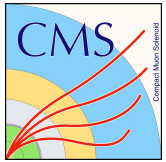
SWAN platform: user feedback

Davide Valsecchi (from CMS), et al.

14/07/2022



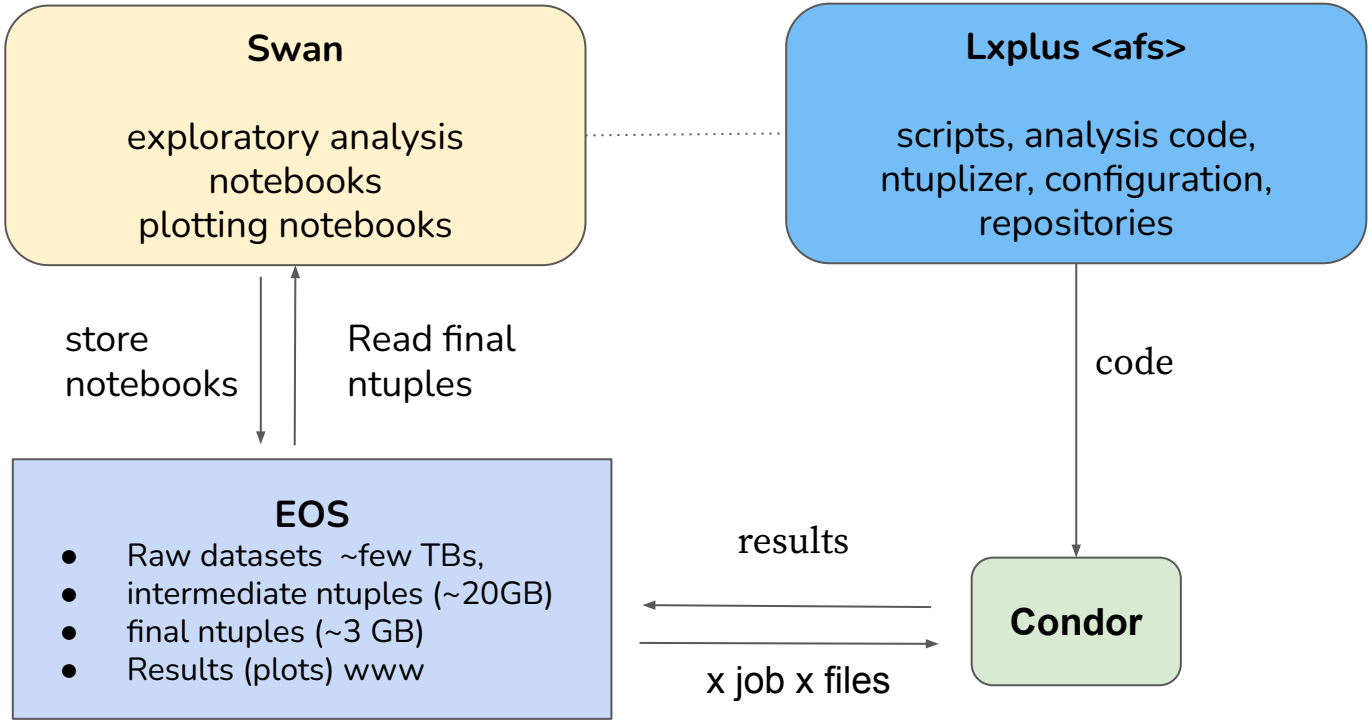
ETH zürich

WARNING

Be aware! Physicist giving opinions on software!
This talk reflects only my biased version of the Lagrangian

- I'm a CMS postdoc working on data-analysis and ML.
- I mostly used Swan in the last 3-4 year (my Phd) for plotting and prototyping
- I may be unaware of the latest developments or cutting edge features
- I extensively used CERN IT resources as lxplus, EOS, HTCondor in a "classical way"
    - I came from classical TTree::Draw ROOT and then moved to RDF (thankfully)
    - I have just tried Spark once and I'm starting to work with Coffea more
    - I have filled HTCondor with many jobs many times manually :)

A template for many of my computing tasks

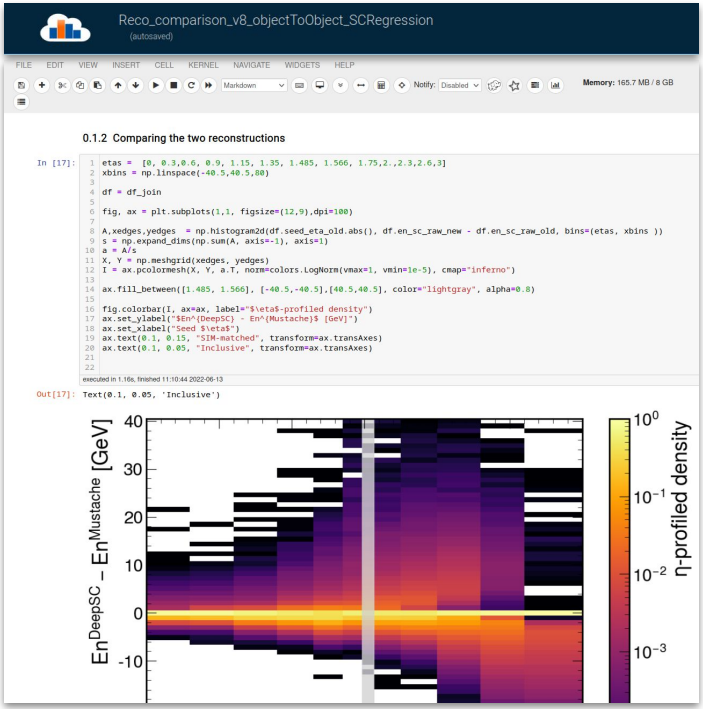


Swan in *my* workflow

- **Swan fits very well my needs for:**
  - prototyping code and algorithms
  - plotting final results
  - working on ML models interactively

- It **fills the gap** between:
  - full-scale analysis (condor jobs)
  - interactive play with the results (difficult to do by running scripts on lxplus) == definition of the jupyter notebook ;)

- **Huge PROs**
  - access to EOS
  - export of plots on EOS/www
  - quite updated software stack (more on this later)
  - Easy access to GPUs
  - keeps the session active if you disconnect for some time



Pros, and Cons..

- Notebooks loose the **connection** randomly (happening more frequently from the migration to K8)
- **EOS storage/connectivity** problems:
  - lost notebooks changes or “disk full” errors.
  - Keeping git repositories with many files on EOS gave me many problems in the past
- The console access is very very limited
  - I use it rarely but it can be useful.
- **Python environments handling and kernels**
  - managing python packages can be cumbersome: conflicts with LCG, other user packages etc
    - e.g. tensorflow packages, pytorch, multiple versions of some package that are needed to interact with other packages for different projects..
  - Would it be possible to work more in a “conda environment style”?
  - No possibility to run additional flavours of kernels from the user environment

Conclusions

- **Dream features:**
  - A nice editor (Jupyter notebook editing is quite limited) to be able to work on the “library” part of the code and keep in notebooks only the steering/results analysis.
  - Nicer interactive shell
  - == Jupyter Lab

- **Pain points:**
  - Sharing by cloning is very limited: never really used in practice
  - Jupyter notebook versioning: I usually do it by creating copies of the notebooks *(v1, final, _veryfinal)*
  - **Reproducibility**: running many times with different inputs and check outputs.
    - Usually done creating copies of the notebook → then problems with sync of the code
    - At some point dump the code to python scripts → not easy to run from Swan

Ending thoughts

- Moving towards new ways of handling the analysis....
  - With RDF, Coffea, the user needs to interact less and less manually with the job systems
    - This is very **nice**!
  - I would like to move on Swan also some heavy lifting operations (by RDF-distributed, Coffea)
    - I think this is quite a close target... but I feel that running code on lxplus is far to be dead..
    - Swan for full-scale analysis should make easier to edit library code, configuration, running code (not only notebooks)
    - Jupyter notebooks are perfect for exploration, but not so much for versioning / sharing / parametrized execution