Leveraging Jupyter on Maxwell HPC: joyful, visual and green computing

Yves Kemp, Arlena Mills-Marzoli, Neele Rahmlow, Sven Sternberger, Axel Wichmann, Frank Schlünzen

Abstract: Jupyter notebooks are great tools to mitigate the complexities of (heterogeneous) HPC systems, like the Maxwell cluster at DESY which serves the computational needs of all user facilities on campus, as well as a wide variety of applications ranging from plasma accelerators to quantum chemistry. We aim to expand the Jupyter ecosystem using frameworks like streamlit to provide application environments tailored to the needs of less experienced users, including realtime visualization capabilities. On this basis we are implementing for example Jupyter-driven remote desktops, user-friendly dashboards to compose or monitor batch-jobs, and visual frontends for data catalogues like SciCat. The implementations are accompanied by visual tools for resource utilization and CO2 footprints suitable both for users as well as admins.



The regular Workflow

0000000

Data Catalog Frontend

Jupyter

slurm

<u>ട്പ്പട</u>്ട

scicat

TCat

Streamlit

maxapp spawner



Job ID: Job name:	3546945 text_job	
Job state: Start: Elapsed:	 running a day ago 1 day, 2 hours, 17 minutes and 33 seconds 	Sep 28 2023 10:49:23 GMT+0200 (1695890963)) 94653)
Partition: #Nodes: Nodelist: shared nod Features: #GPUs:	exfel-wp72 3 max-exf[183-184,186] le(s): no [Gold-6140]Gold-6240]&768G 0	
Command: Std_err: Std out:	/gpfs/exfel/theory_group/user/dasarina/TEST_RUNS/DATA /gpfs/exfel/theory_group/user/dasarina/TEST_RUNS/DATA /gpfs/exfel/theory_group/user/dasarina/TEST_RUNS/DATA	A_NEW_1/DATA_MOTT_2/DATA_20/job_submit.sh NEW_1/DATA_MOTT_2/DATA_20/hostname-%N-3546945.en A_NEW_1/DATA_MOTT_2/DATA_20/hostname-%N-3546945.ou
Add logi	cal cores (HT)	
max-exf1183	: CPU 192 % memory: 32 %	
100 %	CPU usage (Ø: 192 %)	CPU memory (Ø: 32 %)



Maxwell HPC Cluster



The Maxwell High Performance Computing cluster has all the ingredients of a typical HPC platform like low latency, fast network (Infiniband), cluster file-systems and a scheduler (SLURM) to guarantee a rapid and fair distribution of workload on the compute resources. The workloads and requirements are however very diverse, requiring an *atomic* partitioning and very heterogeneous hardware, which makes it impossible harvesting the 4 Petaflops in a single batch job.

Most users will use the graphical login nodes (max-display) for graphical work and submission of batch-jobs. The login nodes are clustered and well equipped with GPGPUs and memory allowing for example CAD modeling. The nodes can be reached via ssh, a FastX client or a web-browser from anywhere in the internet, but are completely isolated from the DESY internal network for security reasons, and even root privileges are not sufficient to modify any files on the cluster filesystem (root squashing). The impact of compromised account will be minimal.



Generic utilities Cat start start Jupyter start scicat JupyterLab hdf5

JupyterHub as a proxy

JupyterHub can proxy almost every kind of application. Our batch-spawner implementation (maxapp-spawner) creates and submits the batch scripts, defining the proxied ports and application specific compute requirements. The proxied application runs as a batch job entirely in user space, which makes it very simple to allow access for example to storage, avoiding the nightmare of implementing GPFS extended ACLs in a generic webservice. User need to remember only a single access point – the jupyterhub – which controls and keeps track of applications.

Jupyterhuk





Virtual Desktops



Scientific Applications

Jupyter

There are a few python frameworks out in the wild making

dashboarding quite easy. Streamlit is one of those

frameworks coming with strong support for data frames and

Maxwell Dashboard

2

Maxwell

visualization.

0

O

0

Π

0

1

0

Streamlit

maxapp spawner



Jupyterhub modified as a dashboard service with a starting collection of preconfigured applications. The Hub uses a custom batch-spawner to launch (most of) the applications as jupyter-proxied batch jobs.

slurm

Applications like comsol, matlab need a graphical frontend. This can also be a local desktop using max-display as a proxy.

DESY network outer space







Data generated from experiments at Petra III, FLASH, Eu.XFEL are stored in GPFS. SciCat is a meta-data catalog implementation giving access to meta-data and snapshots uploaded to the data catalog. SciCat is designed to run on cloud infrastructure being deployed on kubernetes pods. The services naturally run in an unprivileged context, and have consequently no access to any experimental data (and k8s pods are not intended to locally embedded storage). A streamlit application - running in user context - can be used to provide full access to data, and allow simple media tools to view and manipulate for example images or HDF5 container; based on selections customized jupyter notebooks can be launched for data processing. Access to jupyter notebooks and SciCat are achieved through REST API tokens without any need for user actions.

on Digitalisation and Artificial Intelligence.

-jupyter

Rest APIs



Maxwell Job Edito

slurm token

Based on streamlit we are working on a generic dashboard which provides a quick overview on

batch-jobs, resource utilization and power consumption. It allows to create batch-scripts with zero

SLURM knowledge, use batch templates for commonly used applications and submit the job

through SLURMs REST API. An API token is generated automatically in the background, hiding all

from ~690kWh to ~590kWh leading to energy savings of at least 250MWh annually.

......

of the clusters complexity from (inexperienced) users.

Rest API

Some application do not run on the operating system of the Maxwell cluster, and some are much easier deployed as docker or singularity images. To facilitate the process we provide a custom VNC setup for operating systems like Ubuntu 22.04 or RedHat EL9. The batch-spawner launches the corresponding image as a regular batch-job, pulling the image from a Harbor container registry. The user can access the desktop through the jupyterhub in the web-browser, and the session can be secured with the users password. The setup lacks GPU acceleration, and websocket proxying is not yet implemented, but for most users in need of alternative operating systems the setup should be quite sufficient.





Artificial intelligence is heavily used on the cluster, and one of the applications in high demand is automated image segmentation and registration. A prototype has been implemented based on mlexchange. mlexchange is a machine learning pipeline using pre-trained models and interactive web-annotation tools for iterative improvement of the ML model. The setup uses a bluesky/tinder file catalog and plotly dashboards serving both files and application through a jupyterhub proxy'd application. The file catalog is created on the fly as part of the batch job. While the prototype works smoothly, it still needs some components to be implemented, and the need for a file catalog is not exactly matching our storage configuration.

ob ID: 3546945 ob name: text job
b state: • running
tart: a day ago (Sep 28 2023 10:49:23 GM I +0200 (1695890963)) lapsed: 1 day, 2 hours, 17 minutes and 33 seconds (94653)
artition: exfel-wp72 Nodes: 3

Monitor batch job

save