

Seamless Scaling of Jupyter Notebook Workflows with SLURM: From a Single Catchment to the Full Caravan Dataset

Mark Melotto^{TUD, 1, and 1}

^{TUD}ADDRESS

¹ADDRESS

Correspondence: NAME (EMAIL)

Abstract. TEXT

Copyright statement. TEXT

1 Introduction

This paper introduces a software package that is able to expand one particular workflow on a subject to the same workflow on multiple subjects. This is a tandem paper with the eWaterCycle climate change analysis ? project. This analysis is a good example of the use of this package. What do you need to go from a hydrological analysis on one particular region, to all the regions over the world? This is not a question that is answered by many hydrologists, as they are no computer scientists after all.

Instead of containerization of the multiple regions, this package is designed to scale from a local desktop or cloud environment to any SLURM HPC environment. HPC environments are almost all Linux based systems. Keeping in mind that the average hydrologist is not trained in Linux, this package has to solve that as well. The package gives requirements to the workflow and then delivers the correct bash, SLURM & Python scripts to run on HPC. The entire workflow can then be run from a single Jupyter notebook, but for convenience this one notebook is split up into to more parts: controlling the jobs, moving files, updating the project, and monitoring the system. These notebooks will omit the command line, but in special cases unfortunately, command line is necessary. This is all explained within the notebooks themselves.

1.1 eWaterCycle

eWaterCycle is a hydrology software package, built and designed to do FAIR research. This allows bachelor students to do impactful hydrological research as well as PhD students. eWaterCycle has the required building stones to make the seamless package work. It allows its users to build reproducible notebook workflows. This workflow is what makes this package possible in hydrology. Two of the main philosophies of eWaterCycle are: the user should not use the command line, and the software

headaches are taken away from the user. This enables hydrologists to fully focus on hydrology and the eWaterCycle team on giving them the tools to do their research.

1.2 Importance of FAIR and usability

Hydrology is plagued by irreproducible work ?? (????). Not only is this a problem when urging governments for change, but also in the collective field of Hydrology research. Irreproducible research does not contribute as much to the field as when it is made reproducible. Making way for the disconnection between hydrologists and the many many models. This then also requires new researchers coming in to learn the niche ways the 'researchers' model works. That is why this package is built in such a way that it is accessible for students, as well as professors. It does, not only require reproducibility, it also enforces it. Being built in such a way that it is also usable outside of hydrology, hopefully spreads the message, and what we have learnt along the way, of FAIR research to the broader research community. Enabling better communication and cooperation between research institutes.

2 Methodology

The aim of the package is to adapt your workflow minimally whilst it will still work as a stand-alone. But now is also able to be pushed towards a super computer.

2.1 Backbone: Workflow Standardization

Research with Python is reproducible when done with Python notebooks. The inherent addition of markdown cells allows the creator to explicitly explain what is happening, in something that is not doc-string or comments. Allowing for a nice structure and short pieces of code that are readable and produce the desired results.

2.1.1 Papermill

Papermill ? is a Python package that executes Python notebooks and saves the output to the desired directory. Giving Papermill some arguments, gives the opportunity to run the same workflow notebooks, but with different arguments - think subsets, regions or parameters. In the twin paper we change the region of the research analysis; parsing a country and region id to the Papermill script.

Combining this with a bash and SLURM script, it is possible to submit jobs to a HPC

2.2 HPC Cluster

Simple Linux Utility for Resource Management or SLURM is a workload manager and a job manager used on many high performance computing clusters and/or supercomputers. SLURM is used widely in research HPCs, making it easy to switch between different HPC systems with this package.

SLURM requires set parameters to assess the allocation of required hardware and time. These are given in the SLURM scripts. These parameters are mostly: time, memory, cpu, nodes needed & job names. In our case it also initialises a special conda environment that is made beforehand to have eWaterCycle installed. Then it mounts the web servers that host climate data and finally, it moves to the correct directory and then it runs the Python script that contains Papermill.

2.2.1 Spider Cluster

Spider is a HPC cluster from SURF and focusses on high throughput compute jobs. Meaning that they are more data (I/O) heavy, rather than compute power heavy.

- SLURM
- SLURM Script
- Chosen for Spider
- why, how, details
- 60 – Data handling

2.3 no CLI and how it is done

Everything is run from separate notebook, which mimics commandlines en ssh connections via Python

2.4 eWaterCycle

Misschien niet eens nodig

65 3 Setup

This package requires the correct setup for it to work. This will be explained here.

3.1 Notebook Structure

The first notebook serves as the setup for the entire workflow run. It should create a JavaScript Object Notation (JSON) or YAML Ain't Markup Language (YAML) file that contains all the necessary paths and settings (Appendix example). This file will differ for all runs because it contains the details of that run. This settings file is then loaded in for all the other notebooks in the workflow. Making it so that only the first notebook has to be parsed with the unique ids from Papermill. The settings file contains:

- Details on that particular run

- Absolute paths to download directory
- 75 - Absolute paths to where the data is saved
- Various settings relevant for your project

3.2 File Structure

The following structure is assumed:

```
project/  
80 |- notebooks/  
   | - figures/  
   | - 0_settings.ipynb  
   | - 1_start_workflow.ipynb  
   | - ....ipynb  
85 | - x_analysis.ipynb  
   |- scripts/  
   | - custom_functions_1.py  
   | - custom_functions_x.py  
- structure_builder.ipynb  
90 - analysis.ipynb  
- README.md
```

That structure will become:

```
project/  
|- notebooks/  
95 | - figures/  
   | - 0_settings.ipynb  
   | - 1_start_workflow.ipynb  
   | - ....ipynb  
   | - x_analysis.ipynb  
100 |- scripts/  
    | - seamless.py  
    | - slurm_script.slurm  
    | - submit_jobs.sh  
    | - cancel_jobs.sh
```

```
105 | - structure.json
    | - custom_functions_1.py
    | - custom_functions_x.py
    |- output/
    | - <your output structure>/
110 |- done/
    | - done.csv
    | - done.csv.lock
    - managing_SLURM.ipynb
    - structure_builder.ipynb
115 - analysis.ipynb
    - README.md
```

In this structure the user only has to supply the notebooks and the optional supplementary scripts. The Seamless package will take care of the rest.

3.3 Duality of Use

120 How does one make sure the workflow works locally and on HPC? Unfortunately, this will differ for every user but, this usually lies in paths. Because this is different for every user, there is a function the user will have to alter.

When running locally, one can simply run the 'notebooks' folder and everything should work. Because the workflow is run from the parent directory of 'notebooks' the paths will change.

3.3.1 Importing custom functions

125 To keep the notebooks folder clean, the custom Python functions are kept in another folder. This function is then used for imports for example.

4 Discussion

- NetCDF issues
- CMIP6 issues, can be left out
- 130 - Scaling worth it?
- Storage

5 Conclusions

Conclusion text

Code availability. TEXT

135 *Data availability.* TEXT

Code and data availability. TEXT

Sample availability. TEXT

Video supplement. TEXT

Appendix A

140 **A1**

Author contributions. TEXT

Competing interests. TEXT

Disclaimer. TEXT

Acknowledgements. TEXT

145 **References**

James H Stagege, David E Rosenberg, Adel M Abdallah, Hadia Akbar, Nour A Attallah, and Ryan James. Assessing data availability and research reproducibility in hydrology and water resources. *Scientific data*, 6(1):190030, 2019.