OAK RIDGE National Laboratory

Analysis Workflow on Jupyter

Benjamín Hernández

AI & Analytics Methods at Scale Group

Data Visualization and Analytics Training Series
July, 14th 2022

ORNL is managed by UT-Battelle LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

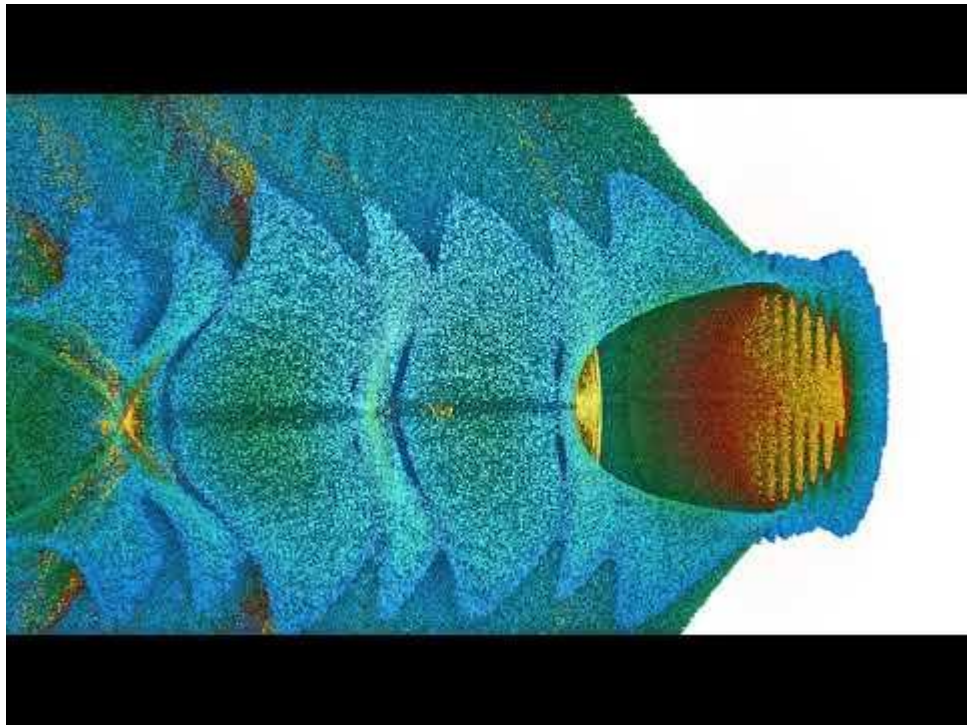
This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

Contents

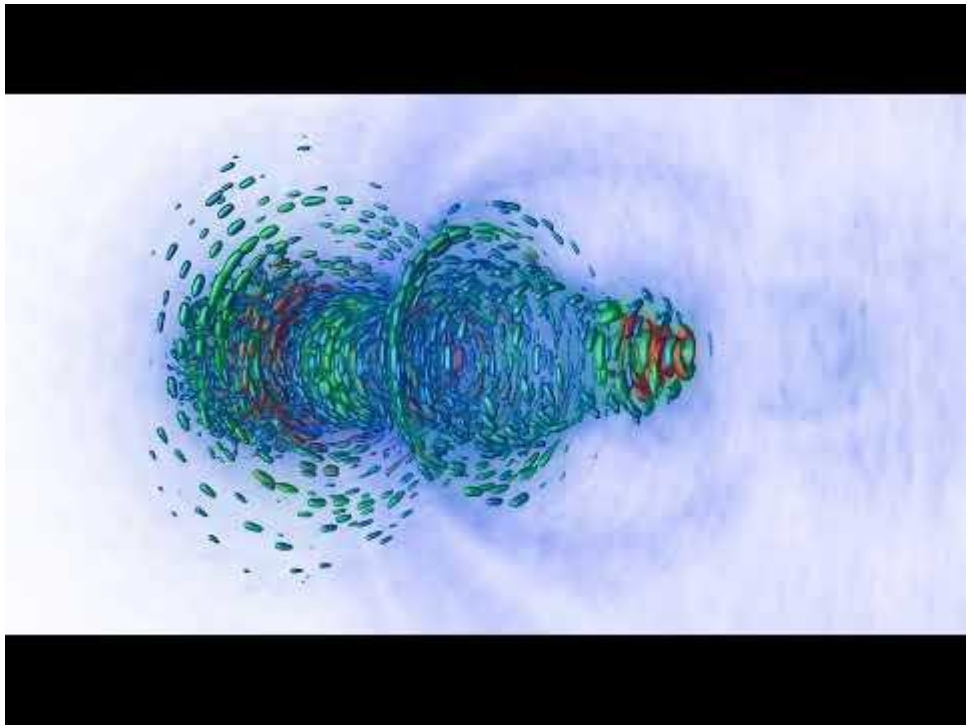
1. Dataset Overview

2. Hands On
   a. How to create a custom environment in Jupyter@OLCF
   b. How to scale analysis with dask using Summit

PIConGPU
Laser wakefield acceleration

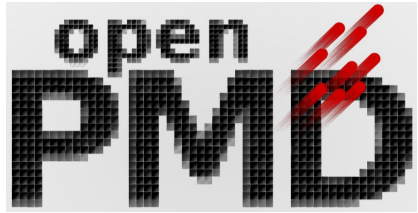


Particle visualization



Volume and
Vector field visualization

Dataset Specifications



4 GPUs

`4gpus-openpmd`

Domain: 128x2048x128, 33.5 M cells

Macroparticles: 67.11 M

Total size: 97 GB

8 GPUs

`8gpus-openpmd`

Domain: 192x2048x160, 62.9 M cells

Macroparticles: 125.8 M

Total size: 186 GB

Time steps: 2048

Output files: 64

Format: openPMD

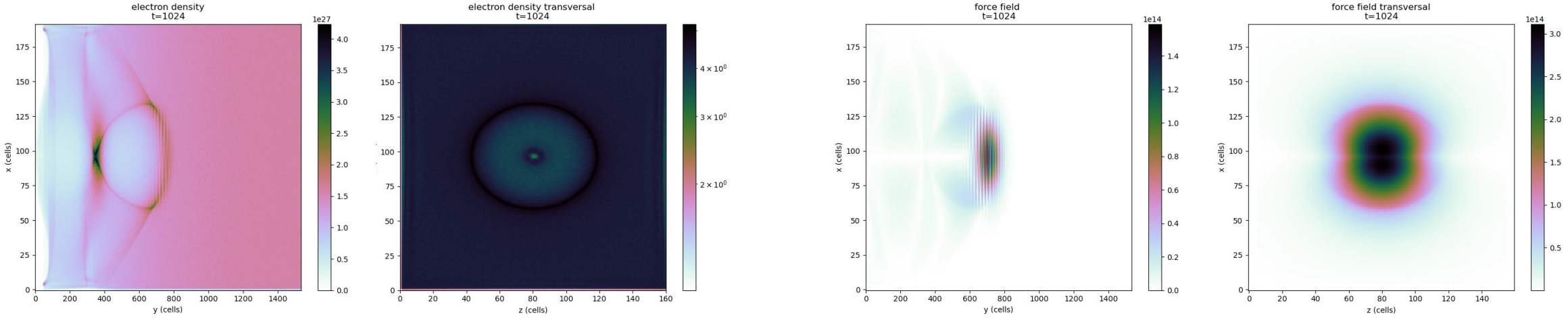
Location:

`/gpfs/alpine/world-shared/stf218/analysis_viz_training/07142022/datasets/lwfa`

Notebook

Main tasks:

1. Load time series
2. Read attributes and records
3. Calculate secondary variables (particle density, force field)
4. Plot results, lateral and transversal views



Hands on Overview

Objectives:

1. How to create a custom environment
2. How to scale analysis with dask using Summit

All materials are available under

`/gpfs/alpine/world-shared/stf218/analysis_viz_training/07142022`

Directories:

`../dask-lsf-script`

`../datasets`

`../envs`

`../notebooks`

Hands on Overview

Objectives:

1. How to create a custom environment
2. How to scale analysis with dask using Summit

Four notebooks are provided

`analysis-lwfa-simple.ipynb`

`analysis-lwfa.ipynb`

`analysis-lwfa-dask.ipynb`

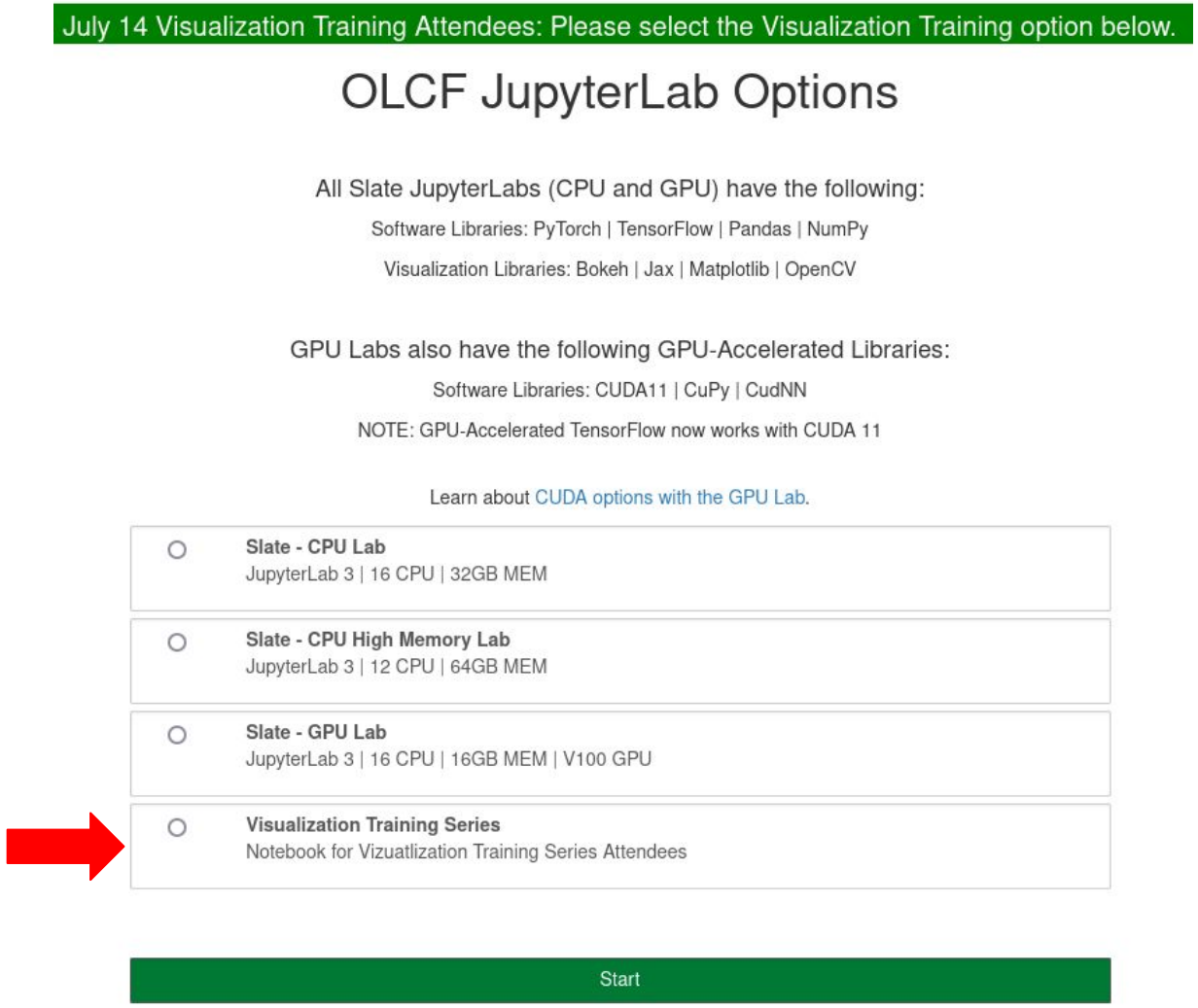
`analysis-lwfa-dask-summit.ipynb`

You can also find them in:

`/gpfs/alpine/world-shared/stf218/analysis_viz_training/07142022/notebooks/`

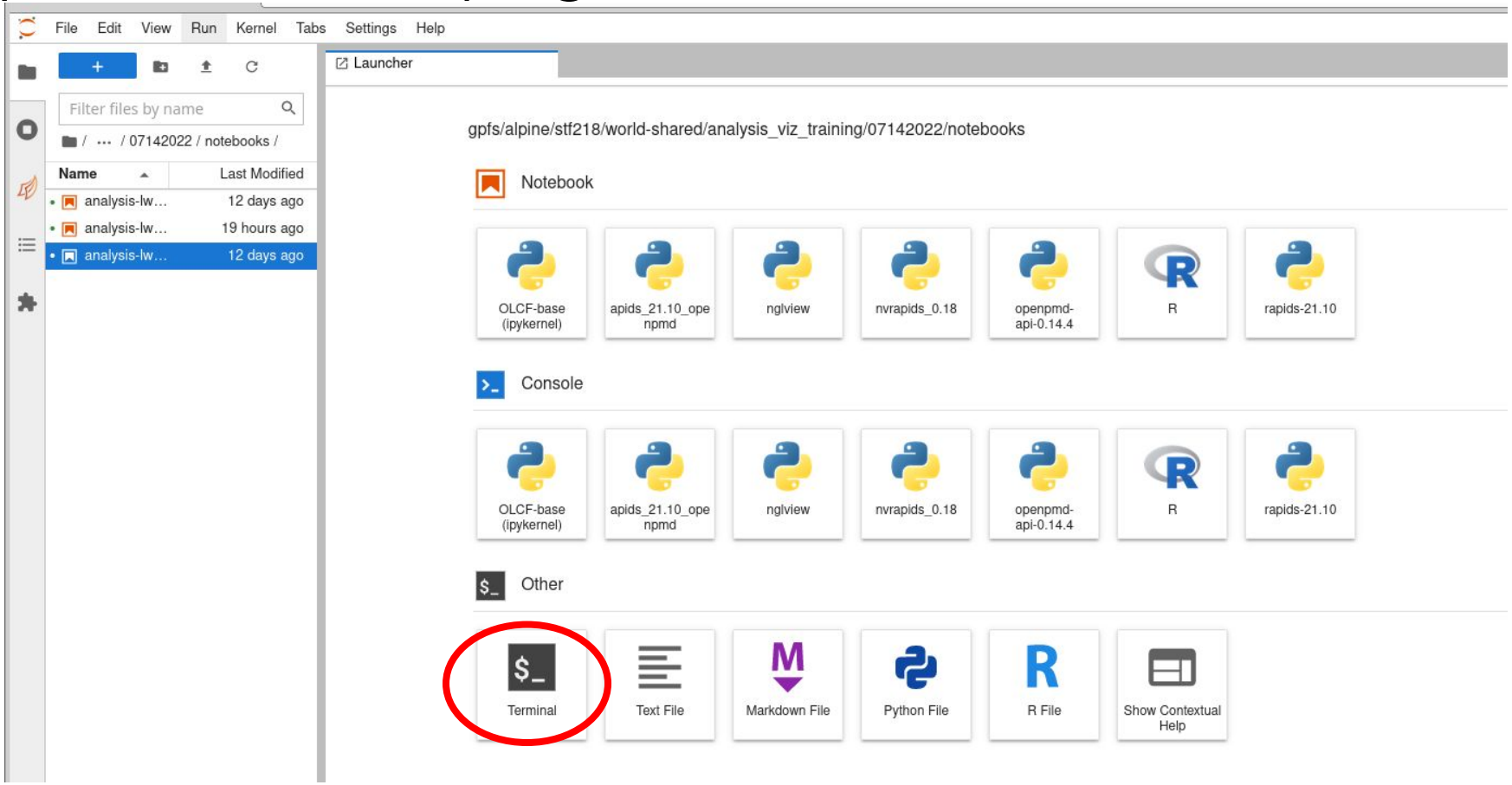
Hands on
How to create a custom environment

- Start a Lab from https://jupyter.olcf.ornl.gov
- Select "Visualization Training Series" Instance



Hands on
How to create a custom environment

- From your Launcher page, click on Terminal.



Hands on
How to create a custom environment

- In the terminal, create a conda environment for OpenPMD:

```
conda create -p <my_dir>/openpmd -c conda-forge openpmd-api
ipykernel pandas scipy matplotlib numpy dask=2021.09.1
distributed=2021.9.1 cloudpickle=2.0.0 msgpack-python=1.0.2
python=3.7.10 toolz=0.11.1 tornado=6.1
```

`<my_dir>` can be a location in `/ccs` or `/gpfs/alpine` that is writable by you.

Hands on
How to create a custom environment

- Activate the environment

```
conda activate <my_path>/openpmd
```

- After activating, to make your created environment visible in JupyterLab, run

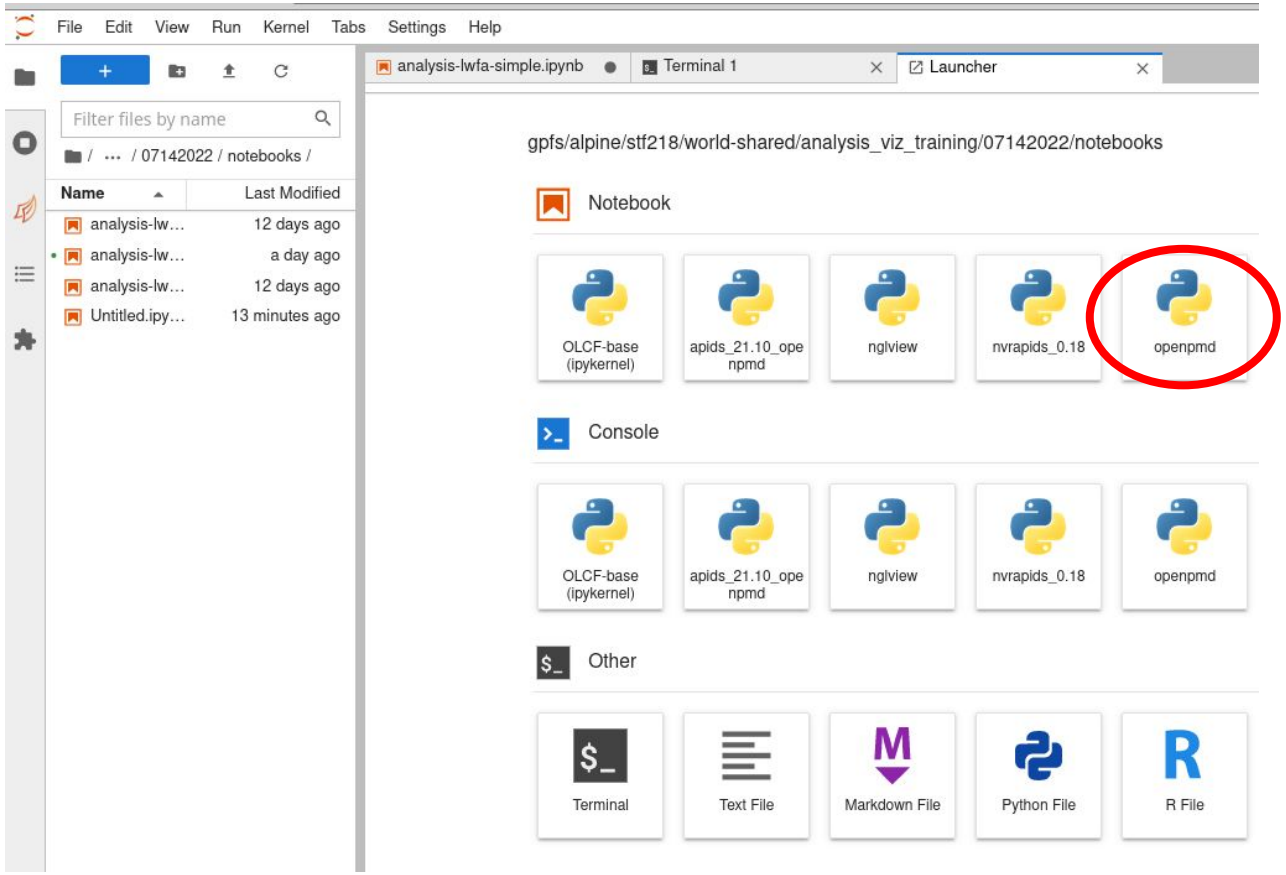
```
python -m ipykernel install --user --name openpmd --display-name openpmd
```

A kernelspec is created in your `/ccs/home/<YOUR_UID>/.local/share/jupyter/kernels` directory which JupyterLab reads to see which custom environments are available for it to use.

Hands on
How to create a custom environment

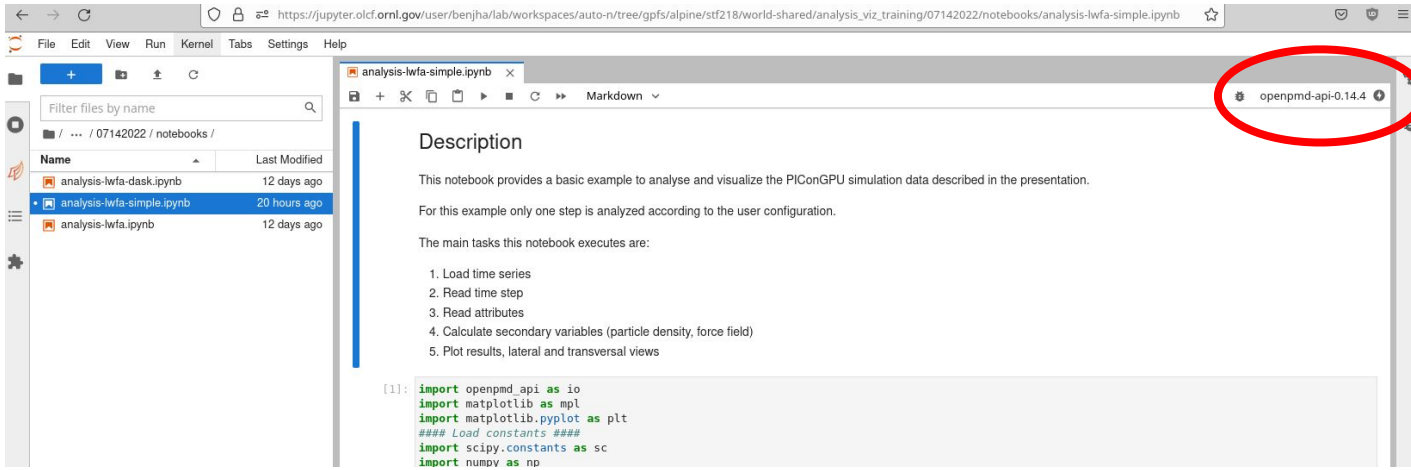
- When you refresh the page and look at the Launcher, you will see buttons labelled as **openpmd**

- Click on the **openpmd** environment

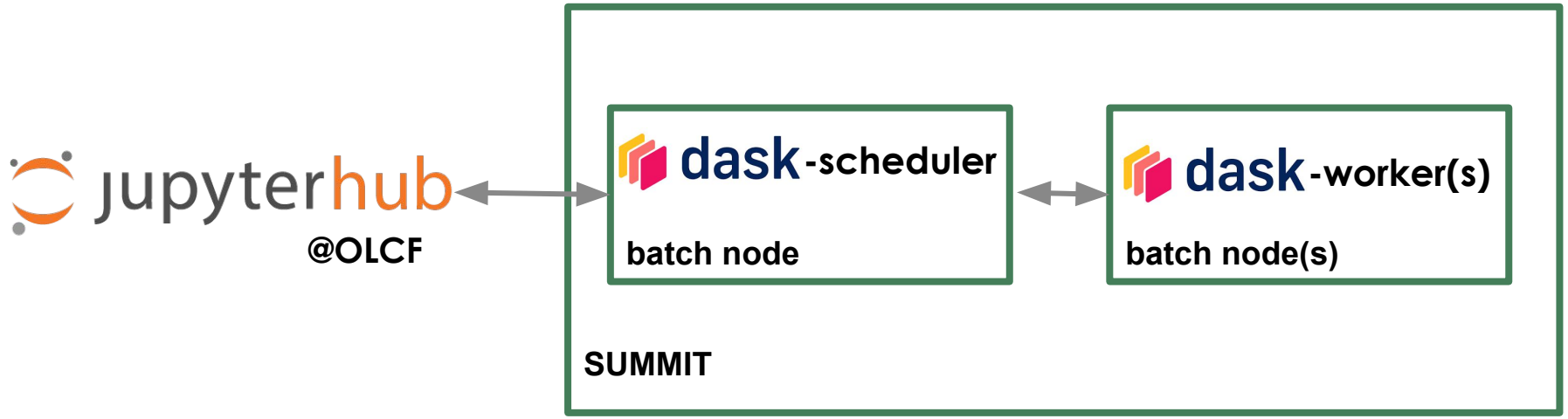


Hands on
How to create a custom environment

- Now copy `analysis-lwfa-simple.ipynb` to your desired location and open `analysis-lwfa-simple.ipynb` from its new location.
- Make sure the notebook is using the OpenPMD environment
- Follow the instructions of the notebook and generate some plots.



Hands on
How to scale analysis with dask using Summit



- Dask makes it easy to scale Python libraries such as NumPy, pandas, and scikit-learn
  - It provides a familiar user interface by mirroring the APIs of other libraries in the PyData ecosystem including: Pandas, Scikit-learn and NumPy.
- More info. https://docs.dask.org/en/stable/

Hands on

How to scale analysis with dask using Summit

Steps

1. Launch a dask cluster on Summit using the helper script, `launch_dask_cluster.lsf`, launch available in:

`/gpfs/alpine/world-shared/stf218/analysis_viz_training/07142022/dask-lsf-script`

2. Modify your python script (client) to connect and submit workloads to the dask cluster. Example provided:

   `analysis-lwfa-dask-summit.ipynb`

Hands on

How to scale analysis with dask using Summit

1. Launch a dask cluster on Summit using the helper script

- Open a regular terminal
- Copy `launch_dask_cluster.lsf` to your work directory
- Open `launch_dask_cluster.lsf`
- Specify you project id (ABC123) in line 3 and line 20
- Specify your email in line 14
- Save changes
- Submit the job

```
$bsub launch_dask_cluster.lsf
```

- Wait until you received an email from LSF confirming your job has started. This means your cluster is up and running
- Inspect the output file `cluster_1node_tcp_%J.out` for further details

Hands on

How to scale analysis with dask using Summit

2. Run, step by step `analysis-lwfa-dask-summit.ipynb` in jupyter@OLCF

- Configure which dataset to work with, the path where images are generated
- Specify the dask's scheduler file. This is used by the client to connect to the dask cluster running on Summit.
  - The scheduler file is automatically created when running the dask cluster.
  - The scheduler file path should be available in the output file `cluster_1node_tcp_%J.out` generated when running `launch_dask_cluster.lsf`

Thanks

Questions ?

help@olcf.ornl.gov

Slack Channel

https://join.slack.com/t/jupyterworkflowatolcf/shared_invite/zt-1c7q5rdyc-fb45Q6peHrgJ_w_qn6A1VA