

Evolution of the HEPS Jupyter-based remote data analysis System

Zhibin Liu*, Qiulan Huang*, Haolai Tian, Yu Hu ,
Jingyan Shi , Ran Du , Hao Hu, Lu Wang , Fazhi Qi

Computing Center, IHEP

Outline

Introduction

Design & Implementation

Heterogeneous Resources Management and scheduling

User Interface

Automated deployment

Muti-User

Integrated Applications

Summary

Introduction

High Energy Photon Source (HEPS) is the fourth-generation synchrotron radiation source with the highest spectral brightness in the world

Big amount of data

- 200TB of original experimental data every day
- the peak value can reach 500TB per day

Demand diversity

- Various algorithms and software: self-development and commercial purchase
- Different Operating systems: Linux and Windows
- Different resource types: CPU and GPU



Goals

WEB-based data analysis

- Not limited by local resources

Integration with IHEP resources

- Access software, user/experiments data

Muti-User

- Integrated IHEP unified certification
- Safe and isolated environment

Smart and flexible scheduling strategy

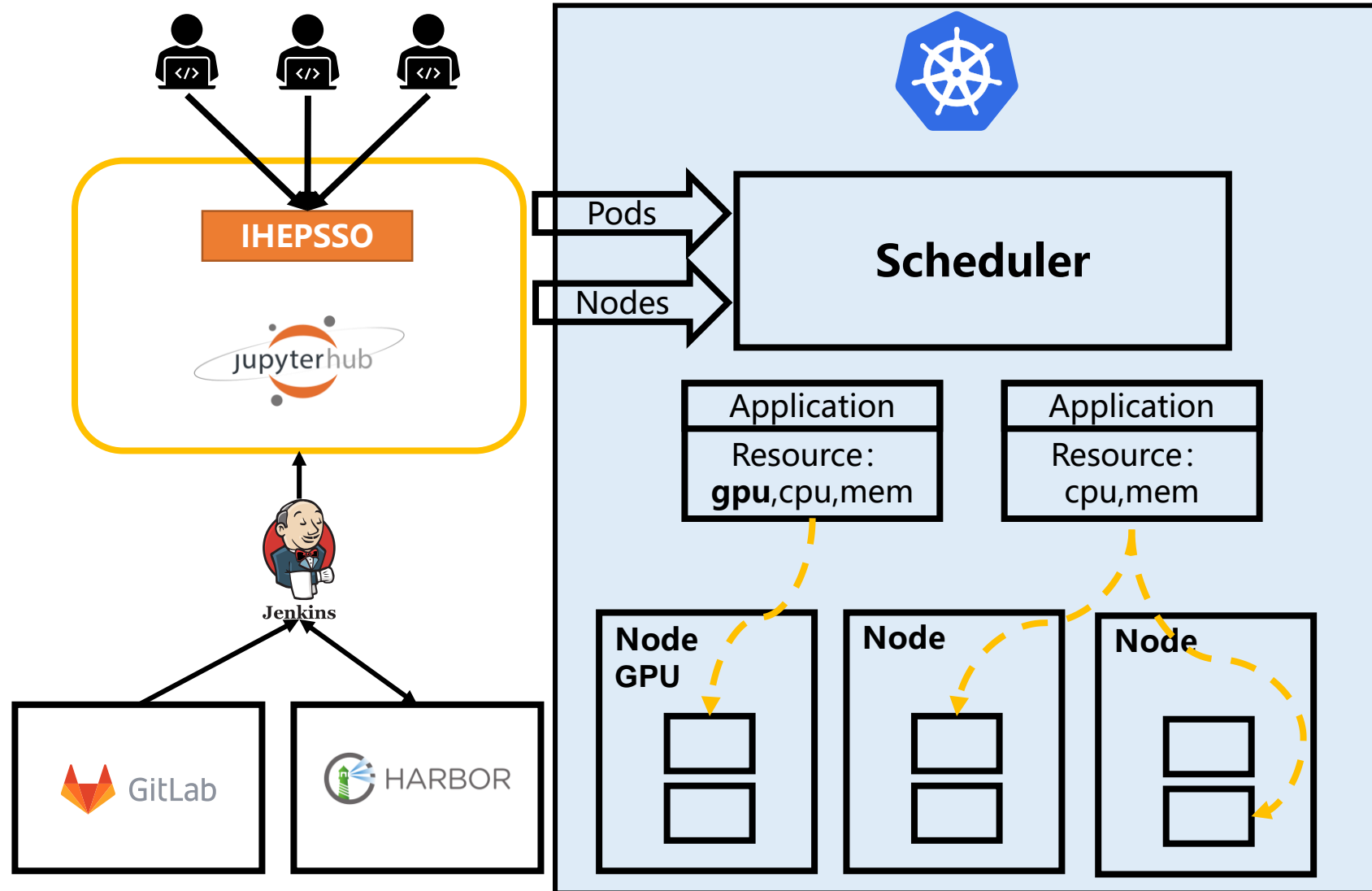
- Improve cluster resource utilization

Easy to deployment

- Automated deployment, streamline the deployment process



Architecture



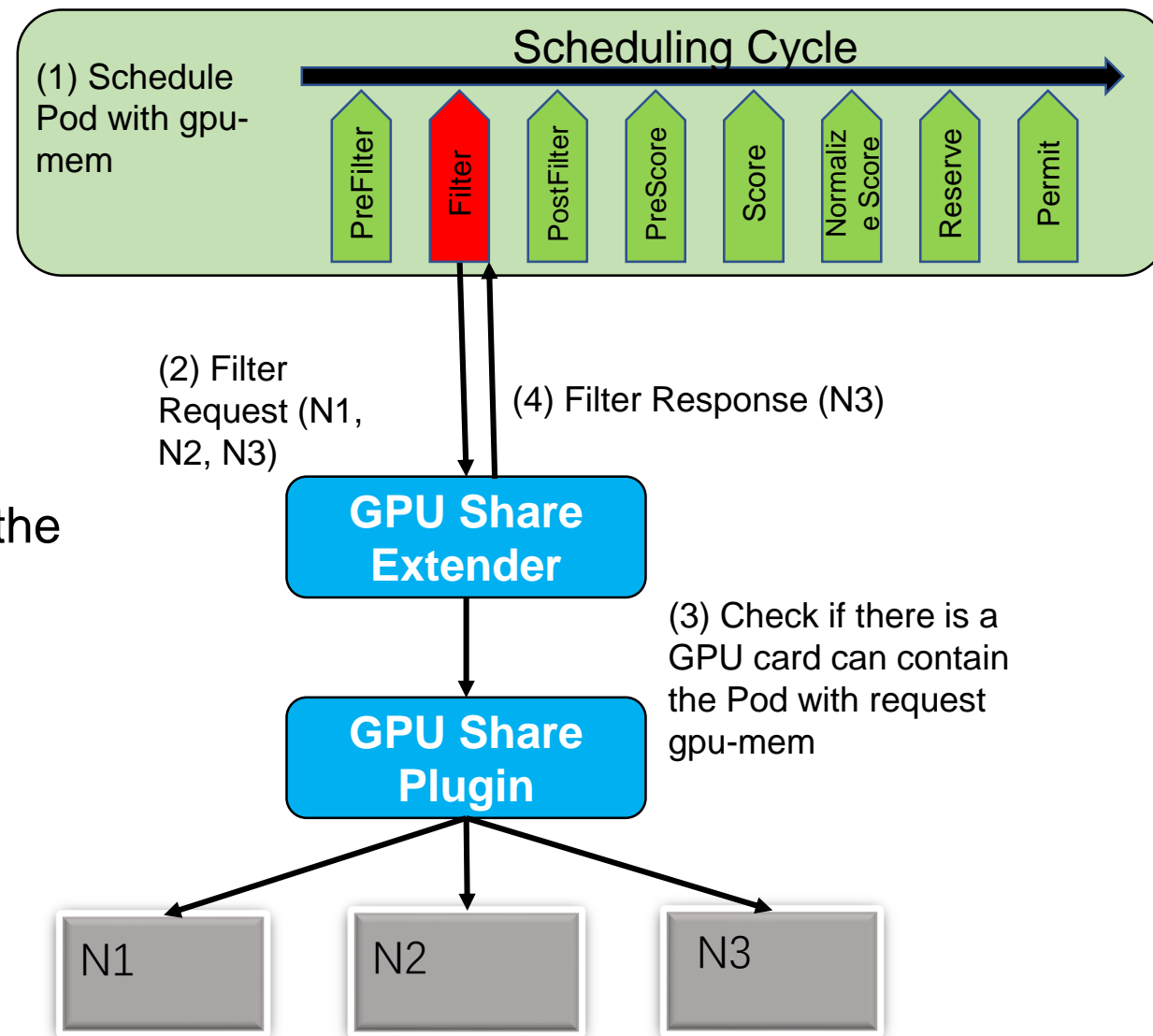
Heterogeneous Resources Management and scheduling

GPU Share Extender

- Collect several metrics like number of GPU cards and memory of GPU nodes
- Detects the GPU allocation result of all the GPU cards

GPU Share Device Plugin

- Responsible for the GPU allocation



Heterogeneous Resources Management and scheduling

Filtering

- × Disk Pressure
- × Memory Pressure

Score

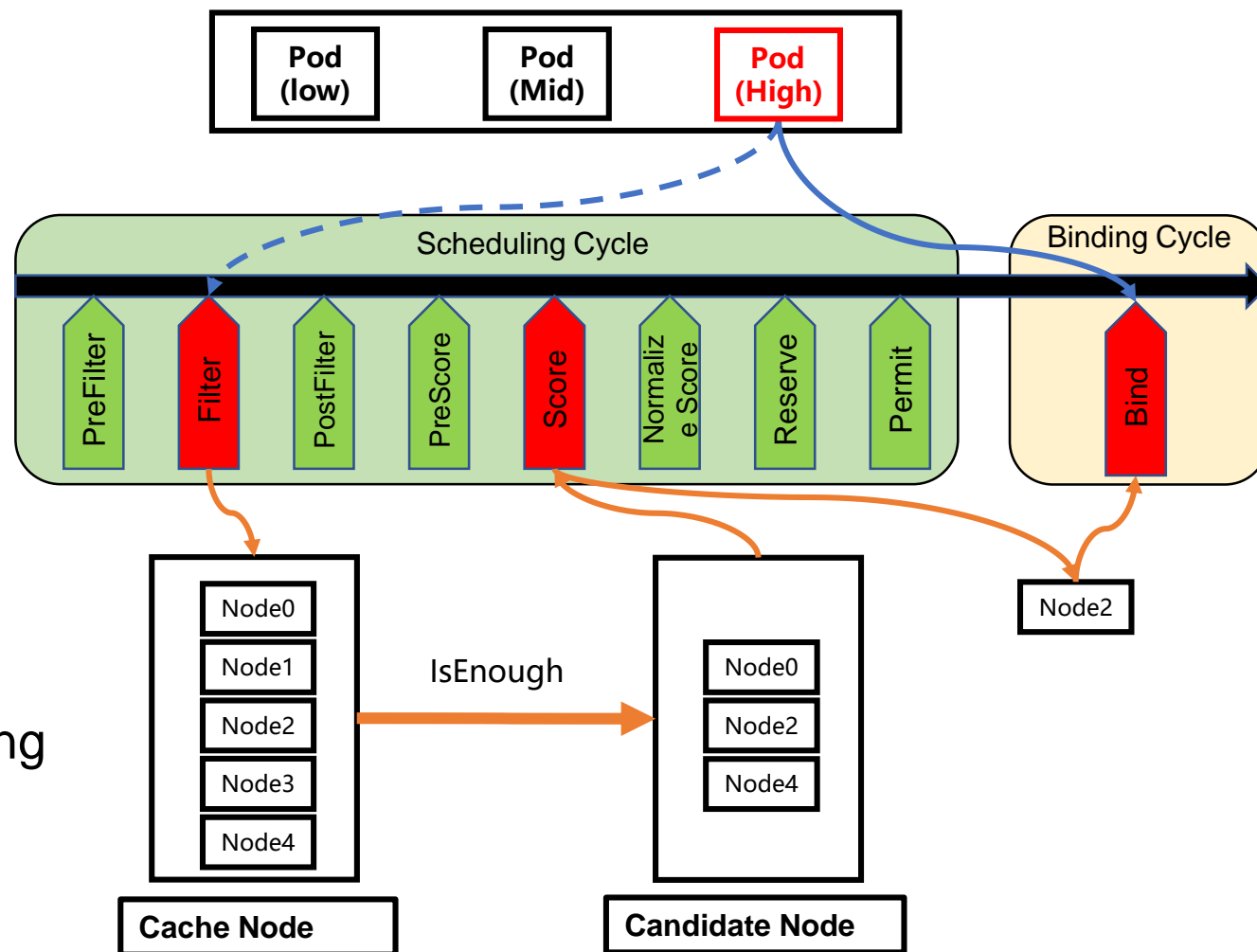
- Pod Terminate, Pod Nominate
- Pod failed times

Bind

The Node and Pod selected in the scoring stage will be bound

Qos(Quality of service)

Pod Priority Queue



User Interface

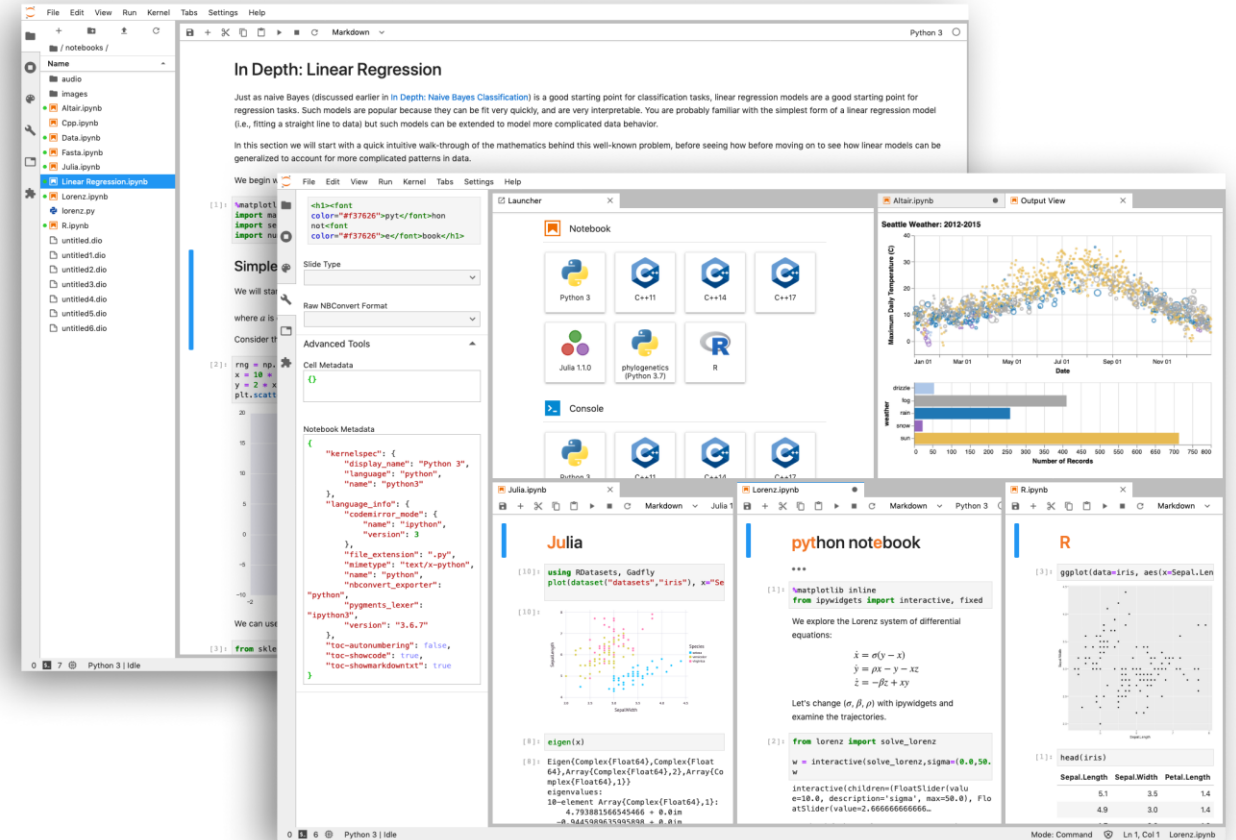
Jupyter are designed to enable interactive data analysis from personal computers to the cloud.

Jupyterhub

- Multi-User Management
- Server Options
 - CT 3D Reconstruction
 - Deep Learning
 - Spark

Jupyterlab

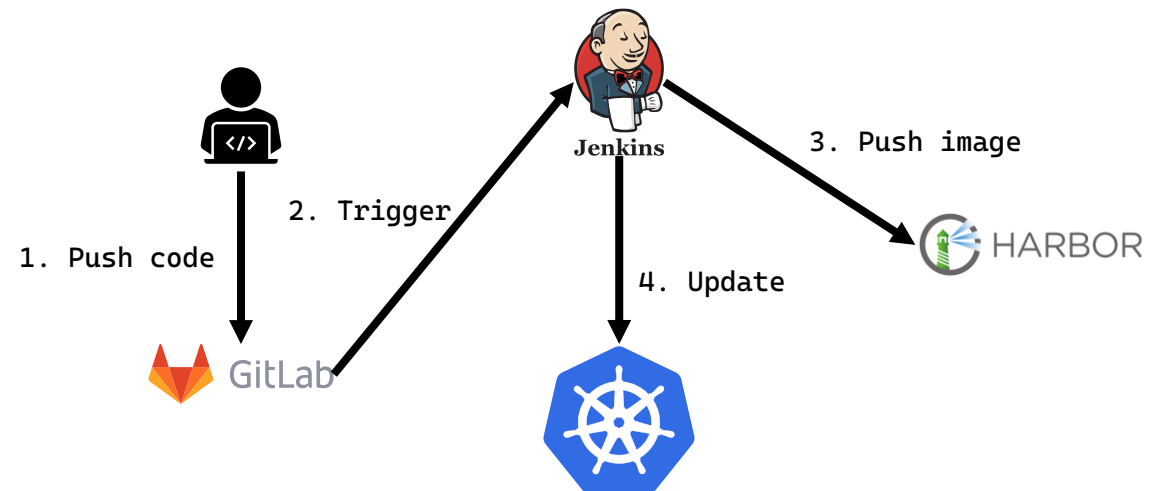
- Web-base
- Code and data
- Notebook
- Terminal



Automated deployment

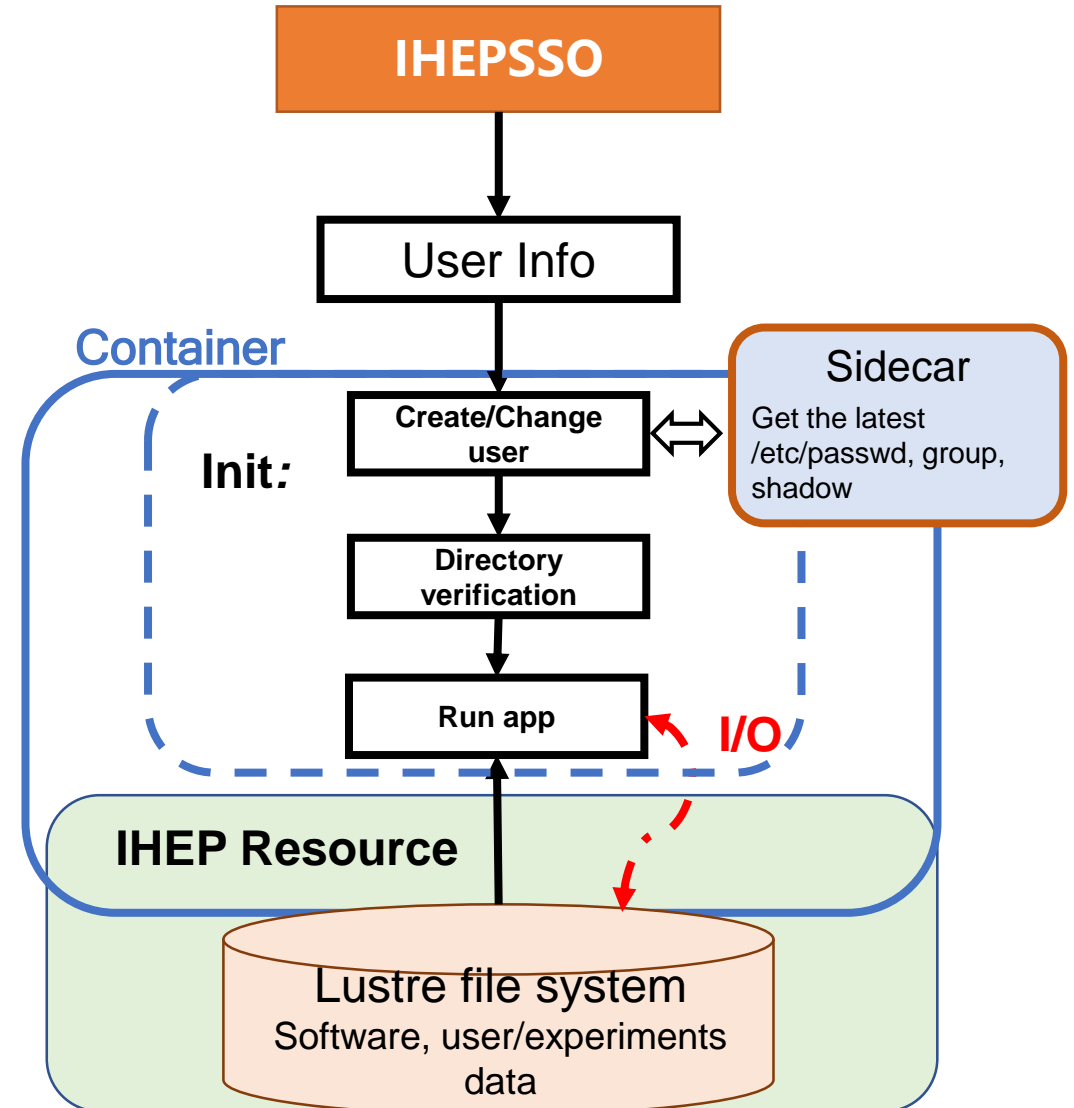
- In order to reduce the workload of new application integration and deployment, we adopt CI/CD way to accelerate build and deployment
1. Push the code to repository
 2. Automatically test and rebuild new applications
 3. Push new images to HARBOR
 4. Pull new images from the repository, and update the configuration file
 5. The kubernetes cluster will deploy new updates

```
Dockerfile
FROM dockerhub.ihep.ac.cn
RUN conda install -y tomopy
USER $NB_USER
WORKDIR /
ENTRYPOINT [
"/usr/bin/tini", "--" ]
CMD [ "start.sh" ]
```



Multi-User Management

- **Multi-User.** It provides role-based fine-grained authentication in a unified way
- **Unified authentication:** Single sign-on (SSO) is supported to achieve unified authentication of user identity.
- **Authorization module:** Integration with IHEP resources. Get the latest user/group information and authorized by existing distributed storage systems
 - Create a new user through the user information returned by SSO
 - Verify user directory
 - Run the application in the user home directory



Integrated Application

scientific computing applications

- CT 3D Reconstruction(DONE)
- HXMT data analysis(Undergoing)
- Deep Learning(DONE)
- Spark(DONE)

Example:

- **3D Reconstruction application** integrates an open-source 3D tomographic reconstruction package, TomoPy
- Provides a Web-based graphic user interface powered by Jupyter Widget

Server Options

☒ **CT 3D reconstruction**
CT 3D reconstruction service based on tomoPy.

☐ **HXMT data analysis**
HXMT interactive data analysis service.

☐ **Deep Learning**
Deep Learning envirmnt includes popula

- tensorflow and keras machine le
- dask, pandas, numexpr, matplotl
- cython, patsy, statsmodel, cloud
- beautifulsoup, protobuf, xird, bo
- ipywidgets and ipympl for intera
- Facets for visualizing machine le
- git, vi (actually vim-tiny), nano (a
- conda: cross-platform, language

☐ **Spark**
Spark envirmnt includes Python, R, and

- Apache Spark with Hadoop binai
- IRKernel to support R code in Ju
- Apache Toree and spylon-kernel
- ggplot2, sparklyr, and rcurl pack

jupyter

CT GUI demo

Reconstruction Center of rotation Data view

Input

☒ Projections
☐ Sinograms

☒ Region(y-step)

21

☐ Do flat-fitted correction

Path: /opt/CT/ZY-2/

Show Slices

Flat - fieldcorrection

Method: Average

Darks: /opt/CT/ZY-2/

Flats: /opt/CT/ZY-2/

Reconstruction

Method: fbp

Center: 1030

Reconstruct

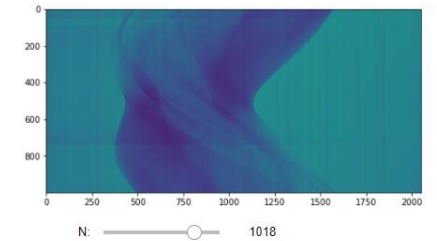
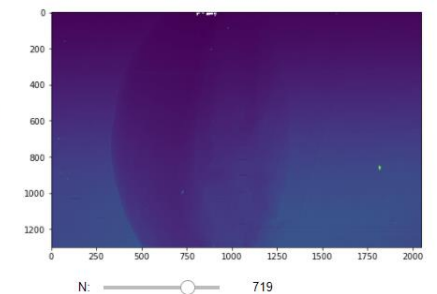
Output

path: Type the output path

Save

Log

```
2020-10-26 11:20:10,684 - [INFO] Reconstruction completed!
2020-10-26 11:18:45,367 - [INFO] Reconstruction processin
g!
2020-10-26 11:18:45,363 - [INFO] Reconstruction start!
```



Example of 3D reconstruction application

Summary

We design and implement of a Jupyter-based remote data analysis system.

- ✓ Optimized the default scheduler of Kubernetes
- ✓ Provided WEB-based interactive data analysis service
- ✓ Streamlining and automating the development, test and production process
- ✓ Developed IHEPSSO authentication plugin

Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC) under Contracts No. 11875283.