Anaconda3-2020.02+ (2020/5/13)

Usage

There so many detailed guides of conda/anaconda elsewhere on the internet. Please also check them.

Initializtion

You can activate conda settings by loading initialization script.

bash, zsh:

```
% . /local/apl/lx/anaconda3-2020.02-May/conda_init.sh
```

csh (tcsh):

```
% source /local/apl/lx/anaconda3-2020.02-May/conda_init.csh
```

Upon this stage, you may be in the "base" environment. The environment name will be shown at the far left on your prompt like below.

```
(base) %
```

Notes

This initialization operation which will access a large number of files would often be very time consuming. This is unavoidable due to the nature of the distributed file system used in RCCS. Therefore, we recommend to read the anaconda environment only when it is necessary.

On the other hand, once loaded, you can read fast for the time being (because it is cached). Similarly, there is basically no impact upon running actual python jobs.

Environments

- base: default environment
  - python 3.7.7
  - numpy 1.18.1, scipy 1.4.1, pandas 1.0.3, pyyaml 5.3.1, pillow 7.1.2, opencv 3.4.2, scikit-image 0.16.2, scikit-learn 0.22.1 他
  - tensorflow 2.1.0, keras 2.3.1, pytorch 1.5.0
    - GPU packages are not installed. Basic anaconda env + deep learning environment (CPU version of tensorflow, keras, pytorch) (chainer is not installed)
    - Most of standard packages used in research such as pandas are available.
- gpuenv: PyTorch, Tesorflow GPU environment
  - python 3.7.7
  - tensorflow 2.1.0, keras(-gpu) 2.3.1, cudatoolkit 10.1.243, cudnn 7.6.5

The installed packages in the environments above won't be updated. In other word, environments will be kept intact. New version of packages will be installed upon creating new environments or creating another anaconda installation.

Switching Environment

You can change environment by executing "conda active" command after the initialization of conda itself.
Example:

```
(base) % conda activate gpuenv
(gpuenv) %
```

Current environment name will be shown at the leftmost of the prompt regardless of your login shell type (csh or bash or zsh).

You can exit from the conda environment by "conda deactivate" command.

```
(gpuenv) % conda deactivate
(base) % conda deactivate
%
```

Creating Your Own Environment

Users cannot add packages in this anaconda directory, since this anaconda is installed in the system domain (/local/apl/lx). Instead, you can build your own conda environment using this pre-installed anaconda. The new environment files will be placed under ~/.conda directory in the default setting.

Example: new anaconda env creation ("myenv" is your new environment name):

```
(base) % conda create -n myenv anaconda
```

Example: clone "base" environment ("myenv" is your new environment name):

```
(base) % conda create -n myenv --clone base
```

(note: conda_init.sh, conda_init.csh, samples_rccs/ will also be cloned. You can remove those files after the cloning.)

You can also directly install anaconda (or miniconda) under your home directory.
If you need a very special environment, this way may be a better solution.

Run MNIST Sample

MNIST sample scripts are available in /local/apl/lx/anaconda3-2020.02-May/samples_rccs directory. You can run them just by copying directory and submit by using jsub. The script initializes conda, chooses environment if necessary, and then run MNIST sample.

Note: if your login shell is "csh" and try to run GPU sample, you may need to modify script (see comments in the script).

Example (pytorch):

```
% cp -r /local/apl/lx/anaconda3-2020.02-May/samples_rccs/pytorch/ ~/pytorch_sample_mnist
% cd ~/pytorch_sample_mnist
% jsub -q PN sample-pytorch-cpu.sh
```

(MNIST data is downloaded on /local/apl/lx/anaconda3-2019Jul/samples_rccs/datasets in advance. Please note that computation nodes cannot access internet.)

Extra: Install packages without conda envs

You can also install packages under your home directory via "pip" or "pip3" command without conda.

Example: (--user is the key option, this is required.)

```
% pip install cupy-cuda91 chainer --user
```

(use pip3 instead of pip if you are using Python3)

Installation Details

Webpage

https://www.anaconda.com/

Version

2020.02

Files Required

- Anaconda3-2020.02-Linux-x86_64.sh (individual version、BSD 3-clause license)

Installation Procedure

The following commands are run by root. Most of logs are cut, since it is too long.

```
% sh Anaconda3-2020.02-Linux-x86_64.sh
...
% /local/apl/lx/anaconda3-2020.02-May/bin/conda shell.bash hook > /local/apl/lx/anaconda3-2020.02-May/conda_init.sh
% /local/apl/lx/anaconda3-2020.02-May/bin/conda shell.csh hook > /local/apl/lx/anaconda3-2020.02-May/conda_init.csh
% . /local/apl/lx/anaconda3-2020.02-May/conda_init.sh
(base) % conda update --all
...
(base) % conda install opencv tensorflow keras
...
(base) % conda install pytorch=1.5.0=py3.7_cpu_0 torchvision cpuonly -c pytorch
...
(base) % conda create --name gpuenv python=3.7 anaconda
...
(base) % conda activate gpuenv
(gpuenv) % conda install cudatoolkit=10.1.243 cudnn opencv
(gpuenv) % conda install tensorflow=2.1.0=gpu_py37h7a4bb67_0 keras
...
(gpuenv) % conda install pytorch torchvision -c pytorch
...
(gpuenv) % conda remove keras
...
(gpuenv) % conda install keras-gpu
...
```

Notes

- (switching keras to keras-gpu in the last part in gpuenv part is just a fix of mistake... Please never mind.)
- If this "gpuenv" is created by cloning "base", installation of tensorflow failed. Not sure why.
    - "python=3.7" is added to use the same python version as "base" env.
    - "update --all" does not work well for gpuenv installation.
- Installing pytorch from the official channel (pytorch) is undoubtedly a wise choice.