How to use the AASS Deep Learning Gear

Todor Stoyanov

July 1, 2019

1 Purpose

The purpose of this document is to give a step-by-step guide to setting up an account on the AASS learning gear desktops and using it with docker.

2 Local computer set-up

- generate an ssh key, if you don't have one yet: `ssh-keygen` This generates the files `/home/.ssh/id_rsa id_rsa.pub` Email the public key to `todor.stoyanov@oru.se`
- setup your hosts file `/etc/hosts` The addresses and suggested computer aliases are:

```
130.243.124.161 tsv-desktop
130.243.124.173 jsk-desktop
130.243.124.188 sly-desktop
130.243.124.159 aam-desktop
```

- Now you should be able to ssh into the desktop using `ssh username@xxx-desktop`
- Download and install the ThinLinc client from `https://www.cendio.com/thinlinc/download`
- Start a ThinLinc client. Enter the computer name as server, your username as username. Click on Options, go to the Security tab and under Authentication Method select "Public key"
- Enter the address of your private key in the dialogue and click Connect.
- If everything works as it should, you will get a session on the remote computer.
- Note: to get out of this session you should either log out, or press F8 and minimize / disconnect the session. **Do not leave sessions running for a long time unless you are doing some computation.**

3 Setting up Docker and running on remote

Each desktop has the Nvidia graphics card driver installed, along with docker and nvidia-docker. The primary idea here is that you should run your code within a docker container, in order to not create a huge mess of dependencies. Make sure you do a docker tutorial in case you are feeling shaky.

To get you started with docker, you can clone this repository: `https://github.com/tstoyanov/tf_openailab_gpu_docker.git` This shows an example of how to build your own docker container with tensorflow on the gpu, a jupyter notebook server, and open AI gym for reinforcement learning. Adapt this as you see fit for your own purposes. For convenience, build the container once and push it to your docker hub account (you can create one for free). You can then pull and execute the container on any of the learning desktops. Note on data management: try to save all checkpoints and load all data from a common data directory. The learning desktops all have a 3.6T hard drive mounted under /media/local-data and a network mount of the Synology DiskStation under /media/DiskStation. If you are running models on several machines, the network mount is a good option.

When you are comfortable with docker, take a look at the following link: `https://docs.google.com/spreadsheets/d/1YYviK8Qsg1OUcYkbPz4hmSZmZap6JrdCUmgYcGUNGcQ/edit?usp=sharing`. This is an experimental signup system, to avoid several users trying to take control of the GPU at the same time. Find a desktop that is currently free and sign up to use it, either partially (for prototyping and development), or exclusively (for training and experiments). You should not sign up for more than 3 consecutive days of prototyping or more than 7 consecutive days of exclusive use. Please do not occupy more than one desktop at a time if there is a high load (two or more other exclusive users).

u