

Name	Last commit	Last update
M♣ <u>README.md</u>	init	45 seconds ago

README.md

HW5

Pipelines & Paralellism

Goals

The goal of this homework assignment is to explore how to build pipelines with slurm and run parallel jobs.

As before, homework must be completed in Markdown, pushed to a private GitLab repository, rendered to PDF, and then saved back into the repository. A zip file of the repository (including markdown, PDF, any other work files, and exclusing the .git directory) must be submitted for peer grading.

Do not simply fork the repository. For this assignment, clone this repository directly, and then set up a second remote with which you push your changes to. This will allow you to pull changes from the originating repository and continue to push changes to your private repository.

Additionally, set up HW1 as a submodule of this HW so there is a copy of questions 1, 2, and 3 to describe the dataset.

Questions

- 1. Make HW1 a submodule of HW5
- 2. Job-based visualization Firstly, I went in my visualization code and instead of saving the layout I made to an html file, I set up a bokeh server using the HoloViews library as such:

renderer = hv.renderer('bokeh')

doc = renderer.server_doc(final_layout)

doc.title = 'HW5 Visualization'

Then, to connect to the server running on the compute node through an SSH tunnel, I used the command where 5049 is my local port, the sbatch job is run on wheat 14, and the remote port was 32923. To find my local port, I opend my command line and used the command "netstat -a" to print out all of my local ports and chose one of them:

L 5049 : wheat14 : 32923

- 3. Building a singularity container After making setting up my account with sylabs.io, I used the given header for my definiton file and began adding in commands to build up my singularity with the library it needs such as os, dash plotly, h5py, holoviews, pandas, and numpy. I included the my def file in this submission folder.
- 4. Pipeline 101

In order to build by pipeline, I started by making 3 jobs. The first job runs analysis1 that's passed in the hdf5 file as a dataset. Then I used "--depend=afterok:" in my second job request to make analysis2 depend on the first job that was submitted. For the third job I used "swarm -f swarmfile --module blast --depend=afterok:" to create a third job with dependency on the second.