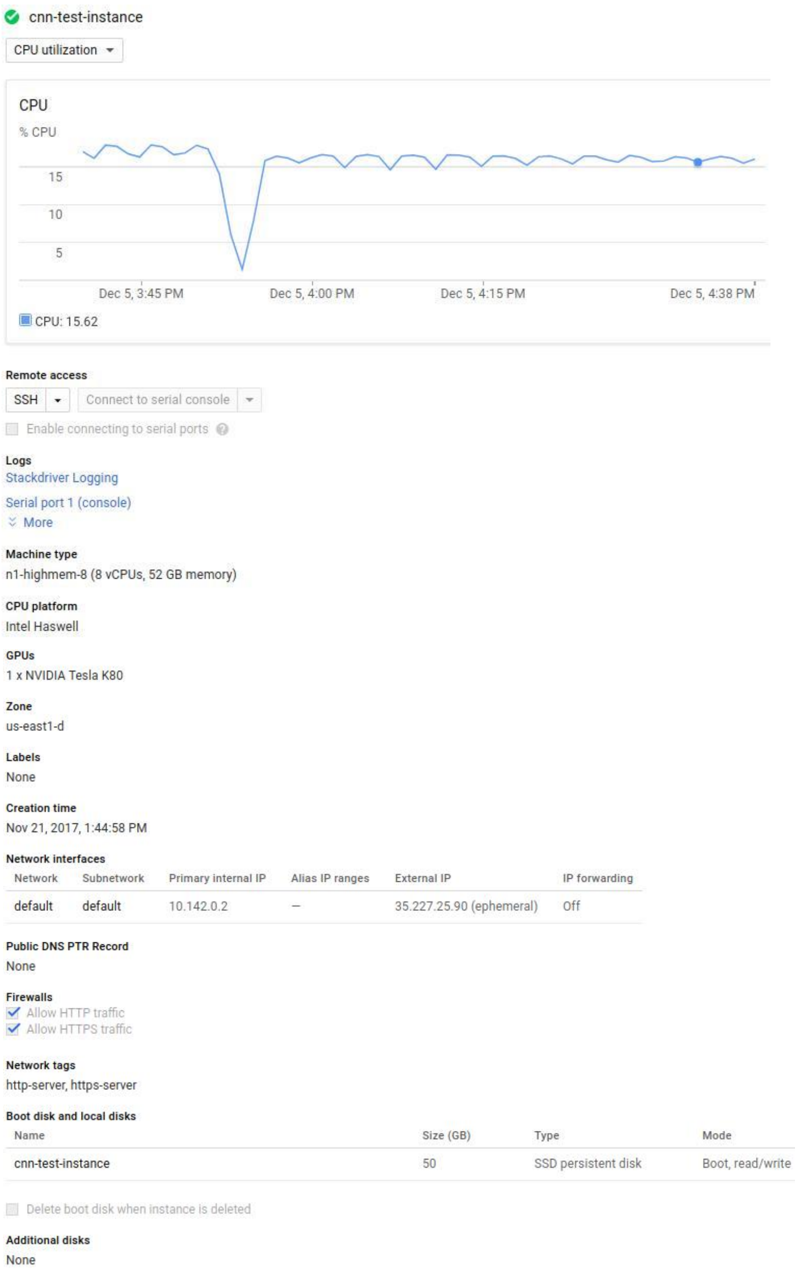
There is a memory error while submitting jobs to Google Cloud. This happens when the training dataset increases to 8000 .jpeg images, so a virtual machine is needed to solve the problem. A virtual instance machine using Google Cloud's Compute engine is created with the Ubuntu 16.04.3 LTS (Xenial Xerus) as the operating system with a Tesla K80 GPU. In order to speed up compilation of the neural network, Tensorflow must be installed from source. With the virtual machine running in the cloud, the splitting of the training dataset into a half is not needed. This allows more accurate training since the learning of the dataset will run on an instance of single image generator instead of two. To connect to the virtual machine, gcloud command line tool needs to be installed and run with the command: gcloud compute ssh --zone=us-east1-d cnn-test-instance



cnn-test-instance

CPU: 15.62

Remote access

□ Enable connecting to serial ports

Logs

Machine type
n1-highmem-8 (8 vCPUs, 52 GB memory)

CPU platform
Intel Haswell

GPUs
1 x NVIDIA Tesla K80

Zone
us-east1-d

Labels
None

Creation time
Nov 21, 2017, 1:44:58 PM

Network interfaces

Network	Subnetwork	Primary internal IP	Alias IP ranges	External IP	IP forwarding
default	default	10.142.0.2	–	35.227.25.90 (ephemeral)	Off

Public DNS PTR Record
None

Firewalls
☑ Allow HTTP traffic
☑ Allow HTTPS traffic

Network tags
http-server, https-server

Boot disk and local disks

Name	Size (GB)	Type	Mode
cnn-test-instance	50	SSD persistent disk	Boot, read/write

□ Delete boot disk when instance is deleted

Additional disks
None

The sources that are used to install Tensorflow from source are below:

- https://hackernoon.com/launch-a-gpu-backed-google-compute-engine-instance-and-setup-tensorflow-keras-and-jupyter-902369ed5272
- https://medium.com/@acrosson/installing-nvidia-cuda-cudnn-tensorflow-and-keras-69bbf33dce8a
- https://www.tensorflow.org/install/#installing-from-sources

These commands are chronologically executed to install Tensorflow from source with CUDA GPU support:

1. sudo apt-get update
2. sudo apt-get upgrade

3. sudo apt-get install -y build-essential
4. lspci | grep -i nvidia
5. sudo apt-get install linux-headers-$(uname -r)
6. mkdir tmp/
7. cd tmp
8. curl -O https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86_64/cuda-repo-ubuntu1604_8.0.61-1_amd64.deb
9. ls
10. md5sum cuda-repo-ubuntu1604_8.0.61-1_amd64.deb
11. sudo dpkg -i cuda-repo-ubuntu1604_8.0.61-1_amd64.deb
12. sudo apt-get update
13. sudo apt-get install -y cuda-8-0
14. nvidia-smi
15. cat ~/.bashrc
16. source ~/.bashrc
17. ls
18. cd ~/NVIDIA_CUDA-8.0_Samples/1_Utilities/deviceQuery
19. make
20. ./deviceQuery

To install onedrive to sync data between virtual and local machine:

1. git clone https://github.com/xybu92/onedrive-d.git

Installing CUDA and CUDNN:

1. sudo EDITOR=vim visudo
2. ./install.sh
3. onedrive-pref
4. sudo chmod u+x install.sh
5. cat README.md
6. sudo touch /var/log/onedrive_d.log
7. ./install.sh
8. onedrive-pref
9. onedrive-d start
10. cd OneDrive/
11. gunzip cudnn-8.0-linux-x64-v5.1.tgz
12. tar xvf cudnn-8.0-linux-x64-v5.1.tar

13. cp cudnn-8.0-linux-x64-v5.1.tar ~/tmp/
14. cd ~/tmp/
15. tar xvf cudnn-8.0-linux-x64-v5.1.tar
16. sudo cp -P cuda/include/cudnn.h $CUDA_HOME/include
17. sudo cp -P cuda/lib64/libcudnn* $CUDA_HOME/lib64
18. sudo chmod u+w $CUDA_HOME/include/cudnn.h
19. sudo chmod a+r $CUDA_HOME/lib64/libcudnn*
20. nvidia-smi
21. Finally, you must also install libcupti-dev by invoking the following command: sudo apt-get install libcupti-dev

Installing python 2.7 as a virtual environment on Ubuntu:

1. mkdir py27
2. sudo apt-get install python-pip python-dev python-virtualenv
3. virtualenv --system-site-packages py27
4. cd py27/
5. source bin/activate
6. easy_install -U pip
7. sudo apt-get install python-numpy python-dev python-pip python-wheel
8. sudo apt-get install openjdk-8-jdk
9. echo "deb [arch=amd64] http://storage.googleapis.com/bazel-apt stable jdk1.8" | sudo tee /etc/apt/sources.list.d/bazel.list
10. curl https://bazel.build/bazel-release.pub.gpg | sudo apt-key add -
11. sudo apt-get update && sudo apt-get install bazel
12. sudo apt-get upgrade bazel
13. cd ~
14. git clone https://github.com/tensorflow/tensorflow
15. cd tensorflow/
16. ./configure
17. type in 5 for cudnn version, /usr/local/cuda-8.0 for cuda path
18. bazel build --config=opt --config=cuda //tensorflow/tools/pip_package:build_pip_package
19. sudo pip install /tmp/tensorflow_pkg/tensorflow-1.4.0-py2-none-any.whl
20. bazel-bin/tensorflow/tools/pip_package/build_pip_package /tmp/tensorflow_pkg
21. sudo pip install /tmp/tensorflow_pkg/tensorflow-1.4.0-cp27-cp27mu-linux_x86_64.whl

Installing libraries to run neural network program:

1. cd py27/
2. source bin/activate
3. python
4. pip install keras==2.0.6
5. pip install joblib
6. pip install opencv-python
7. pip install Pillow
8. pip install h5py