Centre de Calcul
de l'Institut National de Physique Nucléaire
et de Physique des Particules

The Jupyter Notebooks Platform

April, 14th 2022

Bernard Chambon, Sébastien Gadrat, Natthan Pigoux

Outline

- Overview of the platform
- Infrastructure description
- Advanced features provided to the users
- Monitoring and current usage of the platform
- Perspectives

Overview of the platform 1/2

JupyterLab based web application

- **Modern GUI for notebooks accessible from an URL**
  - https://notebook.cc.in2p3.fr
  - Connection either with CC computing credentials or with EduGain

- **Allows to develop and run code interactively (in various langages)**
  - Currently providing Julia, Python, R, C++/ROOT (Cling based), Golang
  - Custom kernels (with specific environment) can be easily created & used

- **Providing a Unix terminal**
  - All softwares provided by CC-IN2P3, i.e. /pbs/software/ is available

- **Customizable and extendable (thanks to extensions/widgets)**

Overview of the platform 2/2

Memory usage

Kernel currently in use in the notebook

File Edit View Run Kernel Git Tabs Settings Help

179 / 28672 MB

Launcher | Deblending_with_deep_learn

Markdown

Python 3.8 - Fidle on GPU

Filter files by name

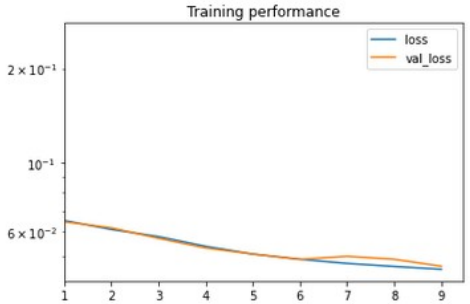
/ notebooks /

Name	Last Modified
cpu-gpu	2 days ago
data	7 months ago
ipympl	2 years ago
ipywidgets	7 months ago
jnp-io-performance	a year ago
julia	22 days ago
logs	7 months ago
mydask	a month ago
qml	3 months ago
test	3 months ago
check_ipympl.ipynb	14 minutes ago
Deblending_with_deep_learning_tf...	8 minutes ago
DynamicPlottingInNotebooks.ipynb	a year ago
julia_demo.ipynb	14 minutes ago
List_R_packages.ipynb	2 years ago
ProblemeMatplotlib.ipynb	9 months ago
ProblemeMatplotlib.md	9 months ago
processSpectra_holo-newspectrac...	6 months ago
scikit-parse.ipynb	5 months ago
SimpleROOTfit.ipynb	5 months ago
test_Go.ipynb	a year ago
test_Julia.ipynb	17 minutes ago
test_R.ipynb	20 minutes ago
test.tex	2 years ago
TestJsROOT.ipynb	10 months ago
tex_demo.ipynb	8 months ago
tex_demo.png	a year ago
tex_unicode_demo.py	2 years ago
Untitled.ipynb	7 months ago
untitled.py	7 months ago
Untitled1.ipynb	9 days ago
Untitled12.ipynb	18 minutes ago

Score: 0.57

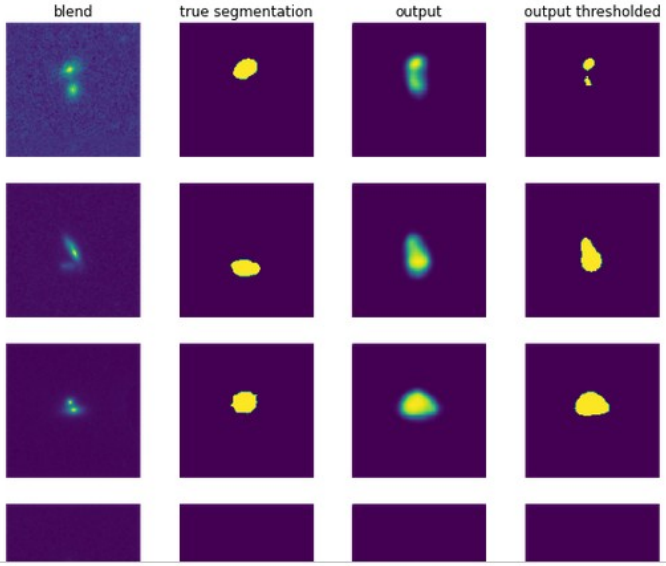
Plot the history

[21]:
```
plot_history(history)
```



Plot some examples

[22]:
```
plot_random_results(obj.model_, X_test, Y_test)
```



Simple 0 1 Python 3.8 - Fidle on GPU | Idle Mem: 179.43 / 28672.00 MB Mode: Command Ln 1, Col 1 Deblending_with_deep_learning_tf2.ipynb

Infrastructure description 1/3

How it works?

- **Access through JupyterHub**
  - HTTP proxy
  - Handles authentication (OAuth)
    - Keycloak for user authentication (and retrieve the uid/gid)
    - IDNUM for account validity & secondary groups
- **Hub's config (jupyterhub_config.py)**
  - Spawn a Docker container
    - Notebook server
    - $HOME and related storage (according to groups) will be mounted ('bind-mount') into the container

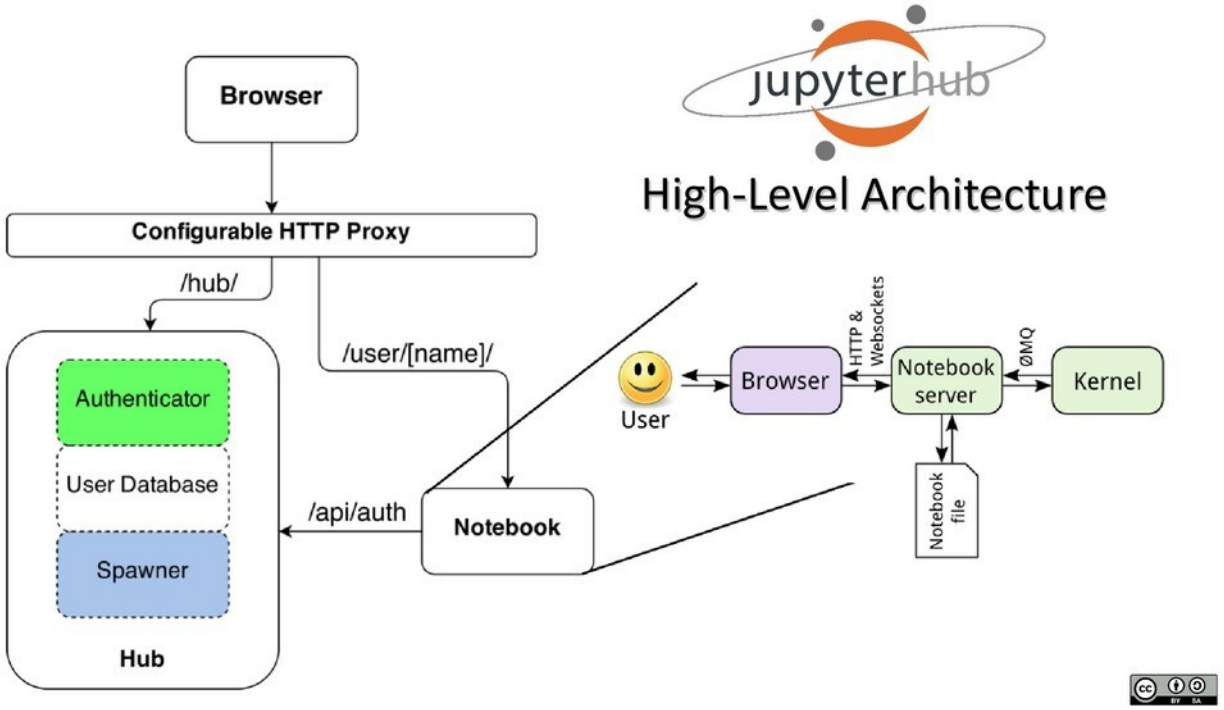


Image taken from https://speakerdeck.com/jhermann/jupyterhub-and-jupyter-notebook-a-view-under-the-hood

Infrastructure description 2/3

How it works?

- **Dockerspawner**
  - The Docker container ‘notebooks server’ will be instantiated using the user’s identity (uid, groups, HOME)
    - Basic Docker image similar to the compute batch farm (based on CentOS7, mostly same libraries)
    - $HOME and related storage will be available
      - HOME, SPS depending on which groups the user belong to
      - THRONG_DIR, PBS_SOFTWARE and CVMFS is provided to all users
      - Automatically mounted by the Dockerspawner inside the container, according to config files
- **Hardware (basic config managed by Puppet)**
  - OpenStack VMs
    - 1 for the JupyterHub
    - 7 workers (for CPU, 5 x 32GB and 2 x 64GB) and 4 dedicated for training (32GB)
  - 5 Nvidia K80 machines (retrieved from GridEngine) providing 20 K80 GPUs
    - Can be used for high memory tasks, as they provide much more memory (130GB per machine)

Infratructure description 3/3

From a user's point of view

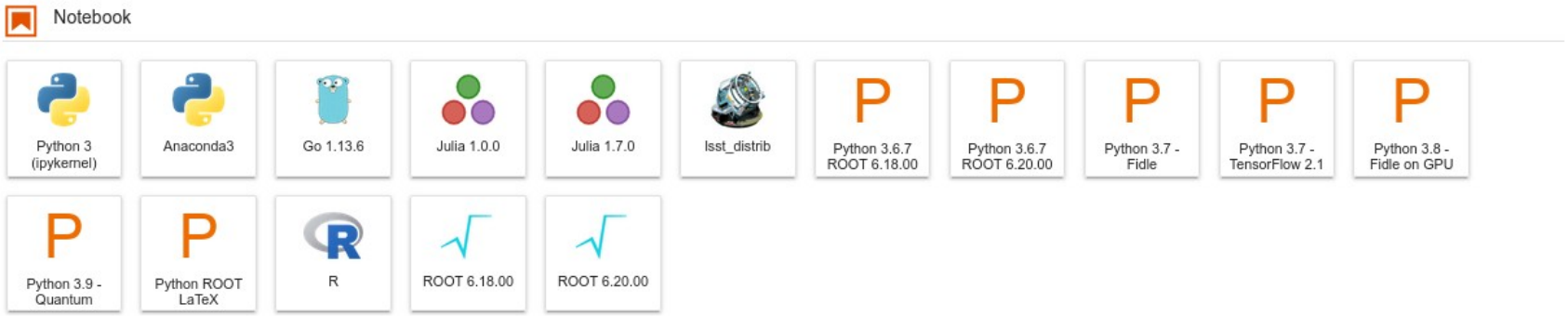
- Notebooks server stays available (with all tabs and running kernels)
  - Up to 3 idle days if CPU (allow to get it back on Monday)
    - Some users have their notebooks server running for several months!
  - Up to 1 idle day if GPU
- Some dedicated config files to handle notebooks server placement
  - Currently 'whitelisted' users to access GPU (as limited number of GPUs available)
  - Placement can also take into account others requirements (training, resources, …)

From an admin's point of view

- User's resources (memory) and associated storage can be updated on the fly
  - Config files read by the dockerspawner at the container instantiation
- The JupyterHub container can be updated without impacting users
  - Currently using an external proxy (from Hub) → hot release of the new Hub version
  - Users' config will be updated the next time they instantiate a new notebooks server

Advanced features provided to the users

Custom kernels



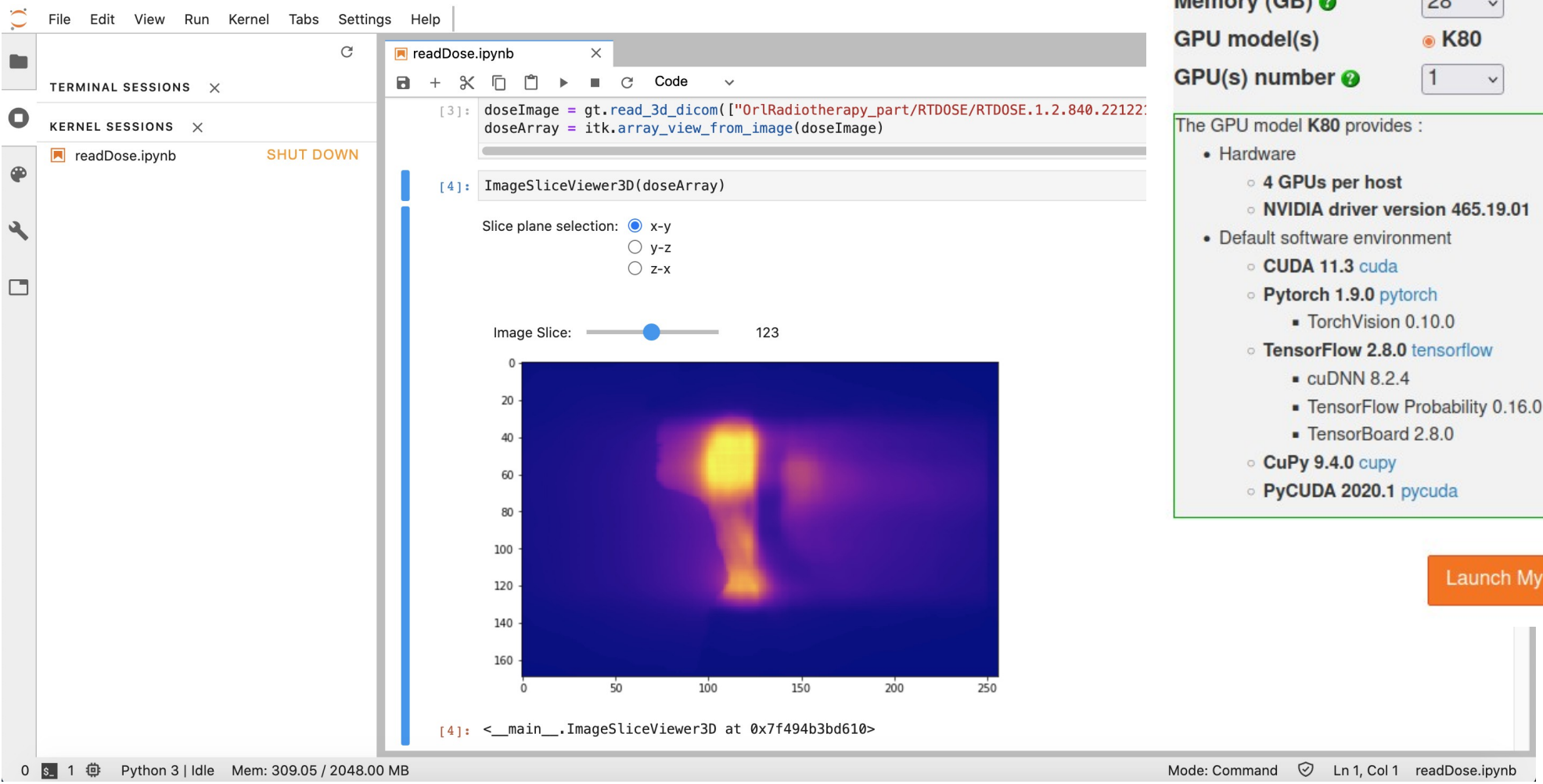
Easy procedure to create such kernels:

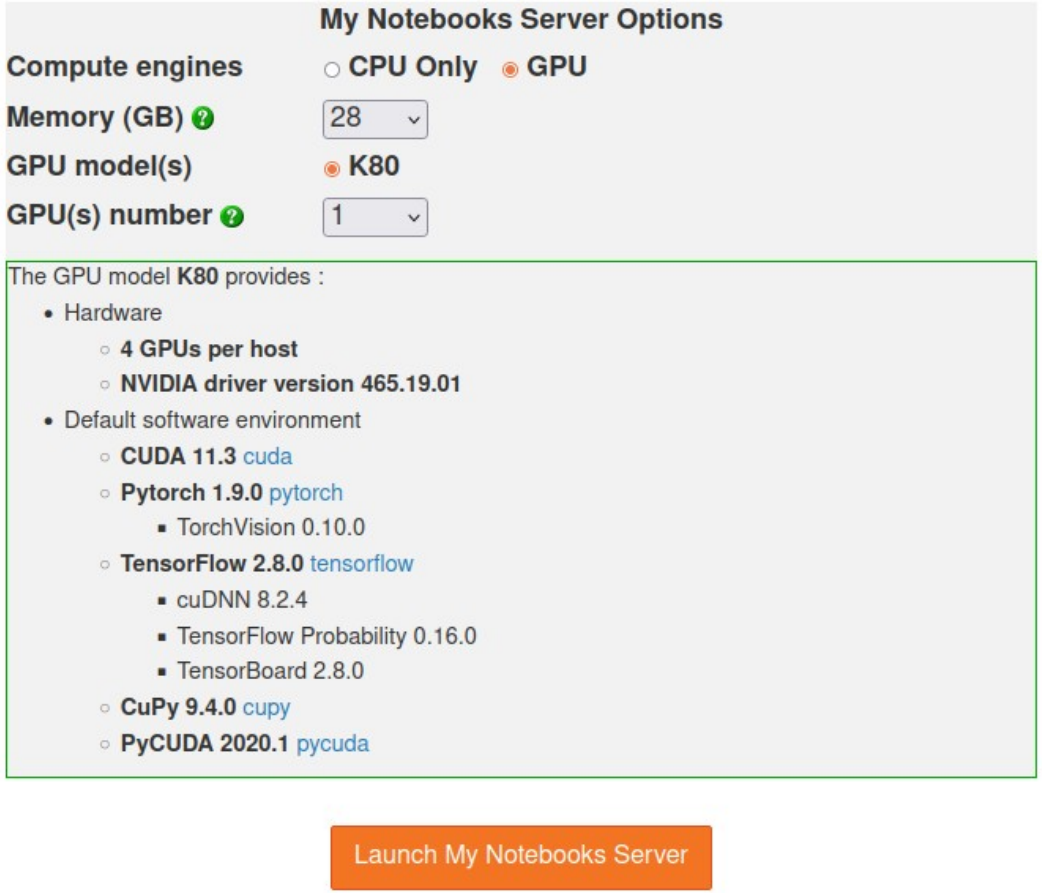
https://doc.cc.in2p3.fr/fr/Daily-usage/daily/jn-platform.html#noyaux-disponibles

Extensions & widgets

- Extend & add new features
  - ipympl, ipywidgets: add interactivity with Python plots (especially with matplotlib)
  - Voilà (export & conversion of notebooks), Git, control version of notebooks, …

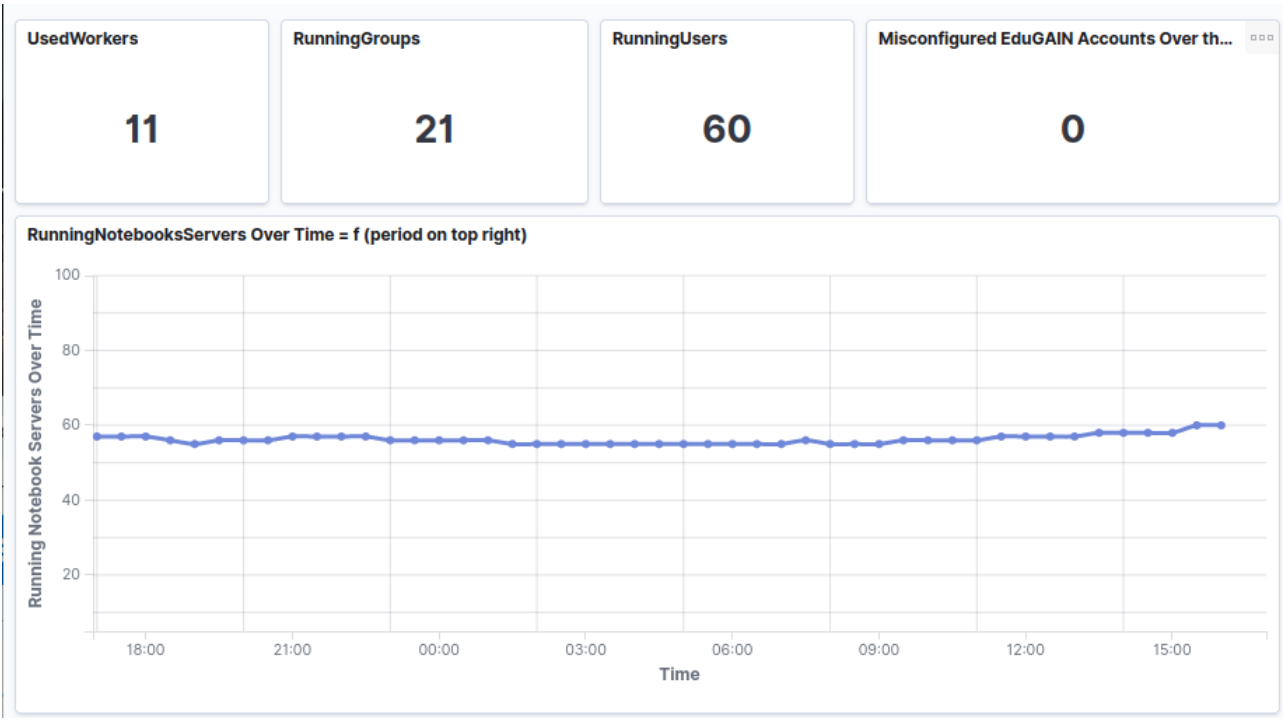
Advanced features provided to the users

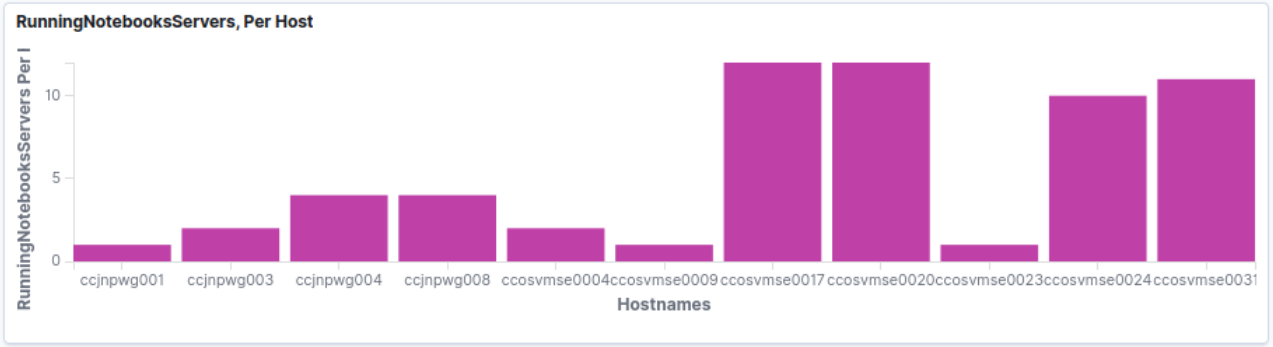


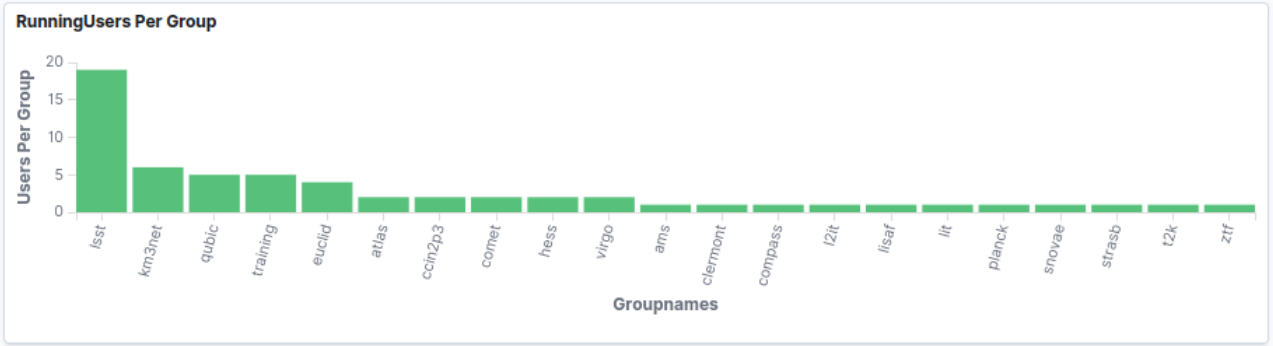


Monitoring and usage of the platform

Monitoring based on ElasticSearch, Grafana stack







Monitoring and usage of the platform (GPU)

GPUs Installed

20

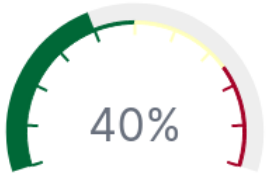
GPUs Used

8

Hosts Used

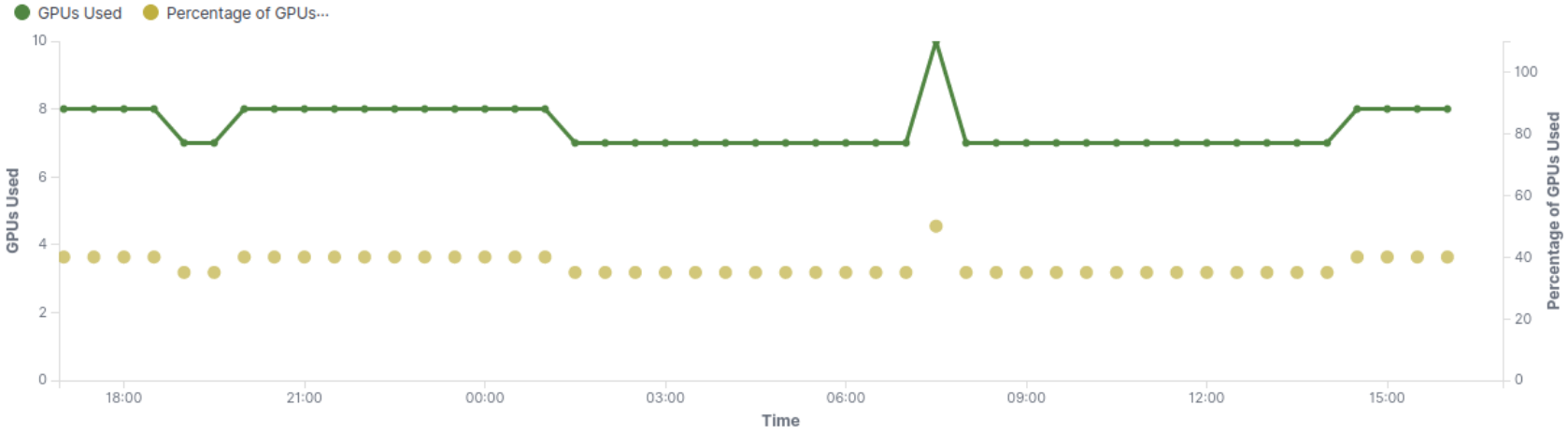
2

Percentage of GPUs Used

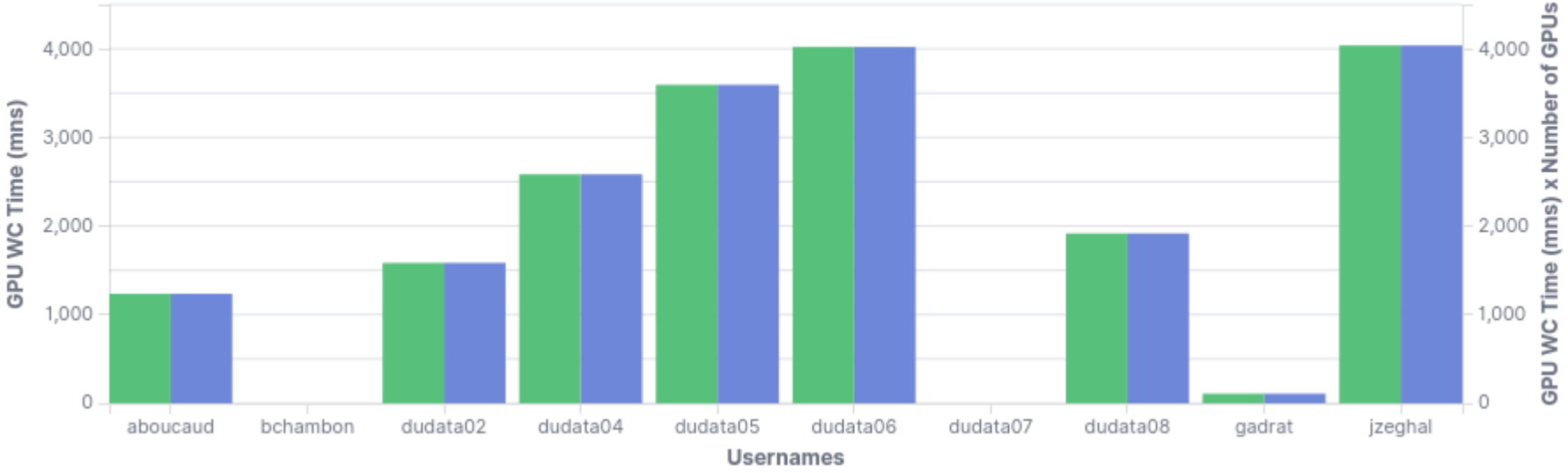


Percentage of GPUs Used

GPUs Used & Percentage Used, Over Time = f (period on top right)



GPUs Wall Clock Time +, Per User



Conclusions and perspectives

The platform usage is increasing

- At least 60 users always logged in (spread in more than 20 groups)
- GPUs (Nvidia K80) begin to be used as well
  - Provided to users for about 1 year (whitelisted only for the time being)
- Providing default (almost) up-to-date Python environment for both CPU and GPU
  - Currently based on Python 3.8.5
- Config files can be updated ‘on the fly’, which quite eases to deal with users’ demands
- Very good feedbacks from current users

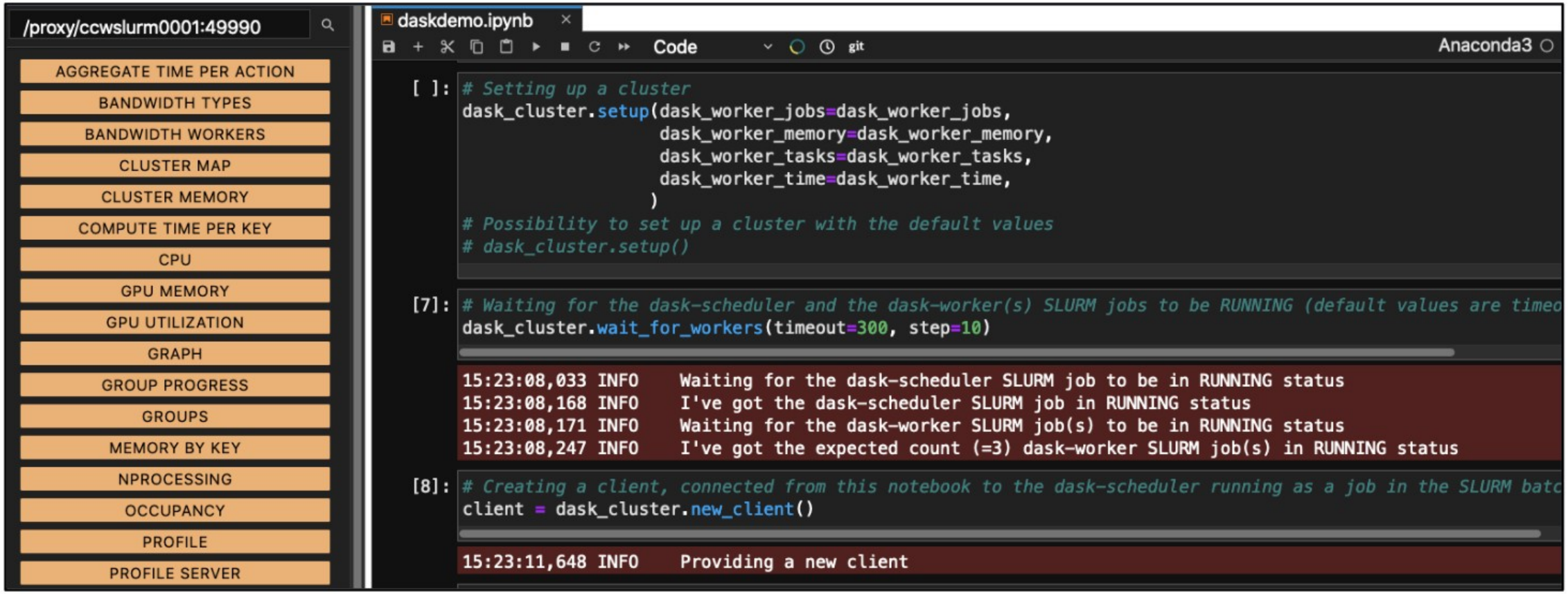
Perspectives

- Main request from users would be to be able to run distributed tasks on the compute farm (Slurm) using Dask
  - Proof-Of-Concept on-going, first results quite promising!

Conclusions and perspectives

Dask https://dask.org/

- Some codes from the current PoC
  - 1 scheduler and 3 workers being spawned in Slurm through the Jupyter Platform



Thanks! Questions?