**Using the remote machines on Ucloud**

Pierre Jouannais, 20.02.2023

https://www.claaudia.aau.dk/platforms-tools/UCloud/

**IMPORTANT : WHY AND WHY NOT USING Ucloud**

I see two main uses:

**A)** If you need a lot of MEMORY, for instance to manipulate huge datasets and plot crazy figures with R or python.

**B)** If you need speed to calculate a lot of similar tasks (e.g. stochastic iterations, applying a model for each row of an enormous dataframe etc.), you can make good use of the biggest machine with 32 or 64 CPUs and substantially divide the calculation

HOWEVER, this mainly improves the speed if your code is written to make the best of many cores (parallelized tasks). If so, using the big machines will drastically increase the speed by a factor 20 , 30 or something like that. See the dedicated notebook for Super Fast MC designed for parallelized work.

IF YOUR CODE IS NOT parallelized , using the server may actually not gives you any additional speed because tasks are performed one after another using 1 single CPU instead of doing one task on each CPU at the same time. Actually, if your code involves disk writing activities such AS CHANGING BW DATABASES; I have experienced that using the server many actually be way slower than locally.

Except for this extreme case, even if your code is not parallelized, using the big machine may bring some improvement thanks to the big memory and the fact that some built-in functions in python are coded for parallel work. So you can try anyway!

FYI it’s not very complicated to turn a “classic” python code into parallel tasks using the package ray <https://docs.ray.io/en/latest/ray-overview/index.html> (crazy documentation but the basis use are easy).

Your files in the drives are permanent across sessions, so closing your job does not destroy your folders. **BUT** the environments and brightway projects are not permanent and you will need to go through the steps everytime you start a new job. Maybe there is a way to avoid this but I have not found it. After a few times, it only takes 5 min to do this though. For R it takes even less time because I directly download packages from the R interface.

**WORKFLOW IF YOU WANT TO DO LCA USING BRIGHTWAY ON JUPYTER**

Here I describe my normal workflow to perfom LCAs with brightway on the server.

***Required material***

We can always reconstitute a conda environment “manually” from the server by downloading packages from the server’s terminal with conda which is already installed. But here I want to create a clone environment of the one I am usually locally, by sharing a yml.file.

I will also share my brightway project with all my databases.

To share all this:

1. To create the environment file. Open your local terminal and conda, activate your environment and then tap:

conda env export > environment\_name.yml

This will create the environment file in the current location.

1. To create the brightway project file, refer to the code instructions export/import in the dedicated notebook.
2. Put the 2 files in the server’s folder.

***Start the instance***

Once you have created a “drive” in the files tab on Ucould, you can start a “Run” by opening the “Apps” Tab.

-Choose the App you want to use.

-Choose the amount of hours you want your instance to be running. (You can always kill the job yourself if you finish earlier, or add some extra time during the process from this same page.)

Graphical user interface, application, Word

Description automatically generated

-Choose the machine

-Choose your Drive as “Folder to use”

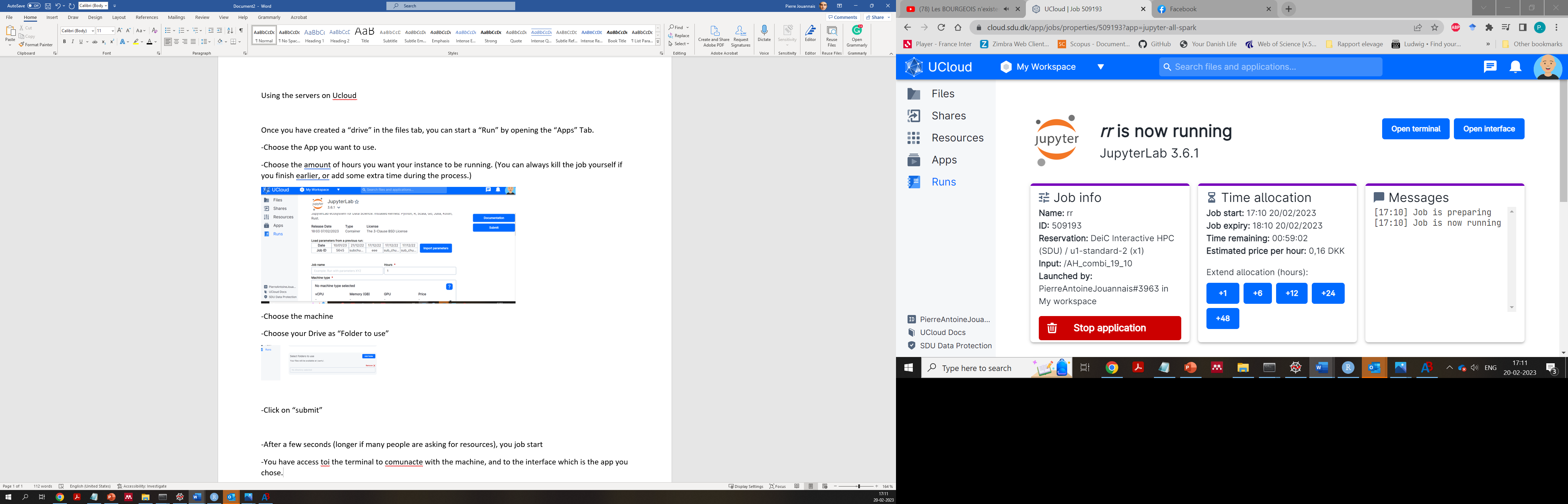
Graphical user interface, application, Word

Description automatically generated

-Click on “submit”

-After a few seconds (longer if many people are asking for resources), your job start

-You have access to the terminal to communicate with the machine, and to the interface which is the app you chose.



You have access to your folder, but you need to recreate your environments with the packages you need.

***Set the environment ready for jupyter notebook***

Now from the server’s terminal,

1. Open terminal
2. Tap : conda init bash
3. Close terminal
4. Open terminal again and get in your folder. Note that when you open the terminal, you are located somewhere above your folder in the architecture. Your folder is given a random number for some reason. Here mine was named 287166.

Here I access my folder as follows:

Graphical user interface, text, application

Description automatically generated

1. I can see my environment\_microalgae.yml file. I can now reconstitute my environment from it.

Tap: conda env create -f environment\_microalgae.yml

When done, I want to be able to choose my environment from the jupyter interface. To do this:

1. conda activate environment\_microalgae
2. conda install ipykernel
3. ipython kernel install --user --name=envi\_micro\_1 ( JUST REPLACE WITH THE NAME OF YOUR ENV)
4. conda deactivate
5. Now open the app interface with the button from Ucloud.

Graphical user interface, application, Word

Description automatically generated

1. From your jupyter notebook, go to the upper right corner and click on “Python 3” to set your environment as a kernel.

Graphical user interface, application, Word

Description automatically generated

1. Select your environment in the list (here I had not set my environment)

Graphical user interface, application, Word

Description automatically generated

Now everything you run from the notebook will use your environment

You can now run the code to import your brightway project. (cf. dedicated notebook).

You are set. If you create some outputs from the notebooks it will be saved in your folder and you can directly see them in your corresponding U cloud folder (need to refresh the page sometimes), and download them.

Graphical user interface, application, Word

Description automatically generated