

Python Virtual Environments στο HPC

ARIS

Έκδοση 1.0 - 12/05/2022

1. Δημιουργία Python Virtual Environment

1.1 Φόρτωση των HPC-optimized βιβλιοθηκών για τις GPU

Πριν ξεκινήσουμε, πρέπει να προσδιορίσουμε ποια βιβλιοθήκη επιτάχυνσης γραφικών χρησιμοποιεί ο κώδικάς μας (πχ tensorflow/pytorch) καθώς και ποια έκδοσή της. Αυτό συνήθως αναφέρεται είτε από το δημιουργό του κώδικα ή μπορεί και να υπάρχει στο αρχείο *requirements.txt*.

Ας υποθέσουμε ότι ο δημιουργός του κώδικα αναφέρει ότι απαιτείται η minimum έκδοση του tensorflow να είναι η 2.2.0. Για να δούμε ποιες εκδόσεις του tensorflow υποστηρίζει το ARIS, δίνουμε:

\$ module avail tensorflow

```
-----  
/apps/modulefiles/applications  
-----  
tensorflow/1.10.1gpu tensorflow/1.5      tensorflow/1.8gpu  tensorflow/2.0.0  
tensorflow/2.3.0  
tensorflow/1.12.0gpu tensorflow/1.5gpu  tensorflow/1.9     tensorflow/2.1.0  
tensorflow/2.4.1  
tensorflow/1.14.0   tensorflow/1.8    tensorflow/1.9gpu  tensorflow/2.2.0  
tensorflow-py2/1.14.0
```

και βλέπουμε ότι η έκδοση 2.2.0 υπάρχει. Γενικά για το tensorflow, αν απαιτείται έκδοση 1.x τότε επιλέγουμε την αντίστοιχη έκδοση που έχει και το gpu στο τέλος, για να μπορεί να τρέχει με optimized τρόπο στις κάρτες γραφικών. Δηλαδή, αν ο κώδικας μας απαιτεί Tensorflow 1.10.1 τουλάχιστον, τότε θα φορτώσουμε το module tensorflow/1.10.1gpu. Στις εκδόσεις 2.x δεν γίνεται αυτός ο διαχωρισμός.

Πριν φορτώσουμε το tensorflow 2.2.0 ελέγχουμε τις εξαρτήσεις του με την *module show* από άλλα modules, όπως παρακάτω:

\$ module show tensorflow/2.2.0

```
-----  
/apps/modulefiles/applications/tensorflow/2.2.0:
```

```

module-whatis  Enable usage for TENSORFLOW version 2.2.0 built with GNU Compilers 8
prereq  gnu/8
prereq  cuda/10.1.168
prereq  intel/18
prereq  intelmpi/2018
setenv        TENSORFLOWROOT /apps/applications/tensorflow/2.2.0/gpu
prepend-path  PATH /apps/applications/tensorflow/2.2.0/gpu/bin
prepend-path  LD_LIBRARY_PATH /apps/applications/tensorflow/2.2.0/gpu/lib
prepend-path  LD_LIBRARY_PATH /apps/applications/tensorflow/2.2.0/gpu/lib64
prepend-path                                     LD_LIBRARY_PATH
/apps/applications/tensorflow/2.2.0/gpu/lib64/engines-1.1
prepend-path  LD_LIBRARY_PATH /apps/applications/tensorflow/2.2.0/gpu/lib/engines-1.1
prepend-path                                     LD_LIBRARY_PATH
/apps/applications/tensorflow/2.2.0/gpu/lib/python3.7/lib-dynload
prepend-path  C_INCLUDE_PATH /usr/include
prepend-path  C_INCLUDE_PATH /apps/compilers/gnu/8.3.0/include
prepend-path  C_INCLUDE_PATH /apps/compilers/gnu/8.3.0/include/c++/8.3.0
prepend-path                                     C_INCLUDE_PATH
/apps/compilers/gnu/8.3.0/lib/gcc/x86_64-linux-gnu/8.3.0/include
prepend-path                                     C_INCLUDE_PATH
/apps/compilers/gnu/8.3.0/lib/gcc/x86_64-linux-gnu/8.3.0/include-fixed
prepend-path                                     C_INCLUDE_PATH
/apps/compilers/gnu/8.3.0/include/c++/8.3.0/x86_64-linux-gnu
prepend-path  C_INCLUDE_PATH /apps/applications/tensorflow/2.2.0/gpu/include
prepend-path  PKG_CONFIG_PATH /apps/applications/tensorflow/2.2.0/gpu/lib/pkgconfig
prepend-path                                     PKG_CONFIG_PATH
/apps/applications/tensorflow/2.2.0/gpu/lib64/pkgconfig
-----

```

Πρώτα γράφουμε *module purge*

\$ module purge

και στη συνέχεια φορτώνουμε τις βιβλιοθήκες που μας επισήμανε η *module show* προηγουμένως:

\$ module load gnu/8 cuda/10.1.168 intel/18 intelmpi/2018 tensorflow/2.2.0

Για να επιβεβαιώσουμε ότι έχουν φορτωθεί δίνουμε *module list*

\$ module list

Currently Loaded Modulefiles:

1) gnu/8 2) cuda/10.1.168 3) intel/18 4) intelmpi/2018 5)
tensorflow/2.2.0

1.2 Δημιουργία του virtual environment

Στη συνέχεια, μέσα στο φάκελο της ομάδας μας δημιουργούμε virtual environment με την [virtualenv](#) ως εξής:

```
$ virtualenv --system-site-packages venv
```

```
created virtual environment CPython3.7.6.final.0-64 in 4270ms
```

```
creator CPython3Posix(dest=/users/pa19/gealexdl/venv, clear=False, global=True)
```

```
seeder FromAppData(download=False, pip=latest, setuptools=latest, wheel=latest,  
via=copy, app_data_dir=/users/pa19/gealexdl/.local/share/virtualenv/seed-app-data/v1.0.1)
```

```
activators
```

```
BashActivator,CShellActivator,FishActivator,PowerShellActivator,PythonActivator,XonshActivator
```

Αν η παραπάνω εντολή δεν δουλέψει (π.χ. βγάλει έξοδο -bash: virtualenv: command not found), τότε δοκιμάστε εναλλακτικά την παρακάτω:

```
$ python -m venv --system-site-packages venv
```

Όπως μπορούμε να δούμε, μέσα στον φάκελο της ομάδας μας έχει δημιουργηθεί υποφάκελος με όνομα *venv*. Μπορούμε να φορτώσουμε το virtual environment χρησιμοποιώντας την εντολή *source venv/bin/activate*. Όταν το κάνουμε αυτό θα δούμε στην αρχή του prompt μέσα σε παρενθέσεις το όνομα του virtual environment. Π.χ. όπως παρακάτω:

```
$ source venv/bin/activate
```

```
(venv) [gealexdl@login02 team0]$
```

1.3 Εγκατάσταση των requirements

Τώρα πλέον είμαστε έτοιμοι να εγκαταστήσουμε τα requirements του κώδικα μας. Μπορούμε να τα κάνουμε ένα-ένα με την εντολή *pip install --upgrade ...* ή μέσω *pip install -r requirements.txt*, αν μας δίνεται αρχείο requirements.txt. Ωστόσο σε αυτή τη δεύτερη περίπτωση θέλει **προσοχή**, για να μη γίνουν overwrite οι optimized βιβλιοθήκες για το HPC (tensorflow/pytorch) με τις generic που βάζει το pip. Για το λόγο αυτό, θα μπορούσατε να κάνετε edit το requirements.txt και να αφαιρέσετε τις βιβλιοθήκες για tensorflow και pytorch (αν χρησιμοποιείτε keras, μπορείτε να το εγκαταστήσετε κανονικά από το requirements).

Αφού τελειώσετε με την εγκατάσταση, δίνετε *deactivate* και έτσι βγαίνετε από το virtual environment.

```
$ deactivate
```

```
[gealexdl@login02 team0]$
```

Όπως συμβαίνει και με τα python virtual environments που χρησιμοποιείτε και στον υπολογιστή σας, η διαδικασία της δημιουργίας γίνεται μόνο μια φορά. Από τη στιγμή που το έχετε φτιάξει και του έχετε περάσει τις βιβλιοθήκες που θέλετε, δεν χρειάζεται να ασχοληθείτε άλλο με αυτό.

2. Χρήση Python Virtual Environment

Για να χρησιμοποιήσετε το virtual environment που μόλις φτιάξατε σε SLURM script θα πρέπει να κάνετε τα εξής:

1. Θα πρέπει αρχικά να δώσετε `module purge` και ακριβώς από κάτω να φορτώσετε τα modules ακριβώς με τη σειρά που το κάνατε στο βήμα 1.1 αυτού του οδηγού
2. Πριν από το `srun python3 ...` θα πρέπει να κάνετε `activate` το virtual environment σας χρησιμοποιώντας την `source [path-to-venv]/bin/activate`
3. Μετά την `srun python3` θα πρέπει να κάνετε `deactivate`

Ένα ολοκληρωμένο script μπορείτε να βρείτε στη διαδρομή `/users/pa19/gealexdl/team0/slurm-venv.sh`

3. Γενικές Παρατηρήσεις

Χρησιμοποιήστε αποκλειστικά `virtualenv`. **Μην χρησιμοποιείτε `conda/anaconda` γιατί περιπλέκει πολύ τα πράγματα.**