# Kubernetes container orchestration as a framework for flexible and effective scientific data analysis

Anton Teslyuk, Sergey Bobkov, Viacheslav Ilyin,
Alexander Novikov, Alexey Poyda, Vasily Velikhov

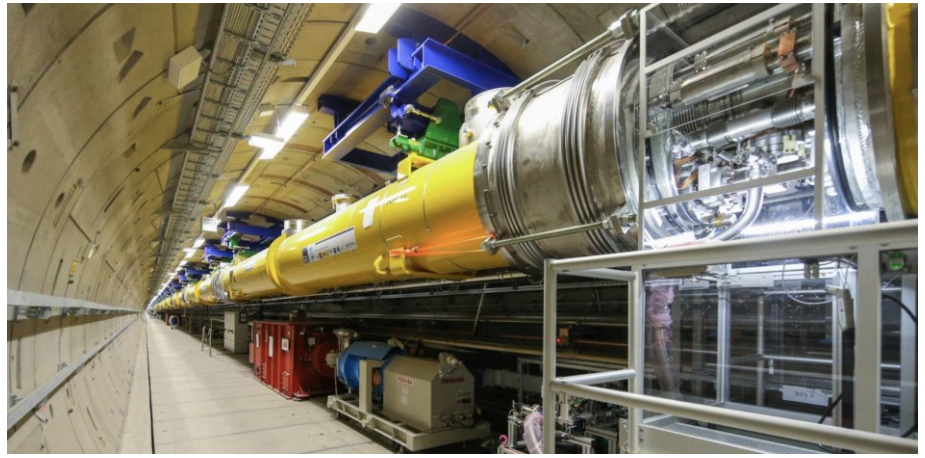
NRC Kurchatov Institute

IVANNIKOV ISP RAS OPEN CONFERENCE

5-6 December 2019

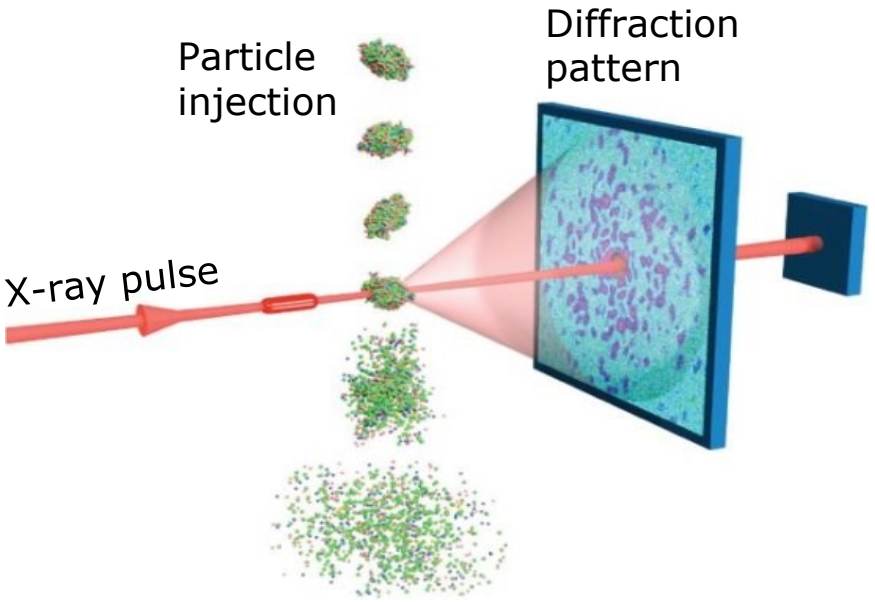
# European XFEL

- X-Ray Free-Electron Laser - mega science research facility
  - High brilliance ($10^9$ times more than conventional X-ray source)
  - High frequency: up to 27000 flashes per second
  - Wavelength range: 0.05-4.7 nm
  - Short pulses: less than 100 fs
- Construction start – Jan 2009
- First experiments – Sep 2017

# SPI Experiments

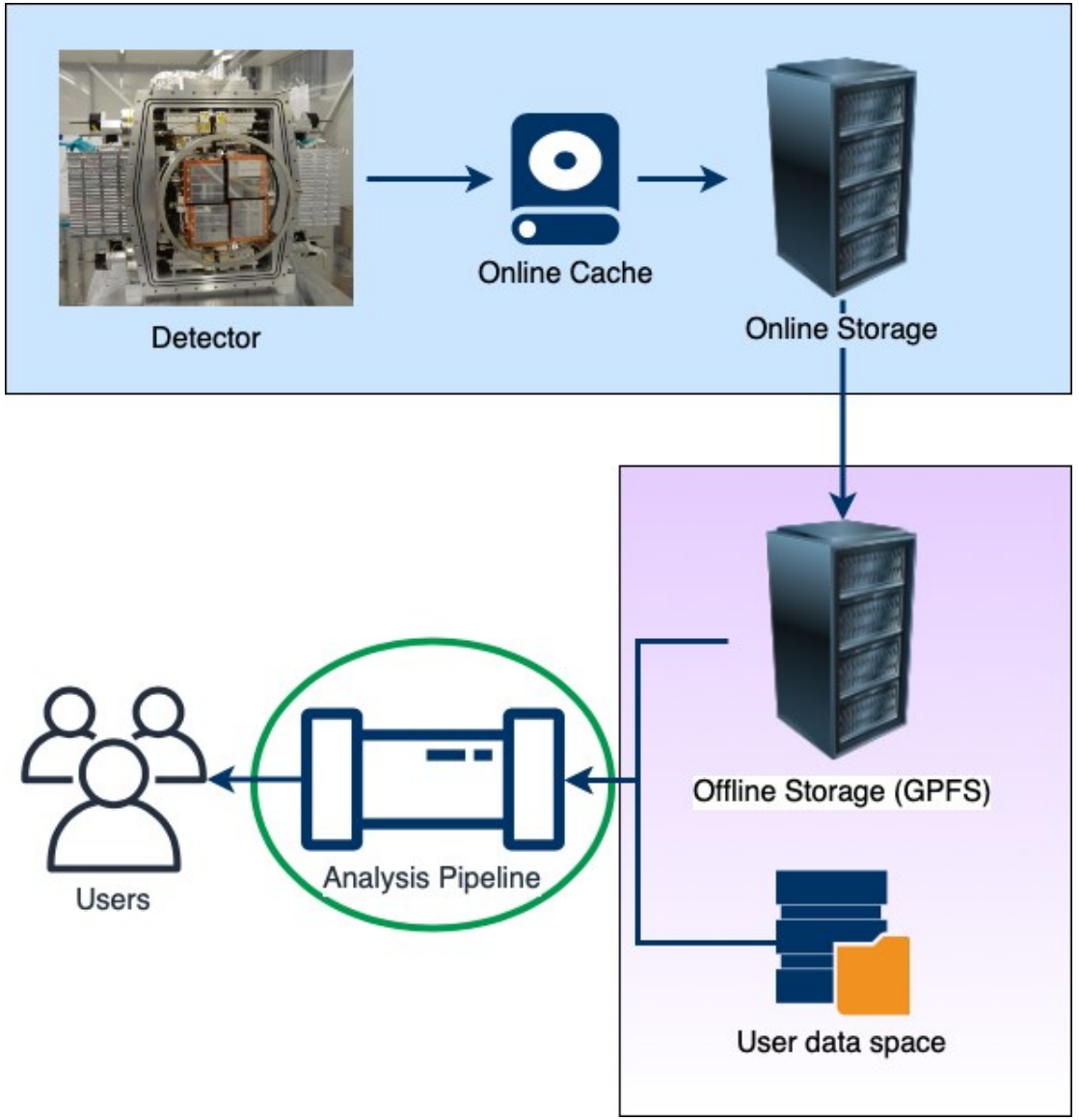
- The goal: Molecule structure at atomic level (1A)
- Big data:
  - 120 Tb per experiment (Dec 2017)
  - 360 Tb per experiment (May 2019)
  - expected to be increased **100x** times!
- Experiments evolve rapidly
- Data Analysis is also under intensive development:
  - Algorithms
  - Software
  - IT services




*Gaffney K. J. & Chapman H. N.// *Science*, 2007.

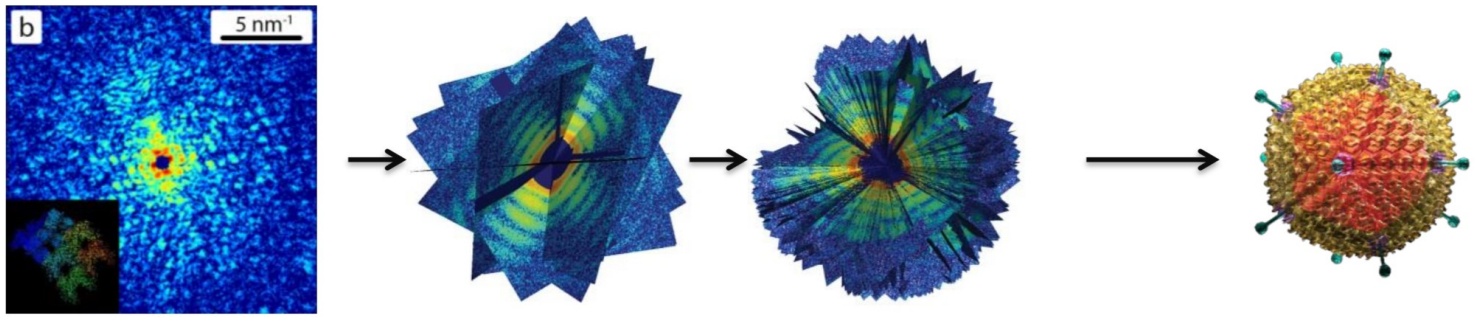
# The Goal of the Project

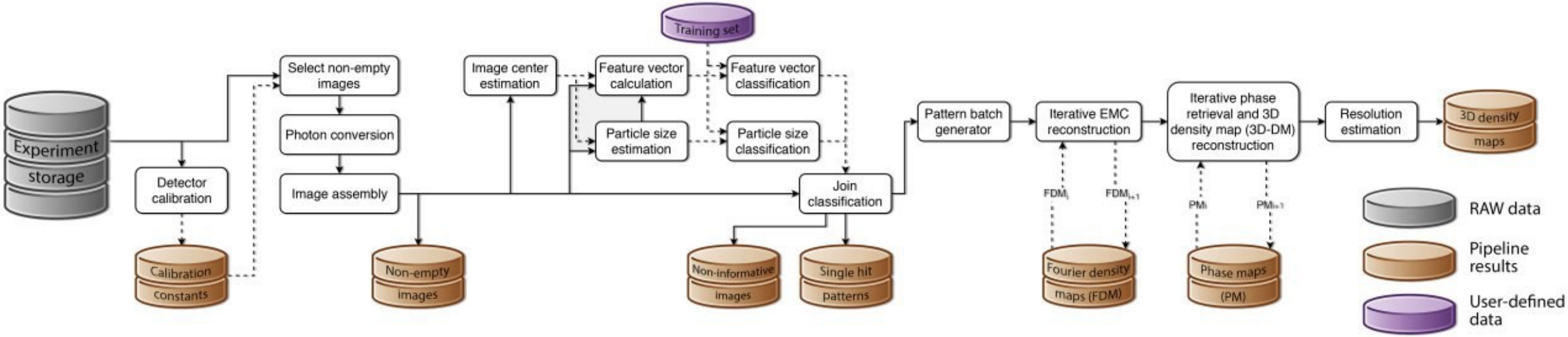
- Software Pipeline for automated data processing
- From diffraction patterns to 3D structure in near real-time
- Core Ideas:
  - Integration of software packages for various stages of data analysis in analysis pipeline
  - Simple configuration and deployment
  - Scalability
  - Extensibility, modular architecture
  - Various workflows



# XFEL data analysis scheme

Briefly



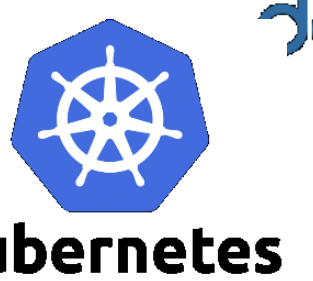




A little bit more detailed

# Realization Strategy

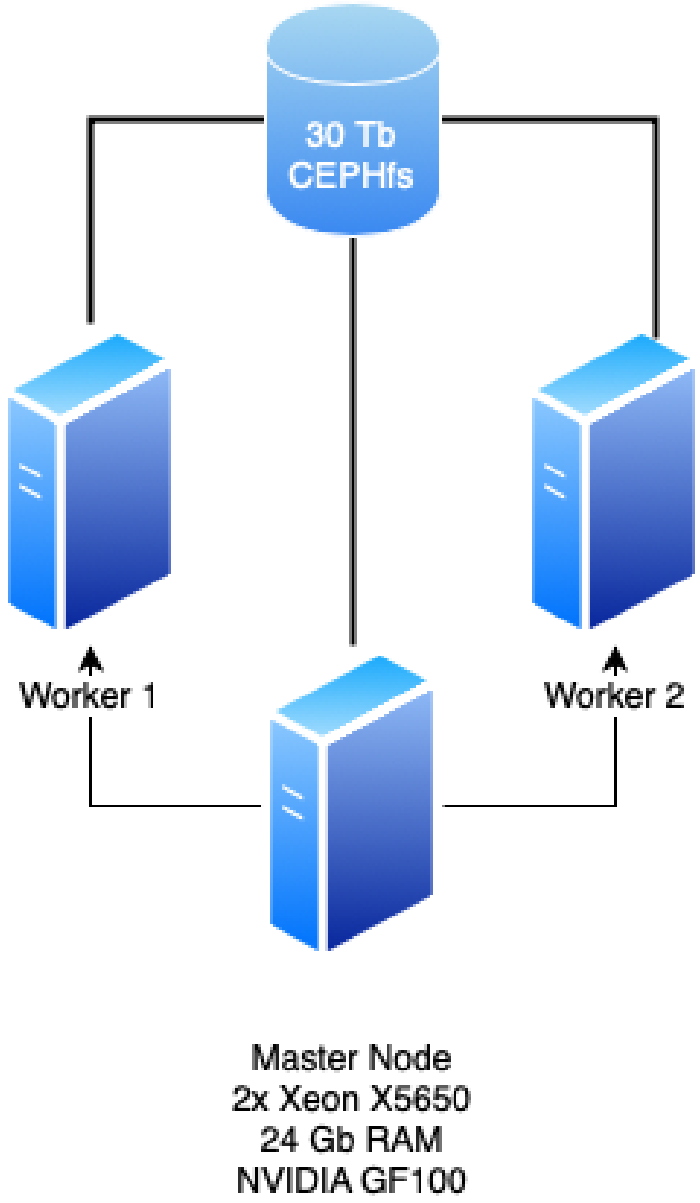
- Container technology for easy software deployment
- Microservices for individual stages of analysis
- Container orchestration for scalability and management
- Shared network filesystem for data I/O





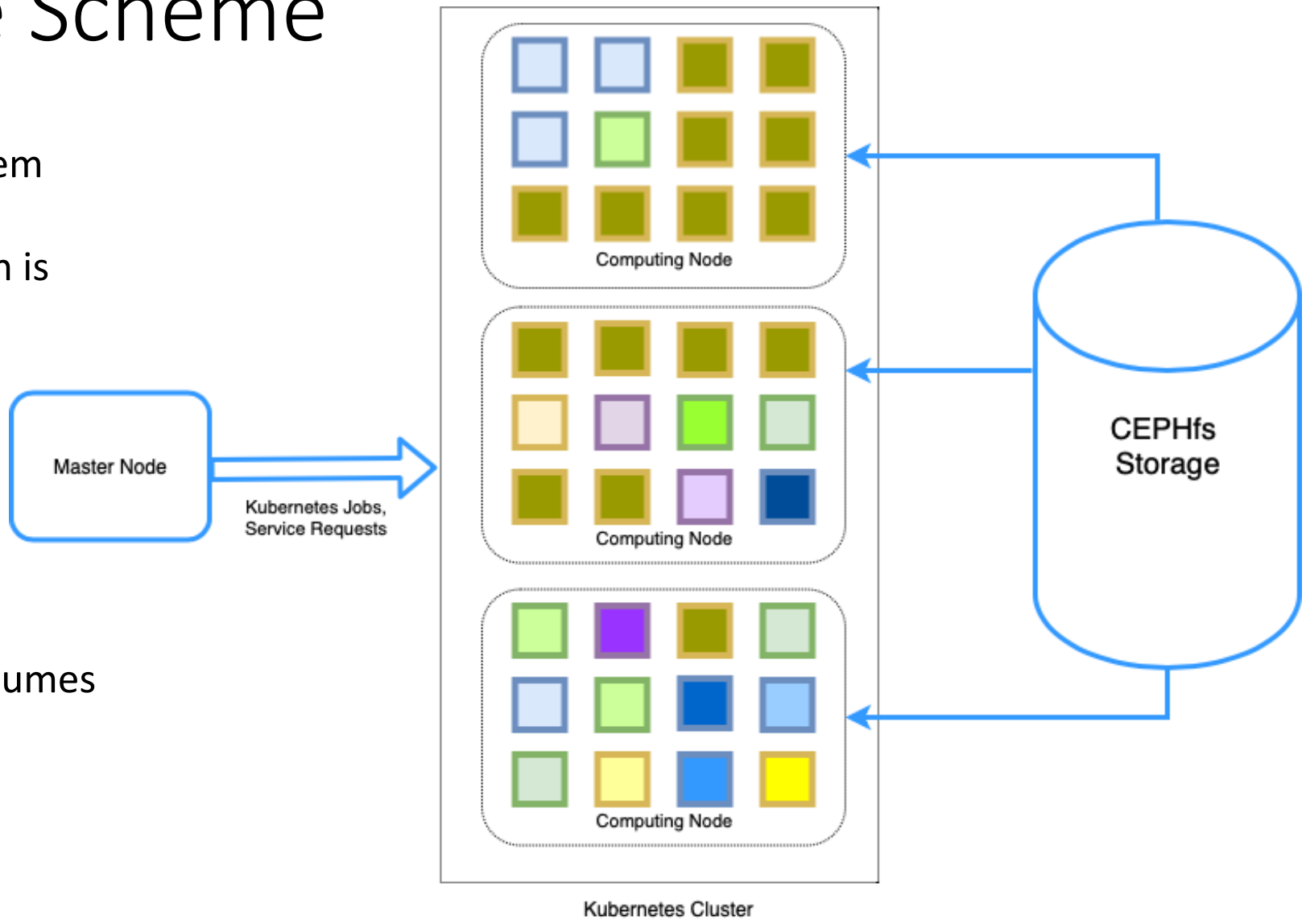

# Testbed

- Dedicated K8s cluster (version v1.15.3) with three nodes
- Dedicated CEPHfs storage
- 1Gbps interconnect
- NVIDIA M2050 GPU cards



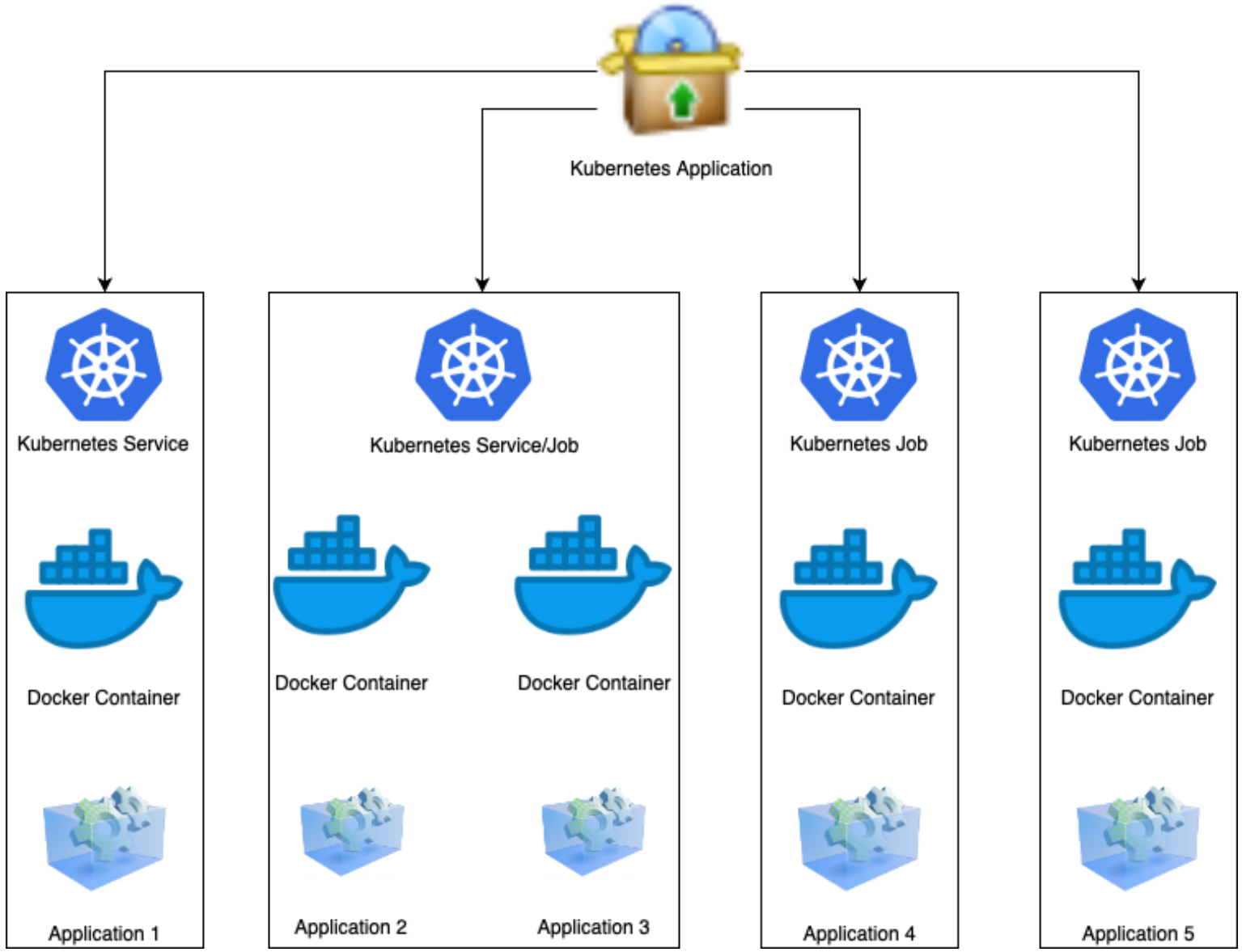
# Data Exchange Scheme

- Data is stored in a shared filesystem (GPFS, Lustre, CEPH) in HDF5 files
- K8s based container orchestration is used for:
  - containers deployment
  - load balancing
  - internal and external communications
  - services monitoring and management
- Native K8s support for CEPHfs volumes



# Technological Layers

- Software Platform Level
- Service/Job Level
- Container Level
- Application level



# Container Level

- Information how to build and install application
- Dockerfile syntax
- Result: application is ready to be used inside the container
- Users can use it directly with Docker!

```
FROM ubuntu:18.04
WORKDIR /root
RUN apt update && apt upgrade -y
RUN apt install -y cmake libtiff5-dev libfftw3-dev gsl-bin
RUN git clone https://github.com/FXIhub/libspimage.git
RUN mkdir -p libspimage/build
WORKDIR libspimage/build
RUN cmake -DCMAKE_VERBOSE_MAKEFILE=ON -DBUILD_LIBRARY=ON -D
RUN make && make install
RUN mkdir /opt/xfel
WORKDIR /opt/xfel
COPY phase.py .
```

# Kubernetes Services/Jobs Level

- Description of how to run the Application:
  - location of container images for job applications
  - location of volumes with the data
  - parallelization patterns
- YAML syntax
- Result: application is connected to data and is parallelized inside K8s cluster

```yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: phaser-sample
spec:
  template:
    spec:
      containers:
      - name: phaser-sample
        image: wn75:5000/phaser
        command: ["/usr/bin/python", "./phase.py",
        volumeMounts:
          - mountPath: "/ceph"
            name: cephfs
      volumes:
      - name: cephfs
        cephfs:
          monitors:
          - ceph55.sandbox.g3.computing.kiae.ru
          user: cephfs
          secretRef:
            name: ceph-secret
          readOnly: false
      restartPolicy: Never
  backoffLimit: 4
```

# Platform Level

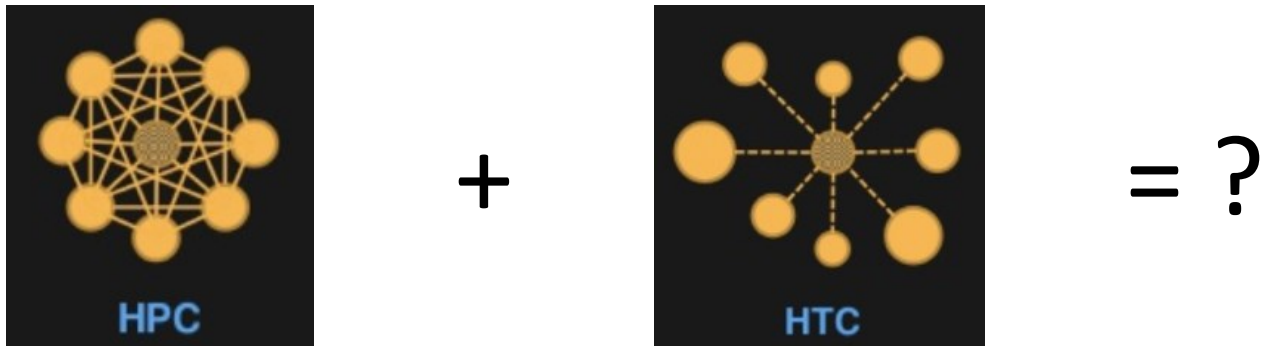
- Data processing platform as a set of Kubernes objects:
  - Services/Jobs
  - Data Volumes (CEPHfs)
  - Configuration Parameters
  - Set of users and user roles, access patterns
- Helm Templates Syntax: charts, releases, deployments
- Available as a package from repository, can be installed in a simple manener:

```
$ helm repo add stable https://bio1.grid.kiae.ru/repo/xfel
$ helm repo update
$ helm install stable/xfel_pipeline --generate-name
```

# Use Cases: Orientations Determination

- Dragonfly
  - EMC algorithm for orientations reconstruction
  - High quality code
  - MPI
  - GUI interface

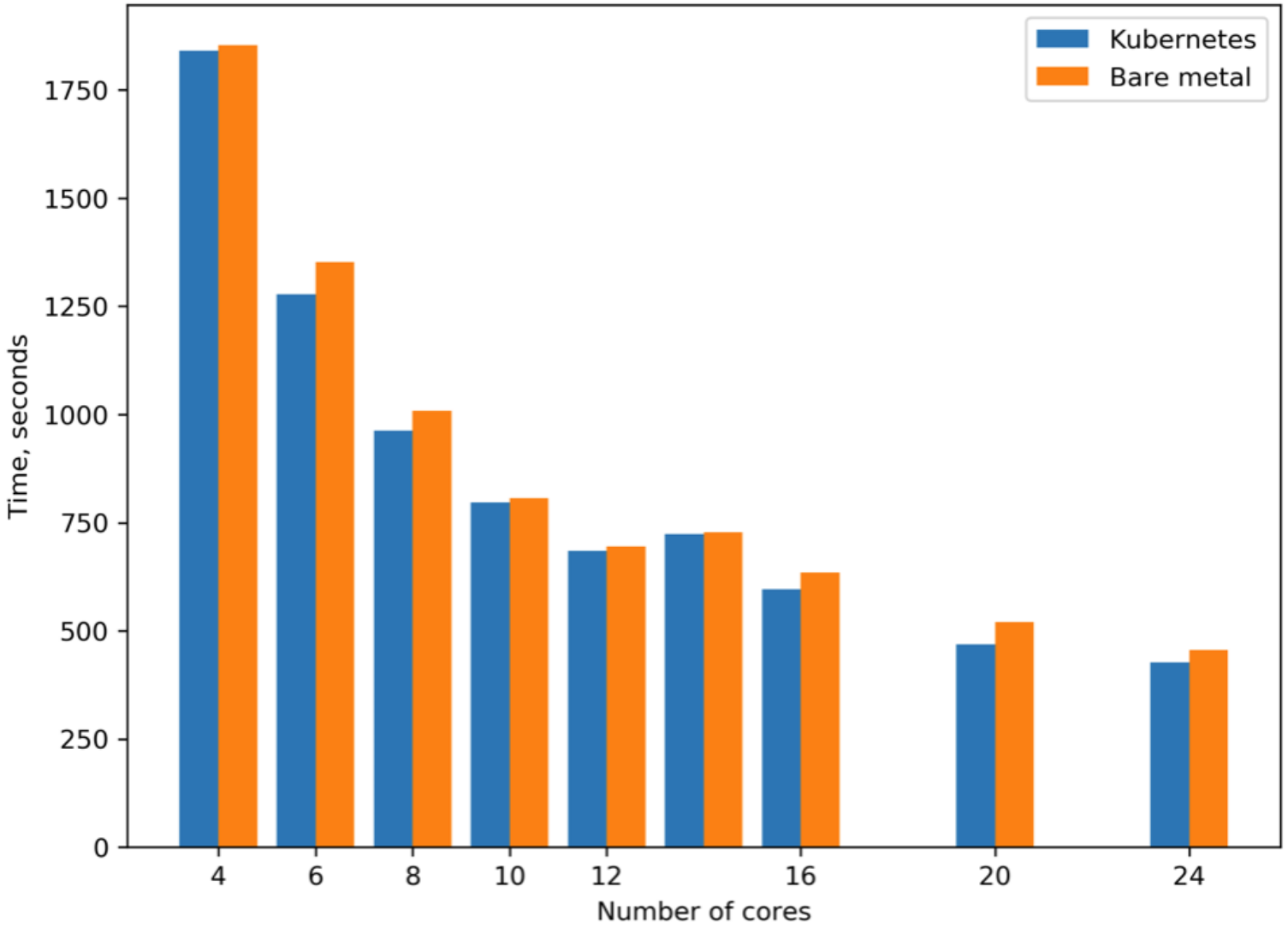
- It is the bright case where HPC application meets HTC (Cloud)!



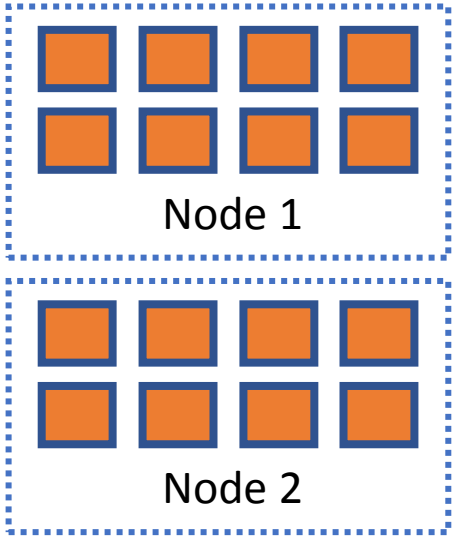
# HPC vs HTC

- Different focus, history, architecture, ecosystem
  - HPC – parallel computing. Intensive communications between nodes
  - HTC – data and services centric. Loosely coupled services
- Possible scenarios of combined usage
  - application code refactoring
  - run HTC workloads in HPC systems (Singularity, Shifter)
  - virtualize HPC infrastructure in HTC systems
  - maintain separate infrastructures

# Dragonfly scaling benchmarks




- Kubernetes jobs vs bare metal Centos 6 installation
- Kubernetes is approx. 4% faster than bare metal!



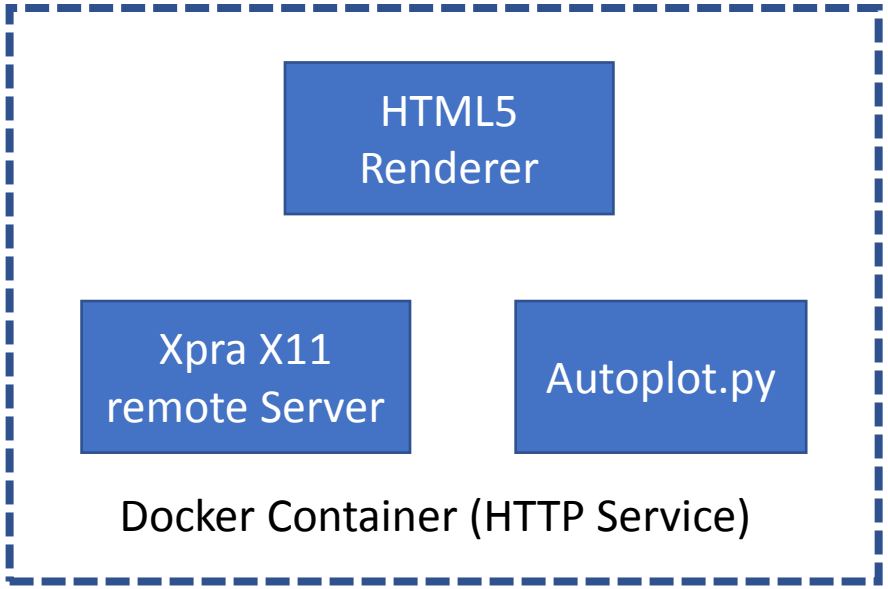
# Components: Phase retrieval

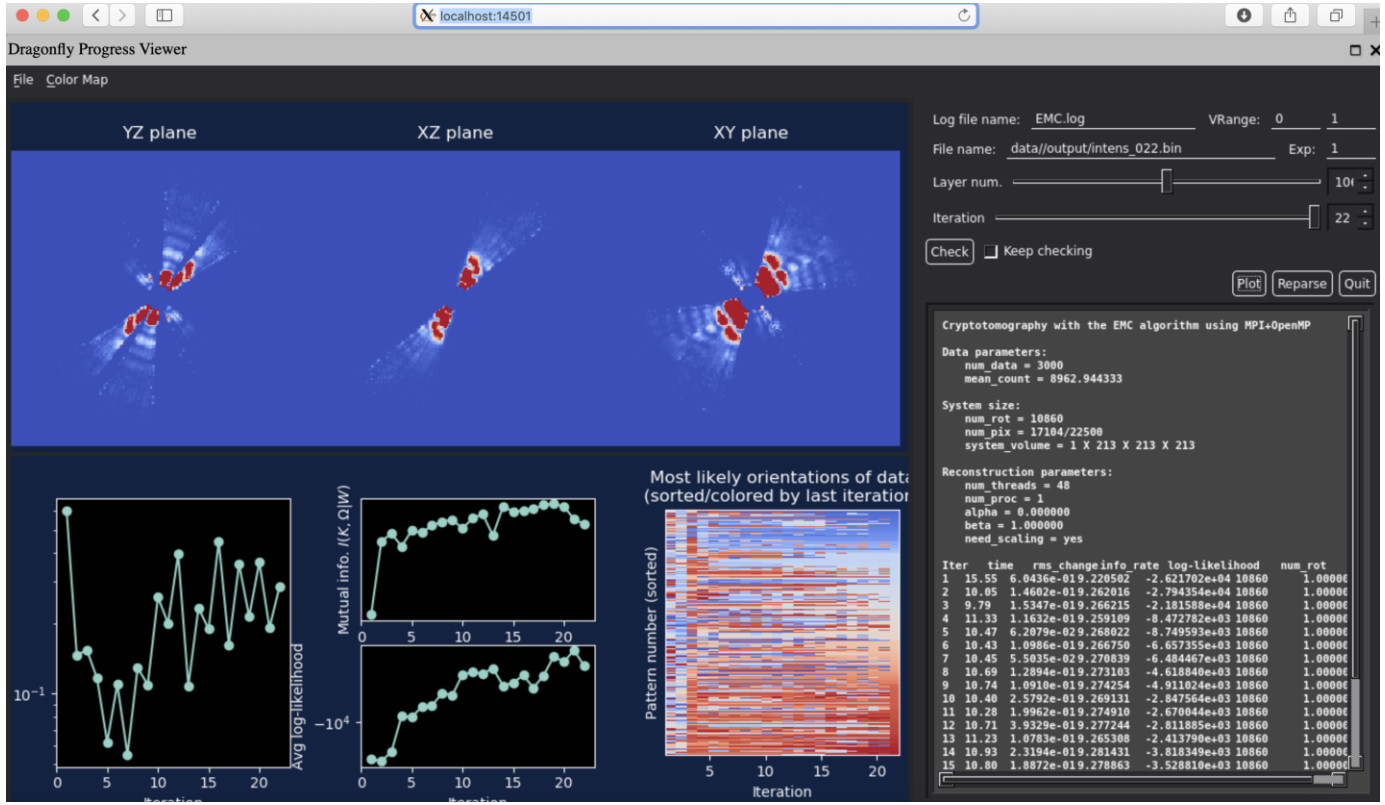
- Orientation Determination
  - Libspsim package
  - Python wrapper to parse input/ouput, compute
- At Docker level: Ubuntu 18 based image, CUDA support
- At K8s level: Works as K8s Job with CEPH filesystem volumes
- Use JSON format as Input/Output
- Use HDF5 to store output

```
{
        "srand": 1,
        "data": "/ceph/teslyuk/xfel/dragonfly/Dragonfly/recon_0002/data/output/intens_010.bin",
        "data_dim": "3d",
        "number_of_outputs_images": 2,
        "number_of_outputs_scores": 10,
        "support_algo": "static",
        "support_size": 30,
        "algo": [
                { "name": "hio", "number_of_iterations": 500, "beta_init": 0.9, "beta_final": 0.9 },
                { "name": "er", "number_of_iterations": 100}
        ],
        "out_file": "/ceph/teslyuk/xfel/dragonfly/Dragonfly/recon_0002/phased_output.h5"
}
```

# GUI applications as a web K8s service

- autoplot.py as a HTTP service
- realtime EMC monitoring from browser



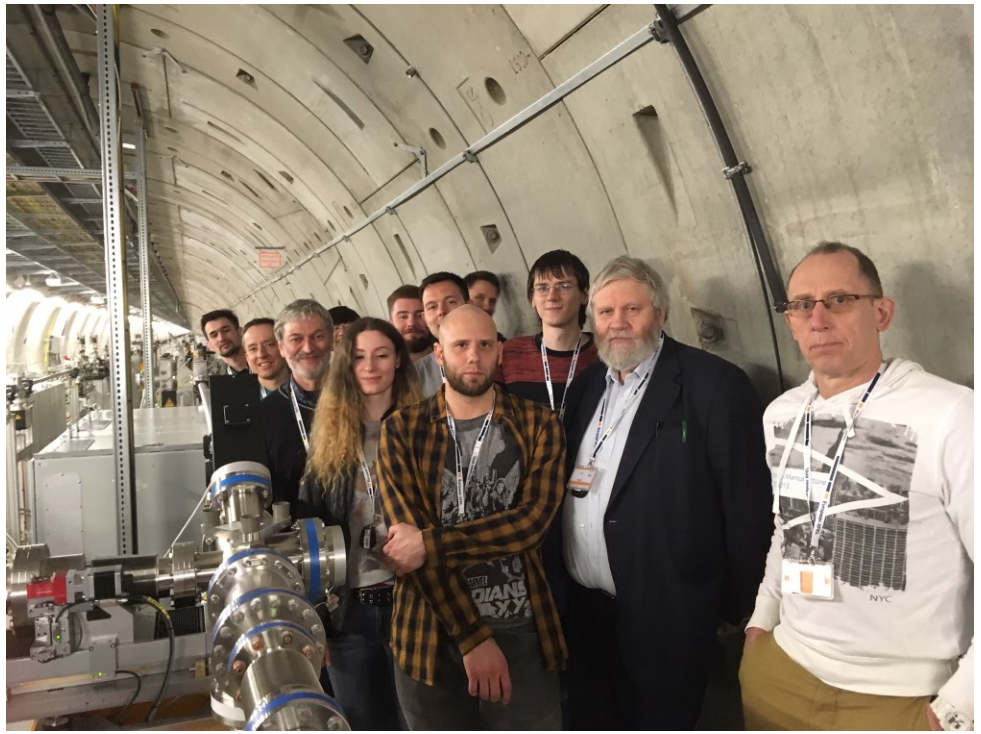



# Summary

- Docker and Kubernetes is a suitable platform to build data analysis pipelines
- K8s infrastructure allows various scenarios of software usage:
  - Data parallel applications
  - MPI applications
  - SMP/Cuda  applications
  - GUI applications as  web services
- From XFEL data analysis testbed to wider applications

# Acknowledgements

Joined Team from
KI and DESY



Presented results are supported by the Helmholtz Associations Initiative and Networking Fund and the Russian Science Foundation (Project No. 18-41-06001).