Finanziato dall'Unione europea
NextGenerationEU

Ministero dell'Università e della Ricerca

Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA

ICSC
Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

ICSC
Centro Nazionale di Ricerca in HPC,
Big Data and Quantum Computing

Benchmarking distributed-interactive HEP analysis workflows on the new Italian National Centre analysis infrastructure

Sabella G., Cirotto F, D'Onofrio A., Gravili F.G., Loffredo S, Rossi E., Spiga D., Spisso B., Tedeschi T.

110° Congresso Nazionale Società Italiana Fisici,
11 settembre 2024, Bologna

ICSC
Centro Nazionale HPC,
Big Data e Quantum Computing

ICSC: National Centre on HPC, Big Data and Quantum Computing project

The rapid growth of data from scientific, industrial, and institutional sources will pose challenges in deriving social and economic value. Supercomputing, AI, numerical simulations, high-performance analytics, and big data management will be vital for tackling societal issues and fostering sustainable innovation.

To implement Italy's National Strategy for HPC and Big Data, key steps include developing advanced supercomputing and cloud infrastructure, establishing centers of excellence, fostering collaboration between academia and industry, training experts, promoting innovation for SMEs, sharing expertise, evaluating societal impacts, and addressing ethical implications through dedicated monitoring.

The CN aims to establish a national digital infrastructure for research and innovation by evolving existing HPC, HTC, and Big Data systems into a cloud datalake model. This infrastructure will offer accessible cloud interfaces, supported by expert teams, and foster a globally competitive ecosystem through public-private partnerships, driving advancements in computing technologies.

ICSC
Centro Nazionale HPC,
Big Data e Quantum Computing

ICSC: National Centre on HPC, Big Data and Quantum Computing project

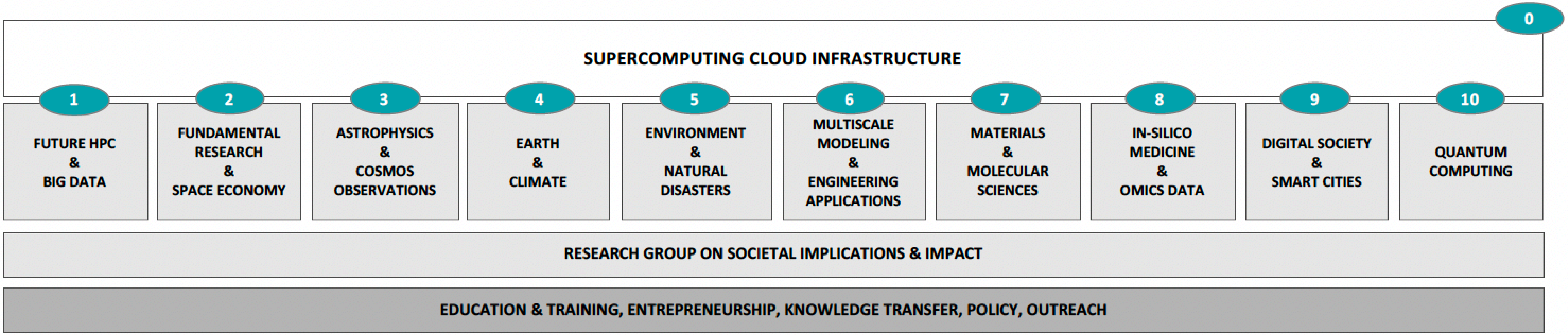


Figure 1 The organization chart of the National Centre

ICSC: National Centre on HPC, Big Data and Quantum Computing project

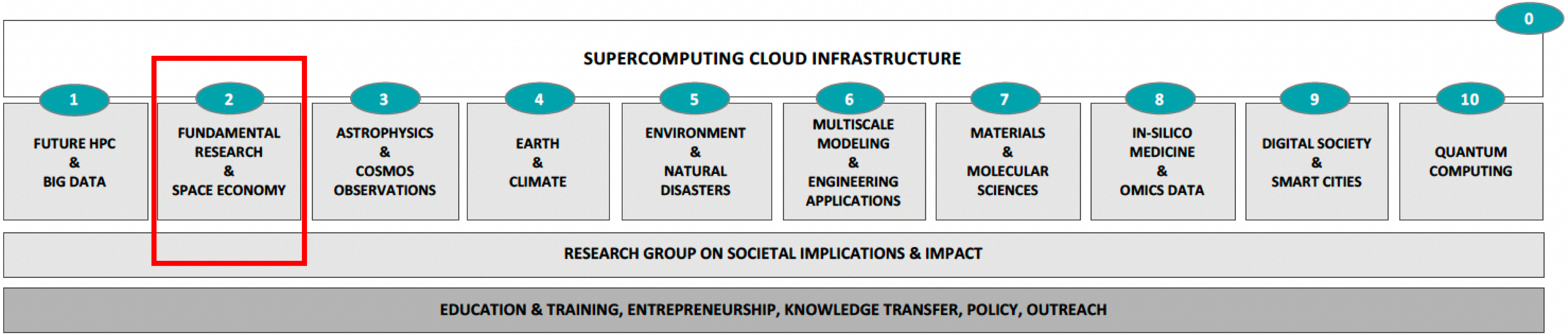
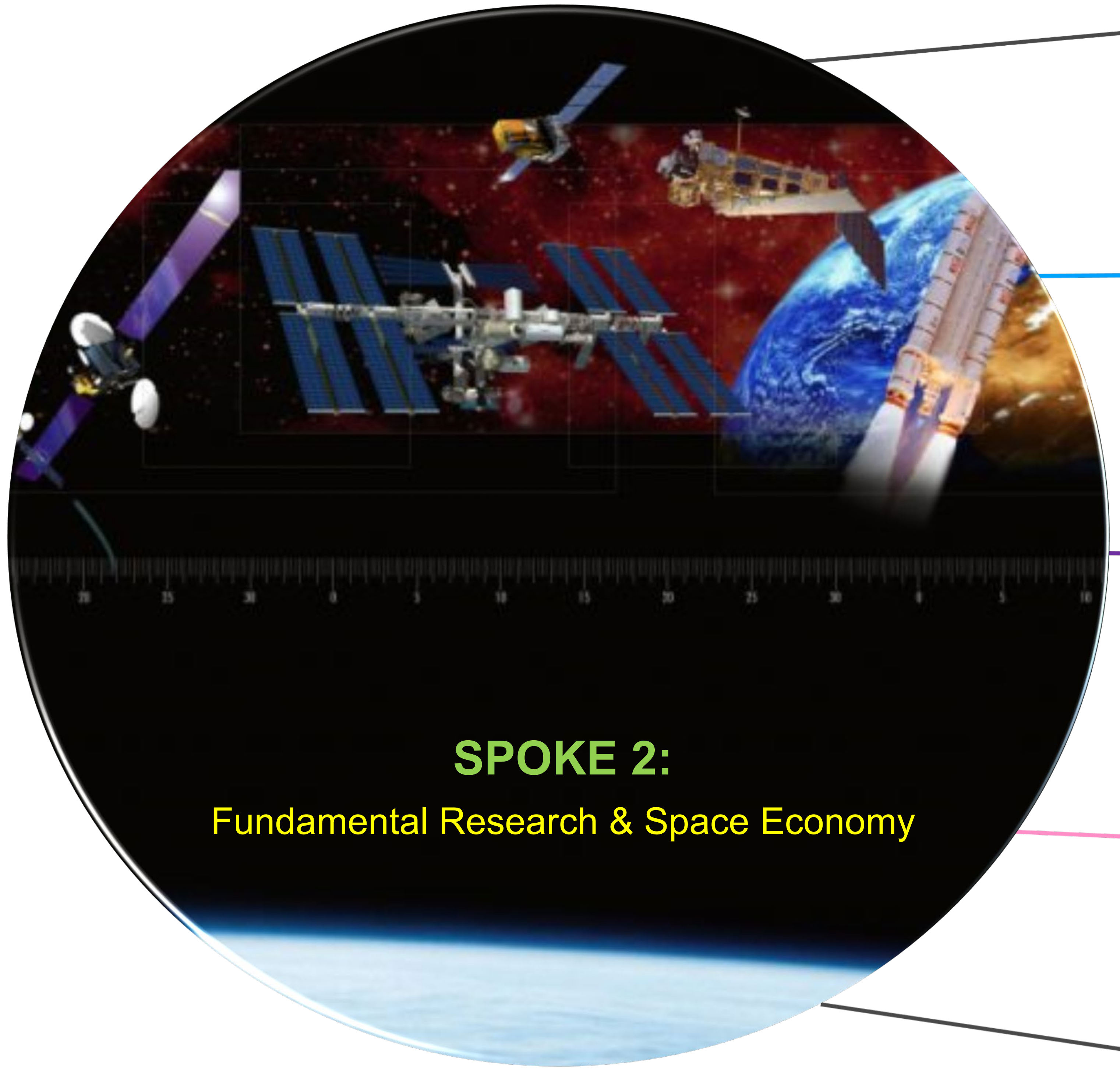
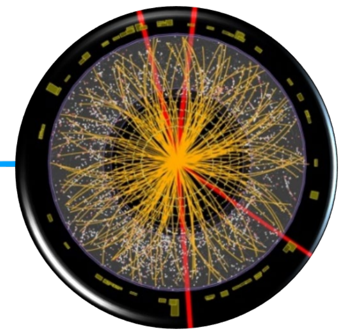


Figure 1 The organization chart of the National Centre

Spoke 2 - Fundamental Research & Space

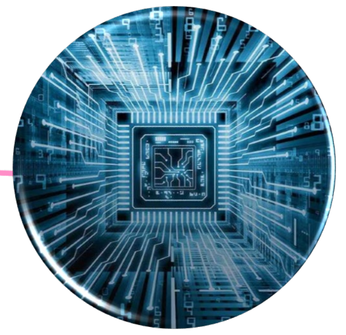


WP1: Design and development of science-driven tools and innovative algorithms for Theoretical Physics



WP2: Design and development of science-driven tools and innovative algorithms for Experimental High Energy Physics

WP3:Design and development of science-driven tools and innovative algorithms for Experimental Astroparticle Physics and Gravitational Waves



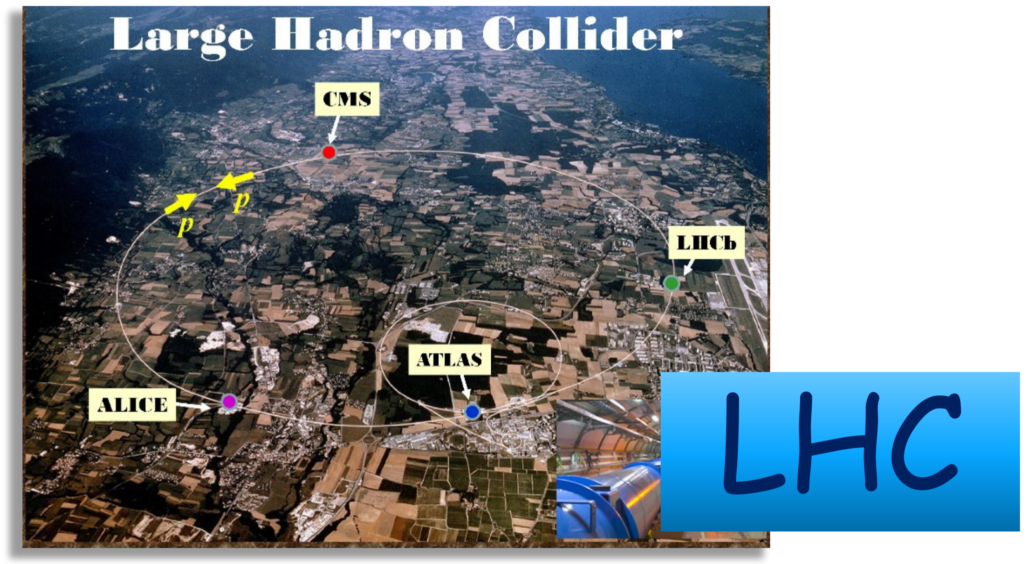
WP4: Boosting the computational performance of Theoretical and Experimental Physics algorithms

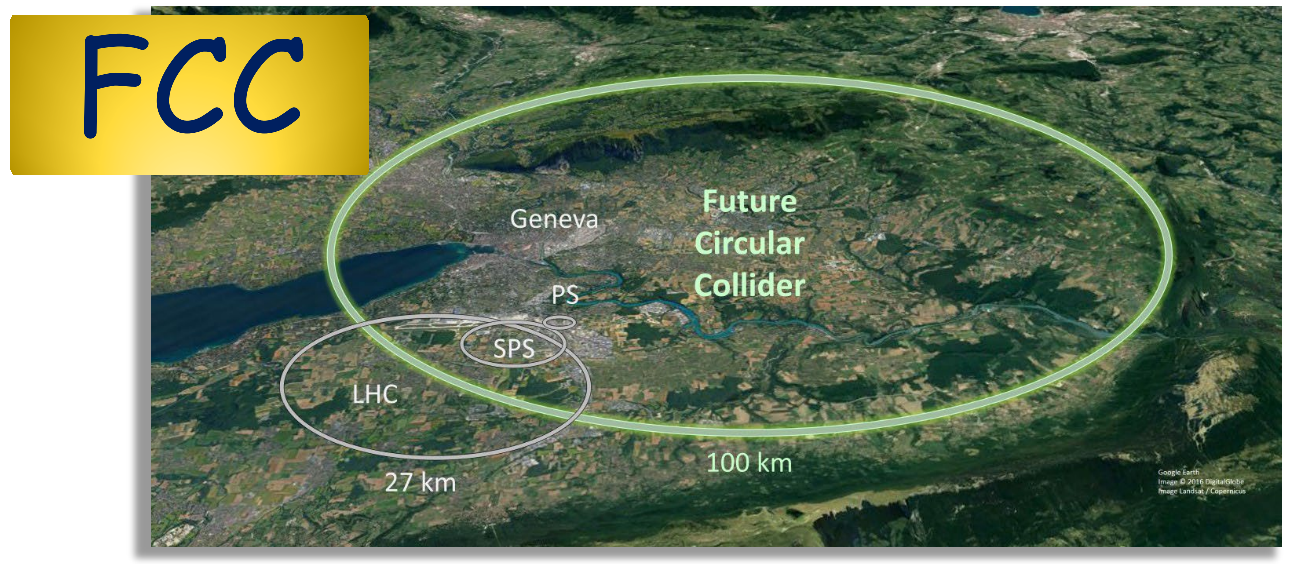
WP5: Architectural Support for Theoretical and Experimental Physics Data Management on the Distributed CN infrastructure

WP6: Cross–domain Initiatives

Motivations

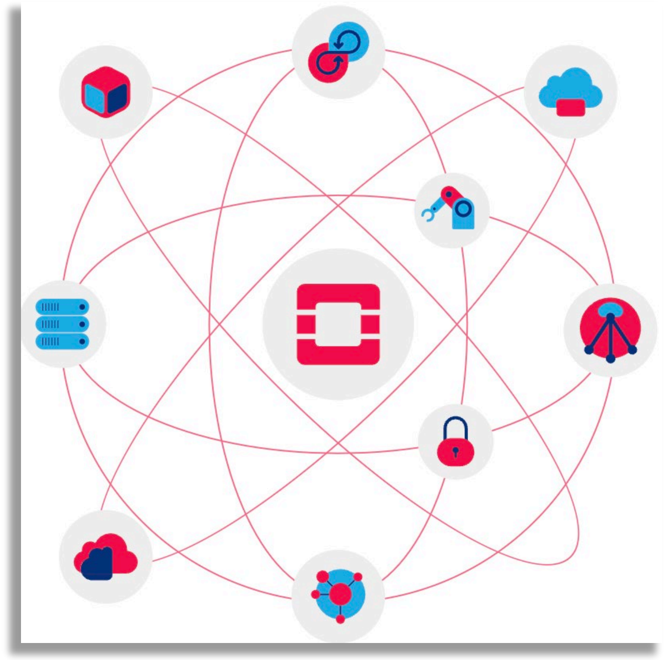
- **Challenges of LHC, HL-LHC and of the Future Colliders** are pushing to **re-think the HEP computing models** having strong impact on several aspects, from software to the computing infrastructure
- From the software perspective, **interactive/quasi-interactive analysis** is a promising paradigm
  - User-friendly environment
  - Adopting open-source industry standards: *Dask, Jupyter Notebooks* and *HTCondor*
  - Validating new frameworks (e.g. *ROOT RDataFrame* with multi-threading)





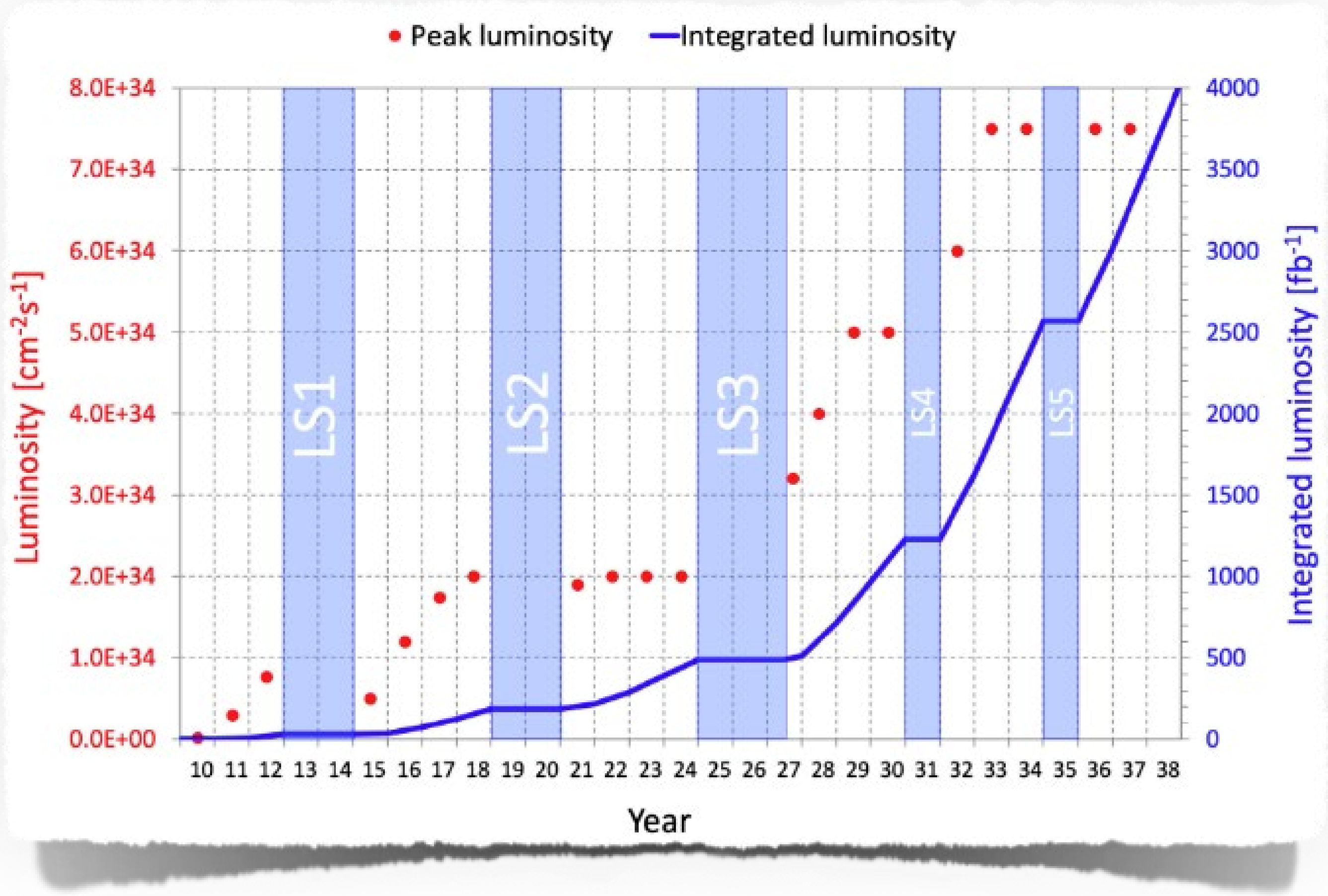
- **Preliminary feasibility studies** exploiting **LHC data** collected by the ATLAS detector and using simulation and studies for **future e^+e^- colliders pseudo-data**
  - Testbed **infrastructure for high throughput data analysis**
  - The local deployment is based on the *Open-Stack Infrastructure as a Service* paradigm (IaaS)

openstack.

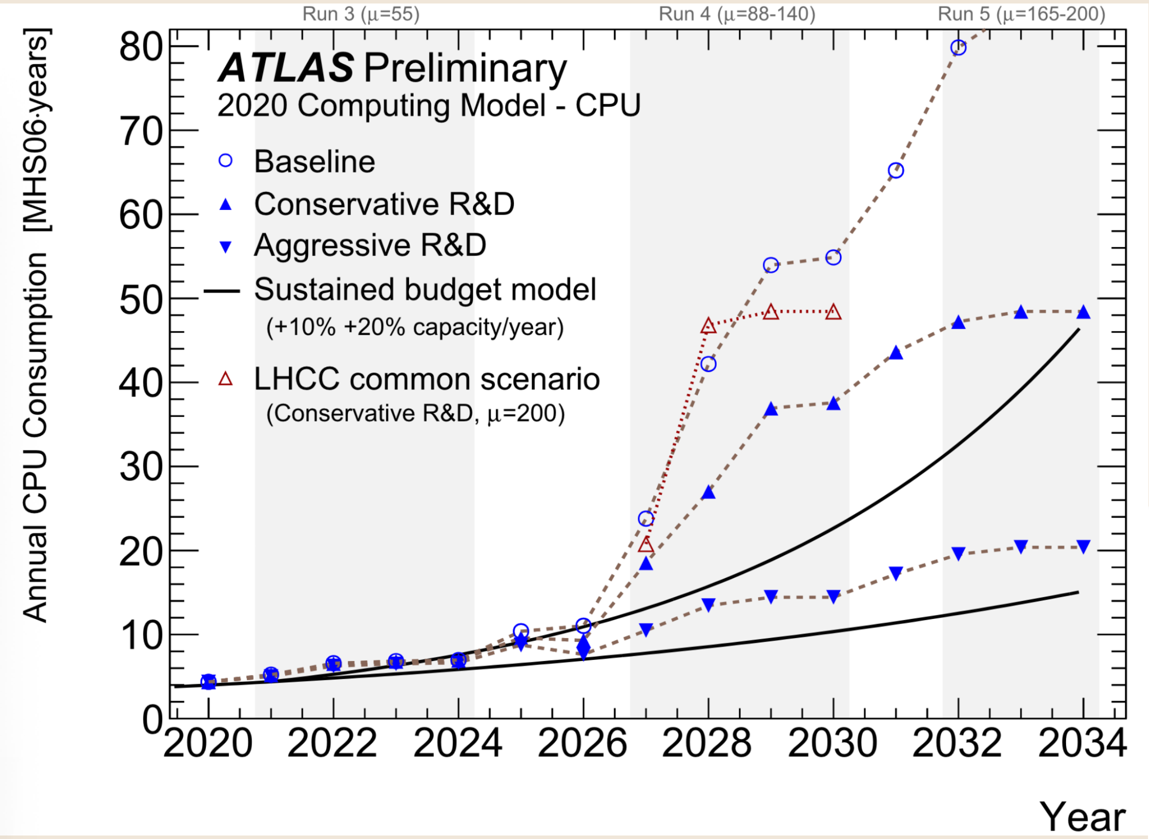


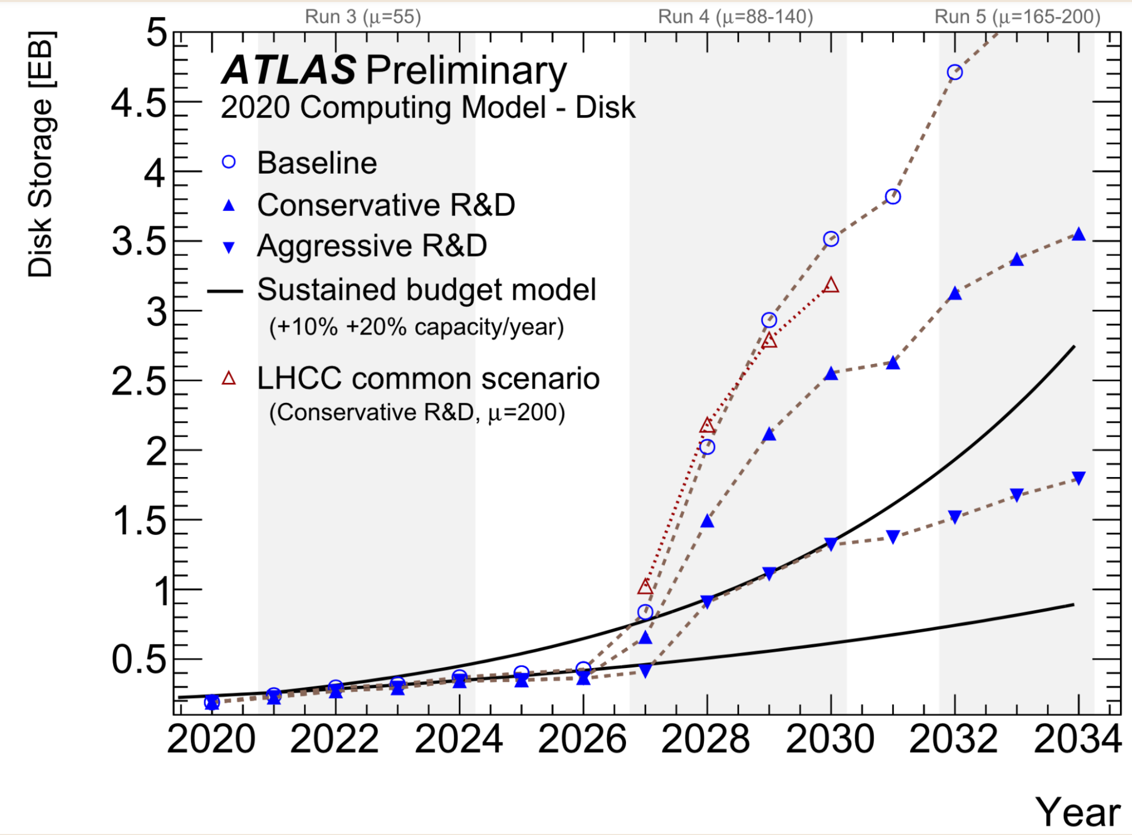
Motivations

- **Challenges of LHC, HL-LHC and of the Future Colliders** are pushing to **re-think the HEP computing models** having strong impact on several aspects, from software to the computing infrastructure



ATLAS HL-LHC projections





To efficiently analyze this increasing amount of Big Data:

- Optimize the usage of CPU and storage;
- Promote the usage of better data formats;
- **Develop new analysis paradigms**!

  - New software based on declarative programming and interactive workflows;
  - Distributed computing on geographically separated resources

Higher rates of collision events → **Higher demand for computing and storage resources**

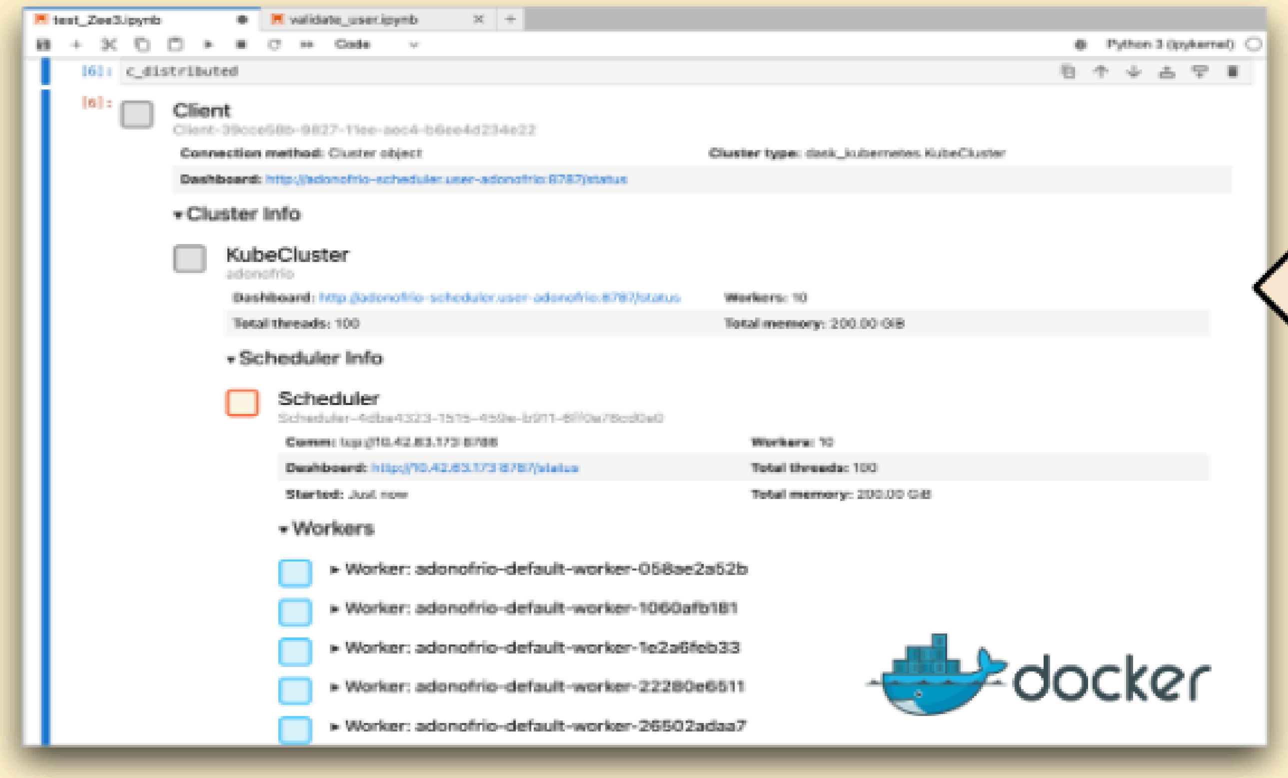
High throughput data analysis platform

Access and security

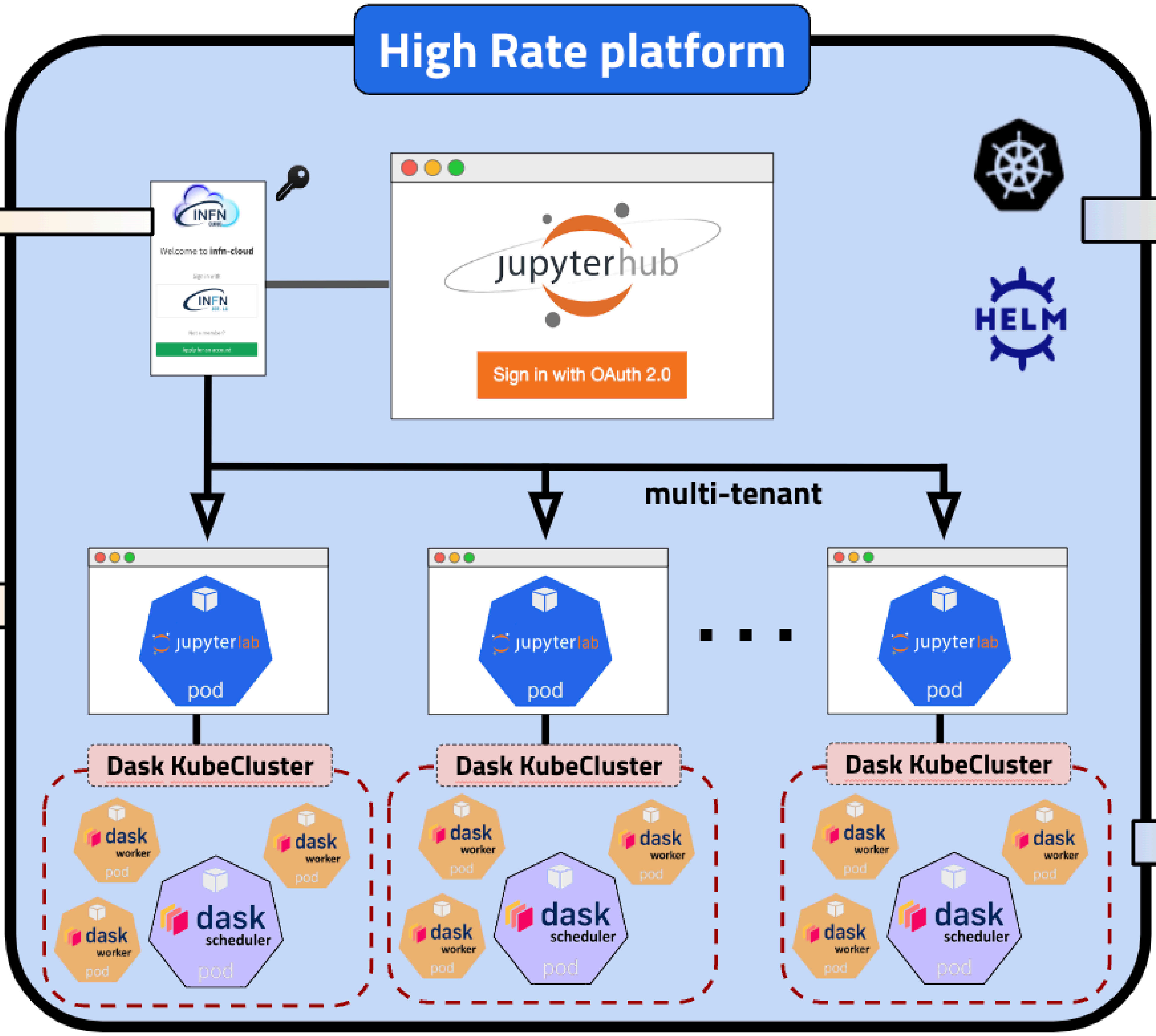
After connecting to an entrypoint URL, the user reaches a **Jupyterhub** instance that, after authentication and authorization via INDIGO-IAM, allocates the required resources for the user's working area

User Interface

The user interface is based on **Jupyterlab**, customised with specific plugins for specific purposes (e.g. Dask).



The working environment is highly customizable, using tailored Docker containers. This is important when analyses require specific software (collaboration-wise)



Deployment

The deployment of the **Kubernetes** resources needed for the spawning of this platform, is handled via **HELM charts** available in the GitHub organization.

Check the docs!

This allows a seamless, flexible, scalable and fault-tolerant deployment on the available resources, with a limited impact on the admin's work time

Software

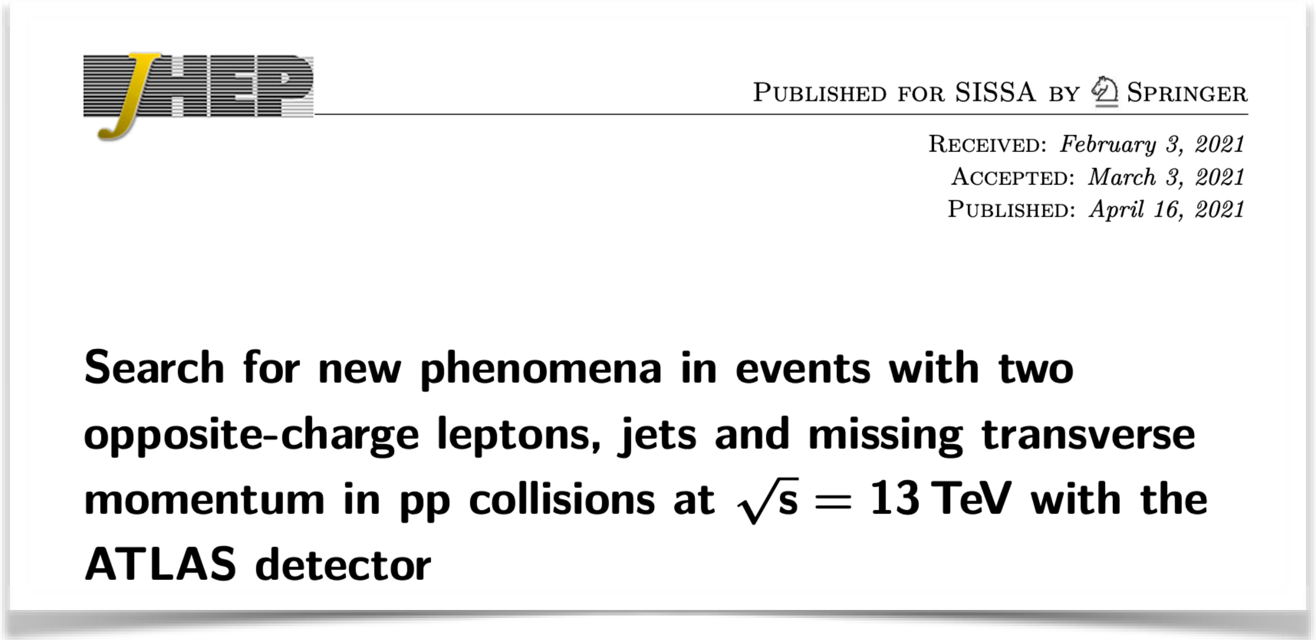
From the software perspective, interactive/quasi interactive analysis is a promising paradigm

- User-friendly environment
- Adopting open-source industry standards: *Dask, Jupyter Notebooks* and *HTCondor*
- Validating new frameworks (e.g. *ROOT RDataFrame* with multi-threading)

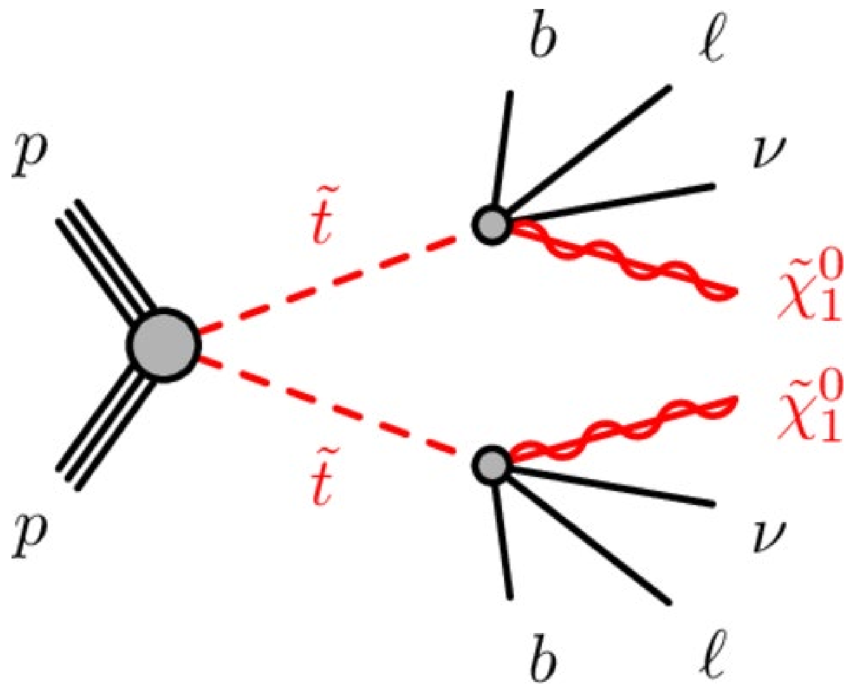
Benchmark interactive analyses
Use-cases

ATLAS use-case I

SUperSYmmetry: Beyond Standard Model theory


JHEP
Published for SISSA by Springer
Received: February 3, 2021
Accepted: March 3, 2021
Published: April 16, 2021

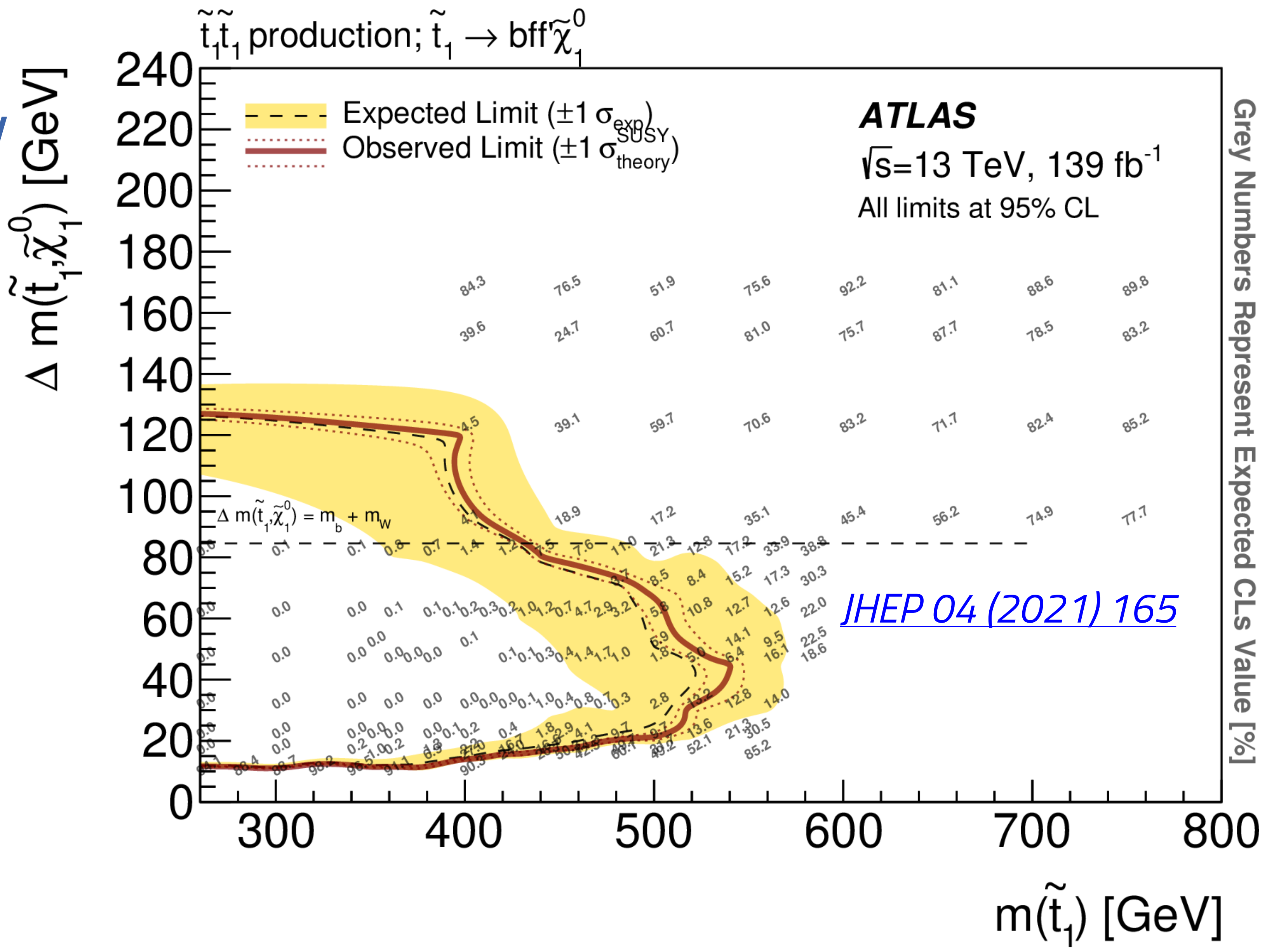
Search for new phenomena in events with two opposite-charge leptons, jets and missing transverse momentum in pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector



Soft leptons coming from a virtual W* boson decay

Compressed mass spectra:
$\Delta m < m_W + m_b$

- Three different analysis in the ***Run 2 paper***, already published, according to mass splitting between *stop* ($\tilde{t}_1$) and *neutralino* ($\tilde{\chi}^0_1$), allowing different decay modes:
  - 2 body → $\Delta m > m_t$
  - 3 body → $m_W + m_b < \Delta m < m_t$
  - 4 body, the one picked up → $\Delta m < m_W + m_b$
- Common final state signature: 2 OS leptons (electrons/muons), jets and missing transverse energy
- Cut & Count based approach



JHEP 04 (2021) 165

INFN Lecce and Naples Collaborative Work

4-body search workflow

Analysis Facility

Skimming
- Provided by the Collaboration
- Offline reconstruction
- $\mathcal{O}$(PB) for data and MC

DAOD reduction

Thinning
- Removal of collections
- Baseline objects and trigger
- Scale Factors retrieval
- $\mathcal{O}$(TB) for data and MC

Slimming
- Removal of object quantities
- New variable definitions
- Weights application
- $\mathcal{O}(10^2$ GB) for data and MC

Event Selection
- Region definitions
- Nominal yields
- Systematic variations
- $\mathcal{O}$(MB), inputs to fitting tool(s)

Slimming

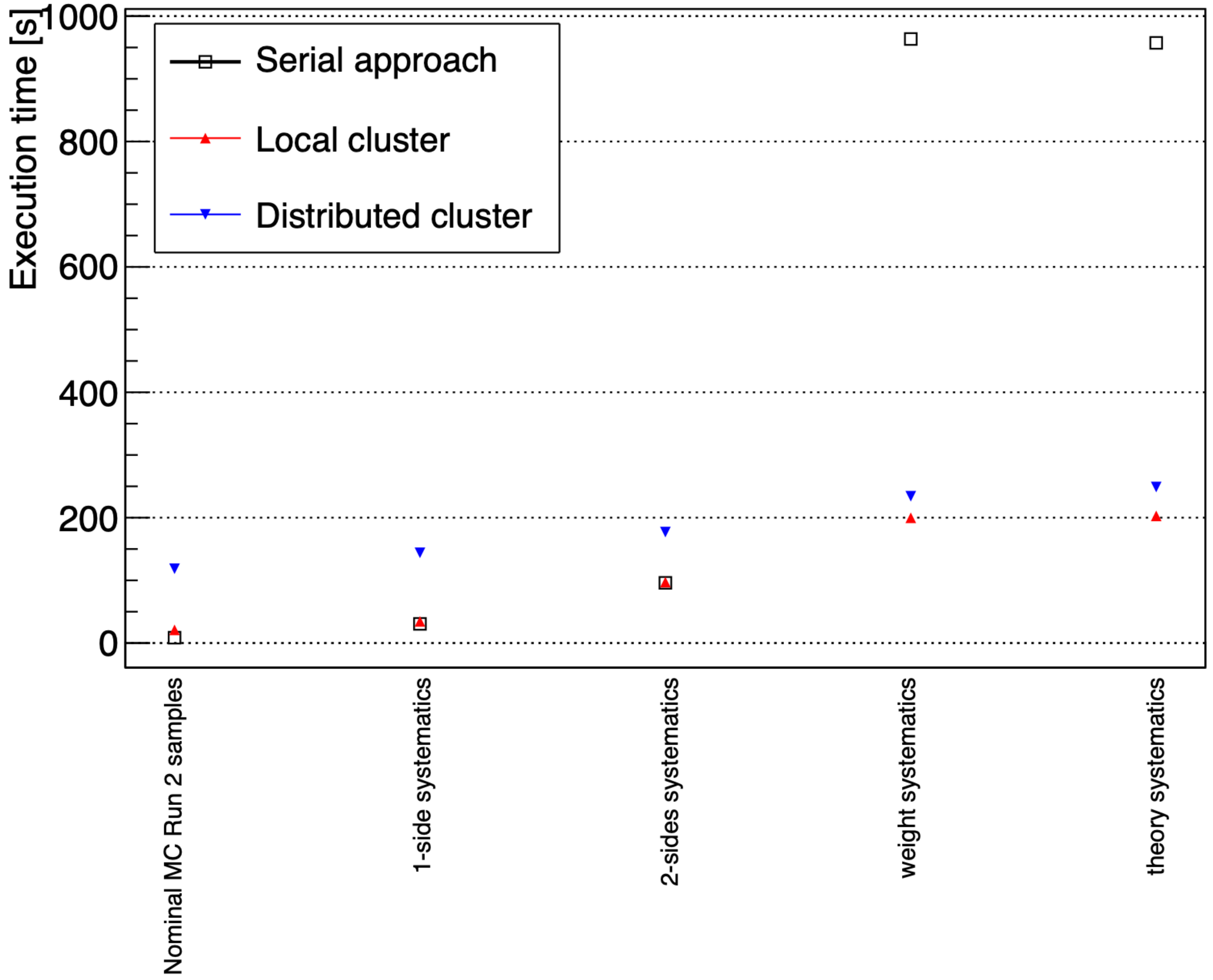
ATLAS slimming code already in RDataFrame, but entirely written and compiled in C++ —> NO dask distributed approach

Event Selection

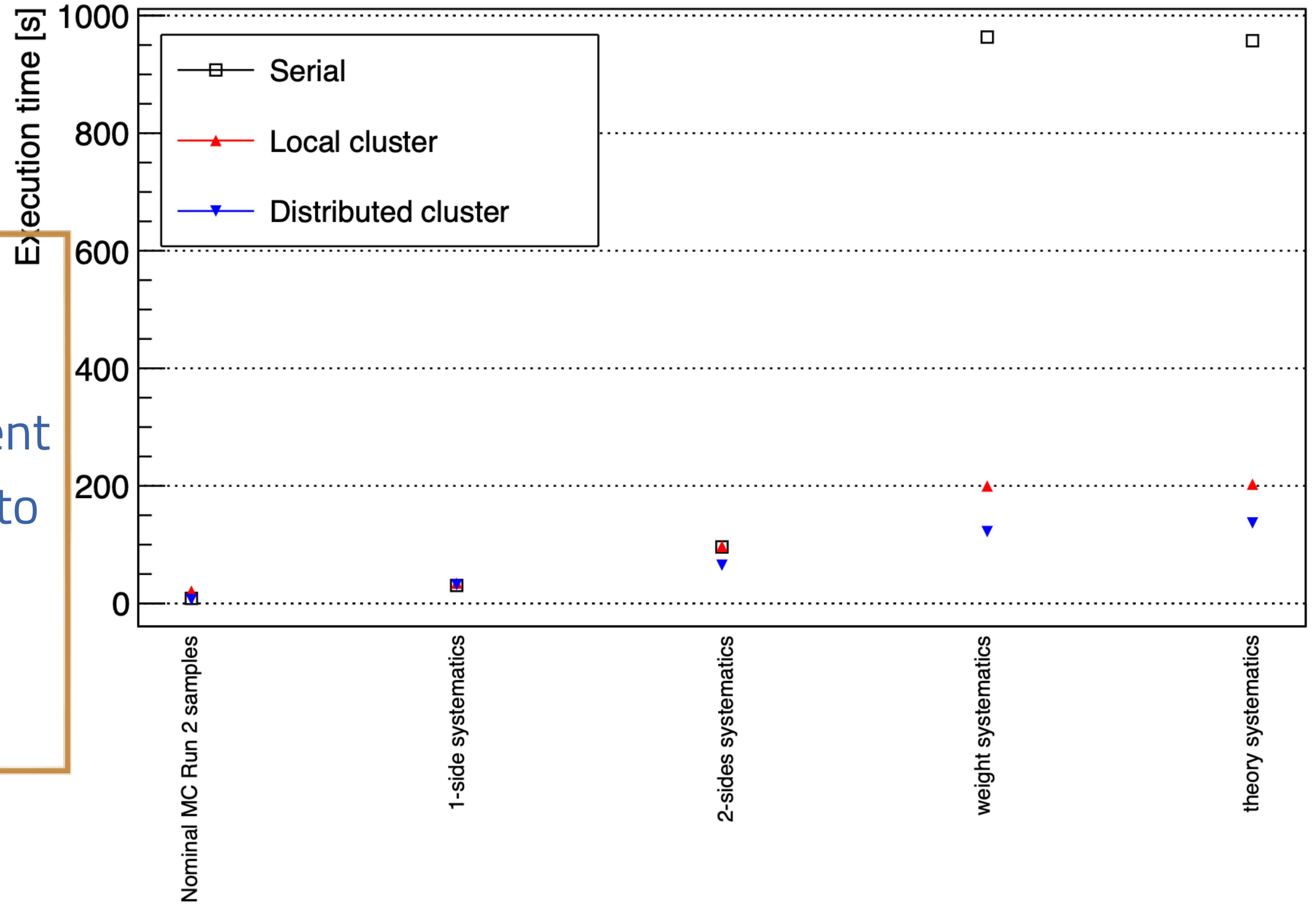
- Event selection for fitting tools
- RDataFrame + Dask applied to Wt bkg sample ~ 1.8 GB
- Code ready to play with other backgrounds

Preliminary results

Defined Metric	
Overall execution time	Time elapsed from the start of the execution (execution triggered) to the end of execution



Limiting factor for the distributed approach: time spent to copy the inputs to the worker nodes (~120 s)—> **TO BE IMPROVED**



- Exploiting the distributed approach, the execution time improves *wrt* the standard/serial approach if we iterate over a significative number of systematic variations (each step in the x-axis includes previous contributions)

ICSC Workshop on Analysis Facilities

- **8, 9, 10 Gennaio 2024,** Bologna. Agenda: link

- First part **open** to everyone, with lectures and hands-on covering aspects on distributed data analysis with ROOT and pure Python.
- Second part **restricted** to experiment communities, covering specific analyses' overview as well as future perspectives given by the collaboration side-groups.

Tommaso Diotalevi (UniBO), Francesco G. Gravili (UniSalento)

Conclusions & Next Steps

- Interactive analyses feasibility studies on INFN cloud succeeded
  - Performance evaluated using Dask on the local client or distributed cluster, *wrt* original implementation

Short term goals:

- Deploy of the code & relative instructions to allow other users to test quasi-interactive high throughput data analysis platform
  - Benchmark studies with local performance evaluation

→ **Medium-long term goals:**

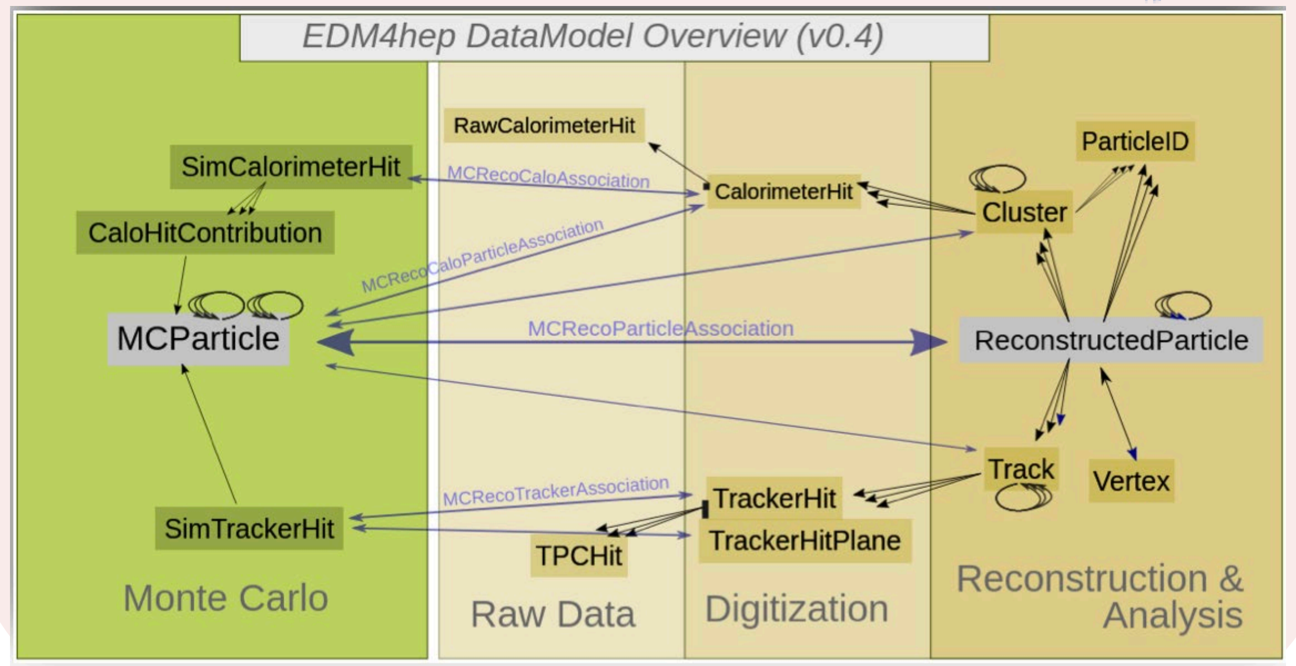
- Evaluate scalability and simultaneous performance with increasing number of workers

Thank you!

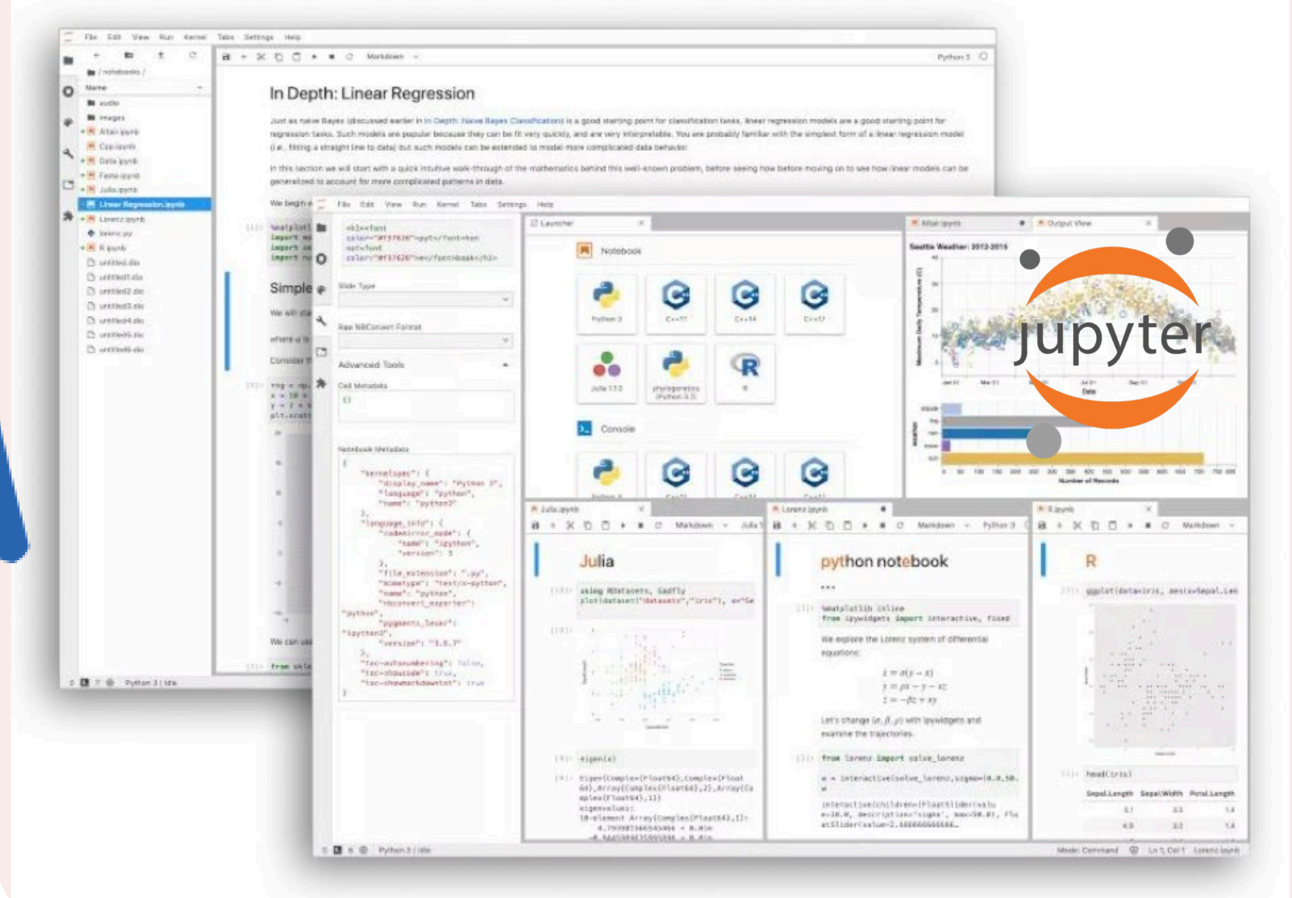
Back - up

FCCe+e- use-case

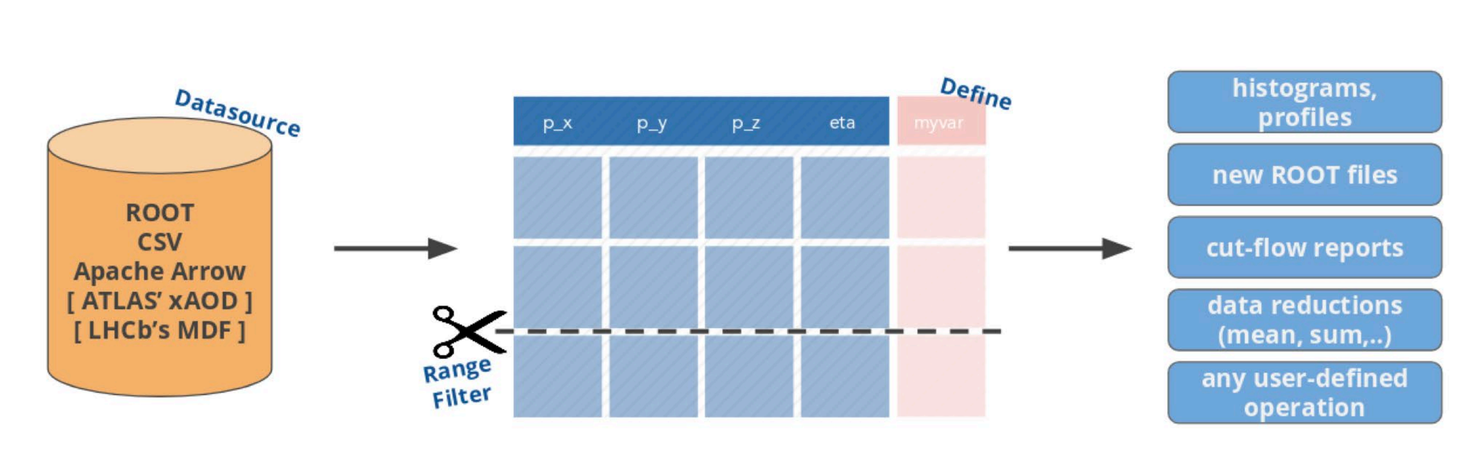
EDM4hep input data format
flat input ntuples



New approach to data analysis

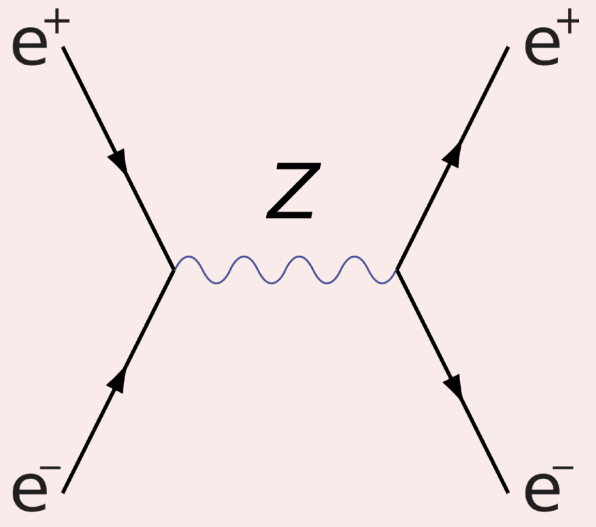


Selection and histogramming interactively via RDataFrame on JupyterHub

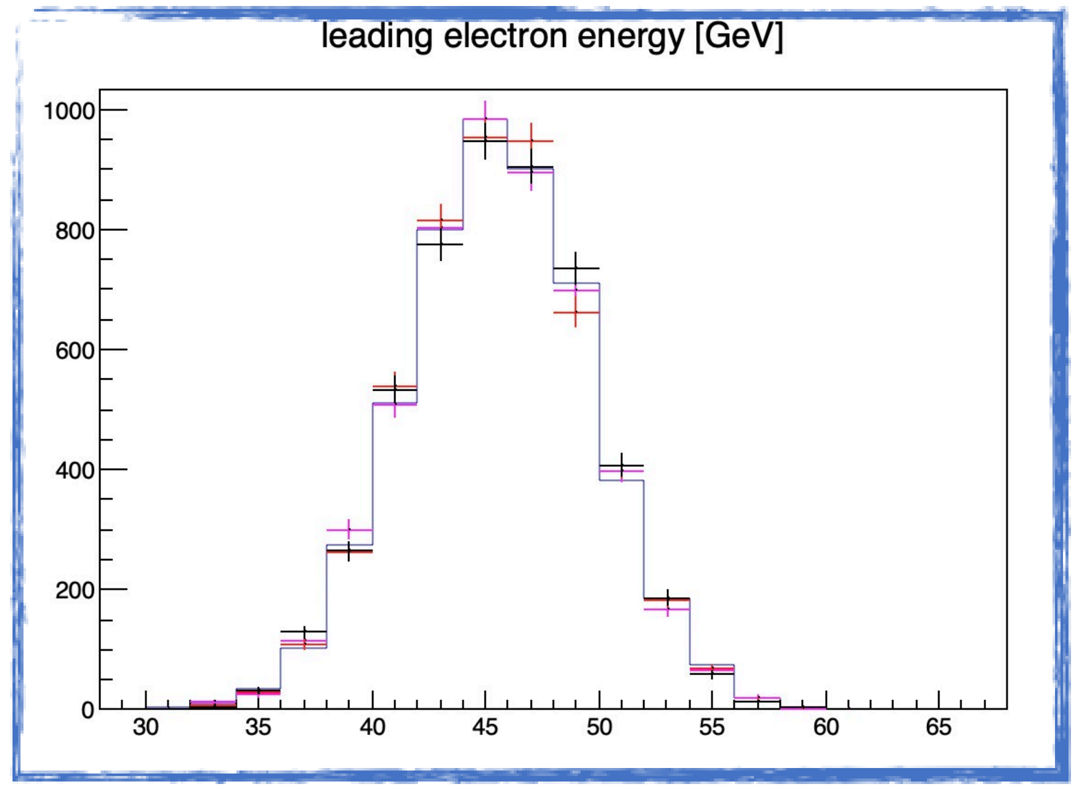


+ dask used as backend

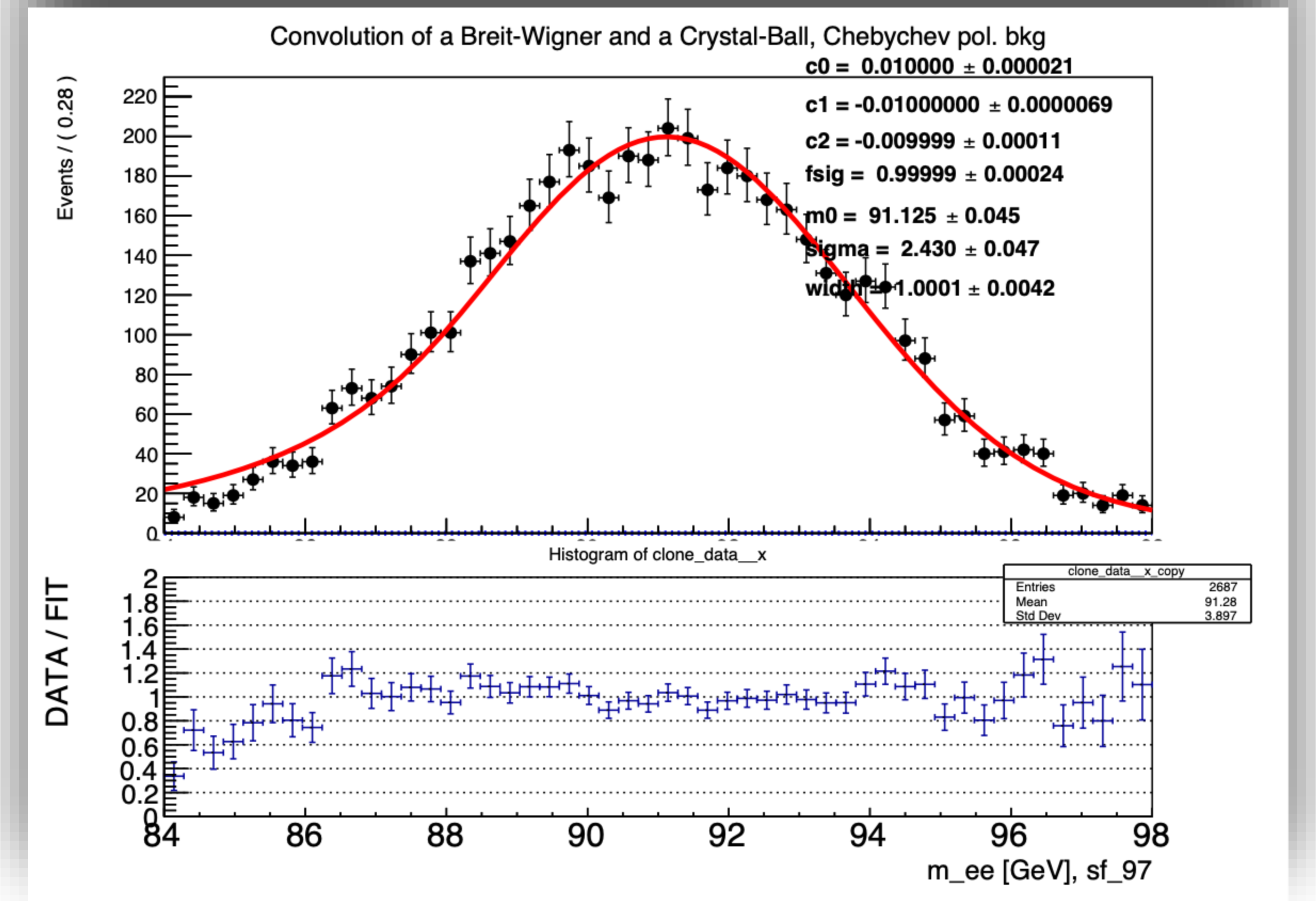
Feasibility study



:



Mee invariant mass fit



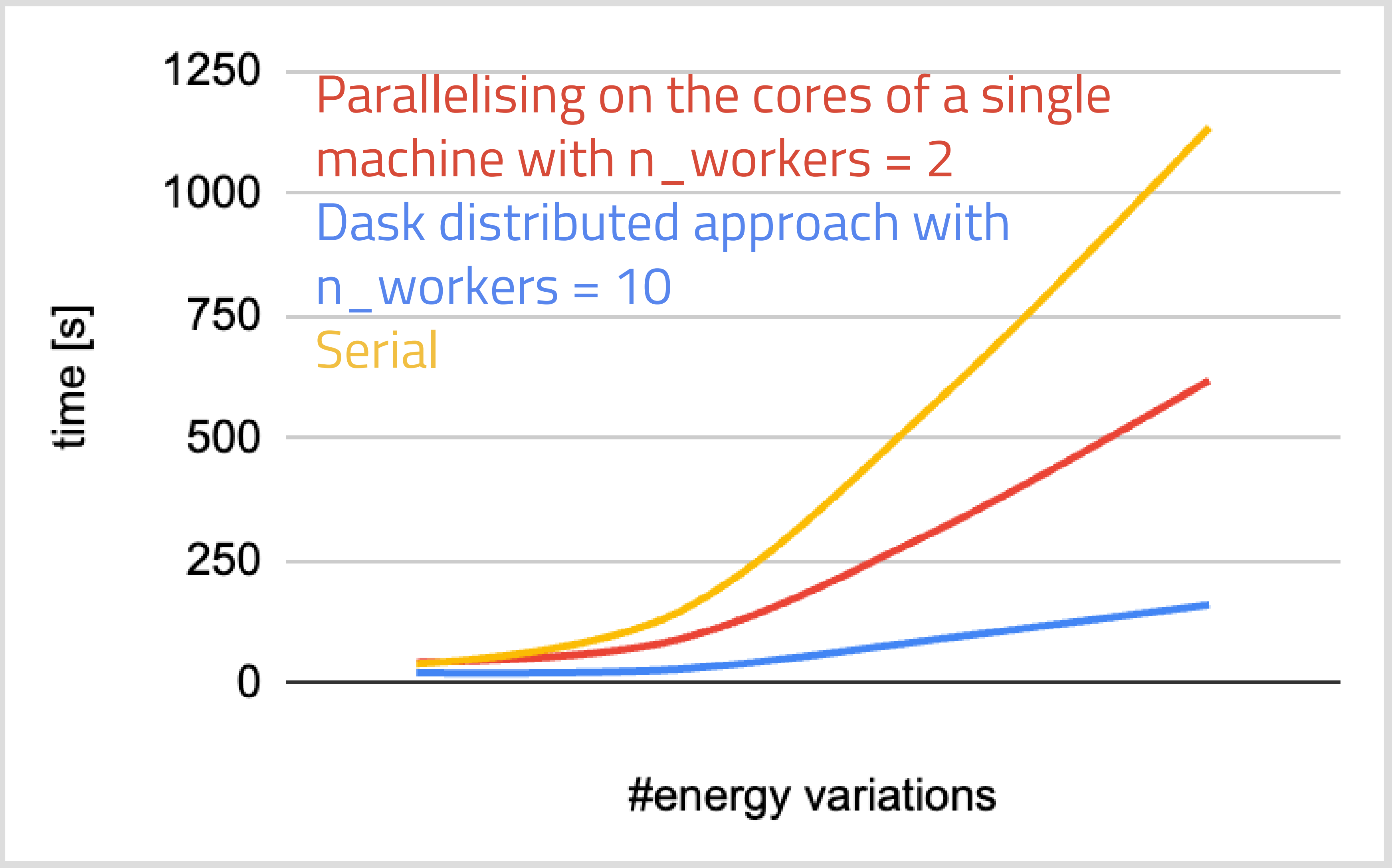
- Mimic systematic variations: e^+e^- energy gaussian smearing

ECFA presentation *link*

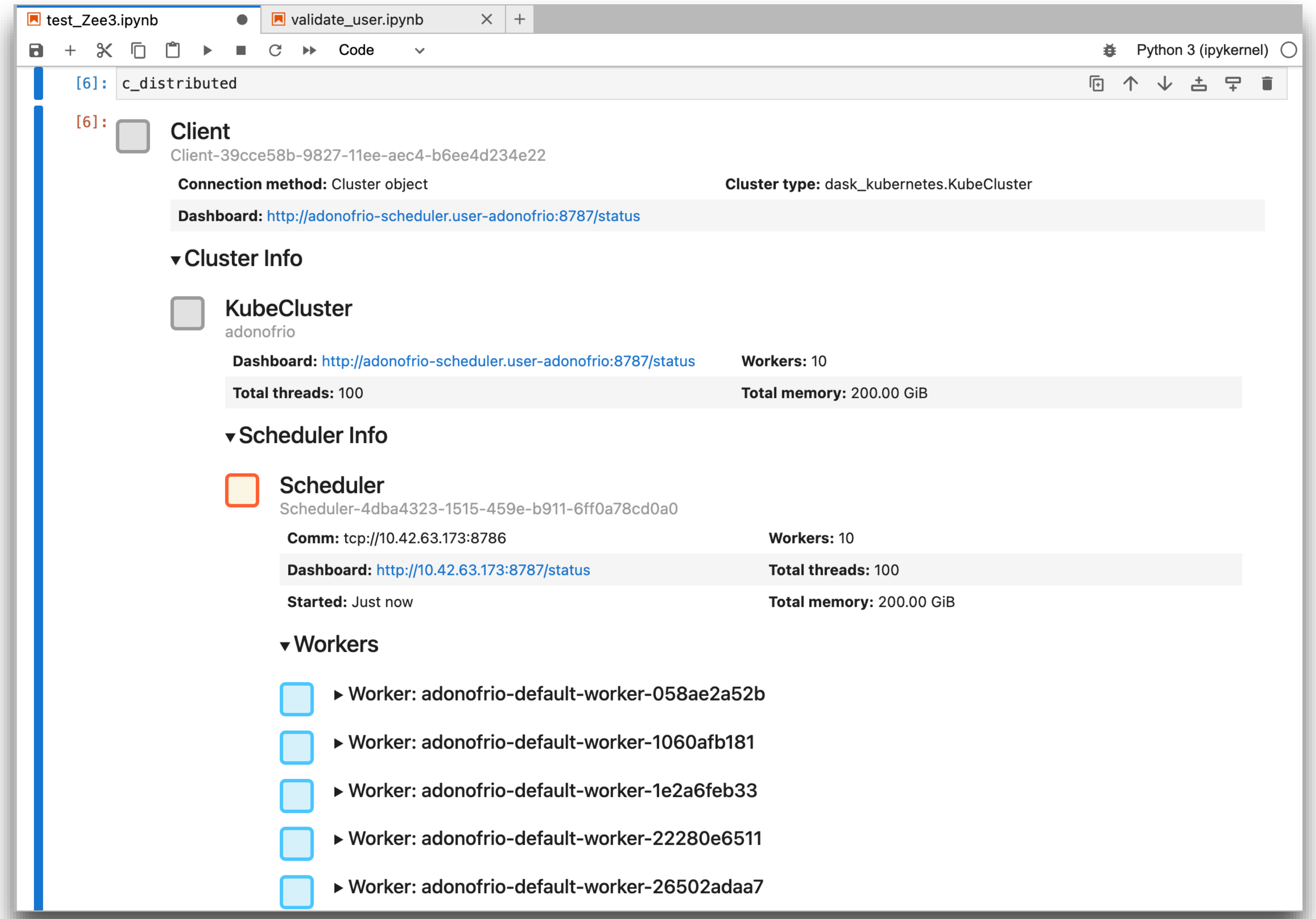
github link to the code

Preliminary results: distributed cluster

- Kubernetes infrastructure: (Kubernetes workers & Kubernetes master) on *Open-stack*



# iterations	Serial approach	Local client Dask	Distributed Dask
50	590 s	320 s	75 s
100	1135 s	618 s	138 s



- Moving to a distributed Dask model and **scaling resources, the performance improves**
- Advantage: use this use case as simple test for who wants to benefit from the infrastructure

Preliminary results: local client

Scaling without changing your code

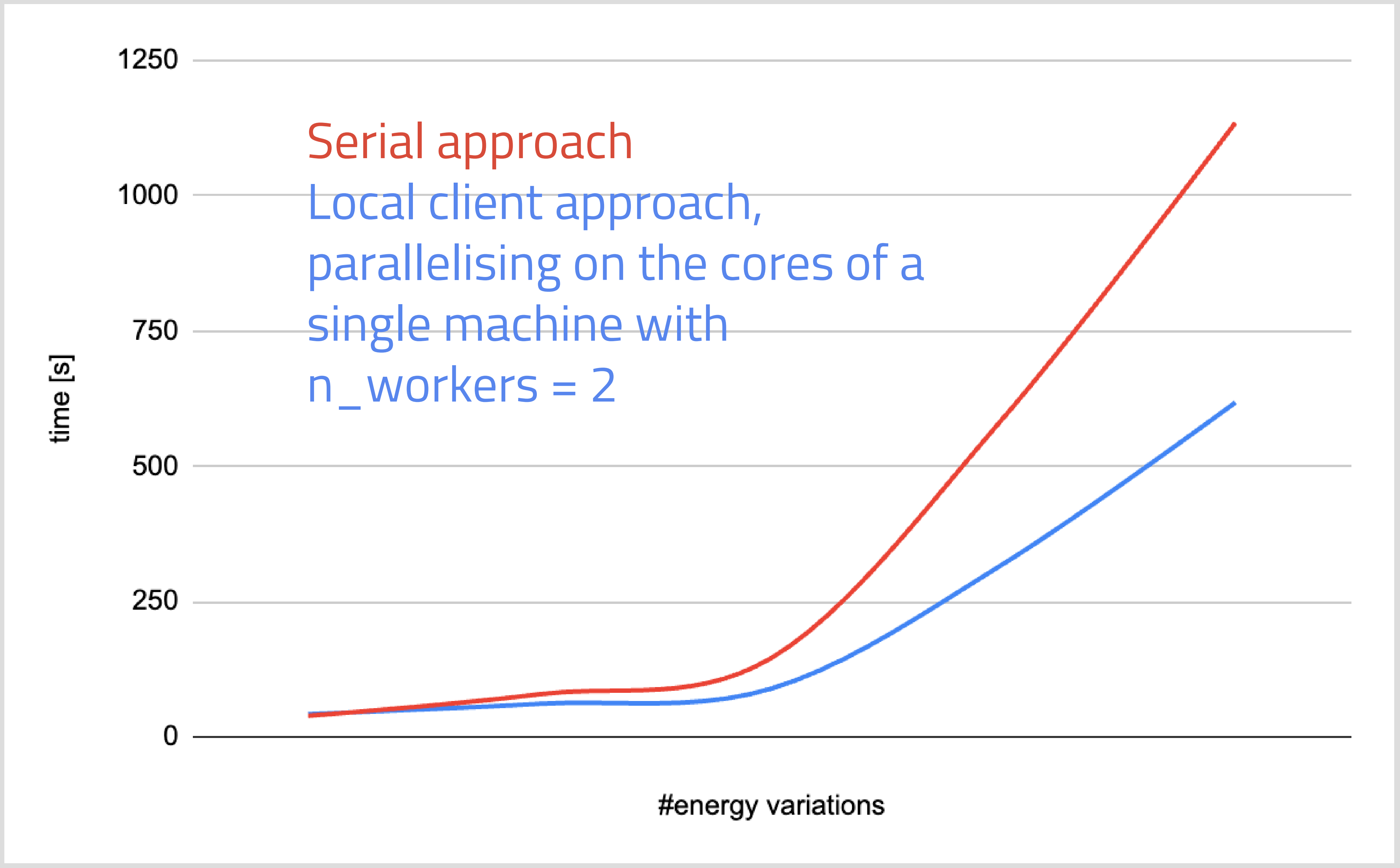
```
from dask.distributed import LocalCluster, Client
if distributed == True:
    RDataFrame = ROOT.RDF.Experimental.Distributed.Dask.RDataFrame
    ROOT.RDF.Experimental.Distributed.initialize(my_initialization_function)
else:
    RDataFrame = ROOT.RDataFrame
    my_initialization_function()
```

Parallel (the `if distributed == True:` branch)

Serial (the `else:` branch)

No changes required to the rest of the code

```
df = df.Define('w_nominal', '1')
df = df.Define("m_e","0.0005124") #GeV
df_ge = df.Define("goodelectrons", "Particle.charge[0]*Particle.charge[1] < 0.").Filter("goodelectrons > 1")
```



How to compare the performance?

Defined Metric	
Overall execution time	Time elapsed from the start of the execution (execution triggered) to the end of execution

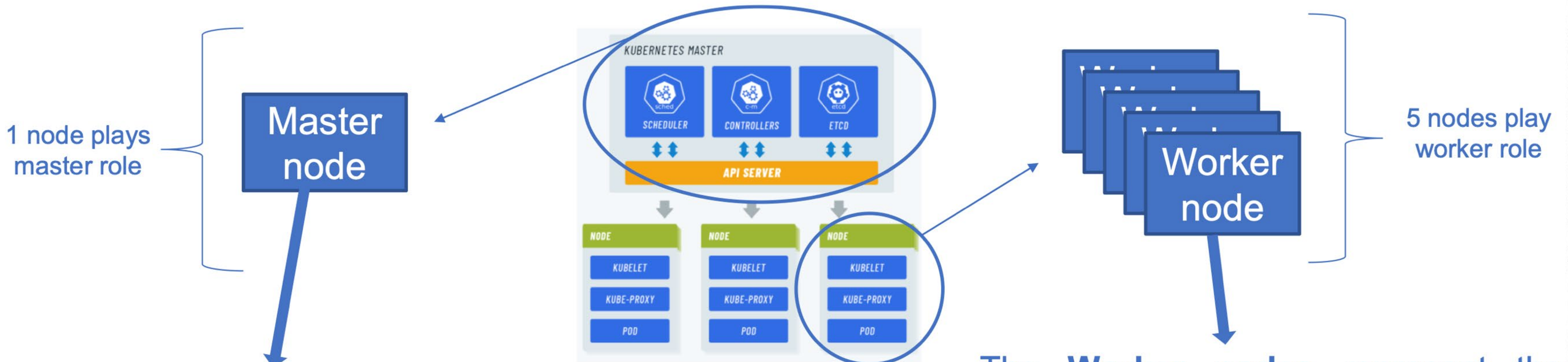
- Exploiting the local client approach, the execution time improves *wrt* the standard/serial approach if we iterate over a significative number of energy variations (> 10)

High throughput data analysis platform

- **Goal:** provide the users with an infrastructure that represents a tradeoff between deployment speed-flexibility, resource efficiency and service performance
- **Solution being tested:** the use of container technology (via Docker 20.10) that runs the applications and the Kubernetes tool for orchestration

Gianluca's talk

Local testbed infrastructure provides 6 nodes, orchestrated via **Kubernetes (1.26.3)**:



The **Master node** coordinates and manages the entire Kubernetes infrastructure. It includes the API Server, the Controller Manager, the Scheduler, and etcd.

The **Worker nodes** represent the nodes of the cluster where containers are executed. Each one hosts the Kubelet, Kube-proxy and the containers, providing the actual computational capacity of the cluster.

Efficient & user friendly infrastructure

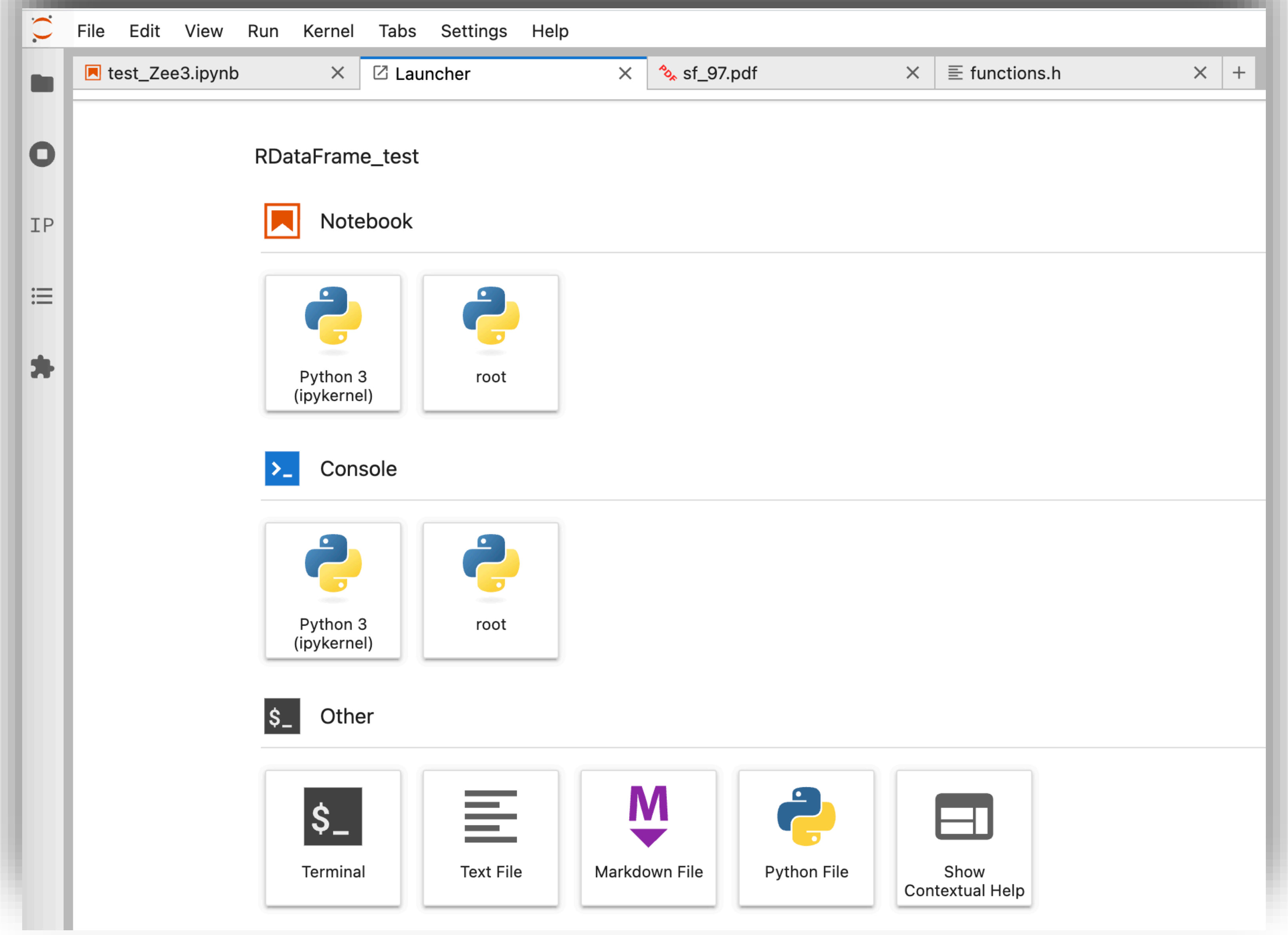
- 2 nodes equipped with **Docker** (20.10) for containerisation and **Kubernetes** (1.26.3) for orchestration

Jupyter

The JupyterLAB environment allows users to exploit data science python libraries and to scale them over the cluster

Dask

A python library to scale python code from multi-core local machines to large distributed clusters in the cloud



- Jupyter interface includes:
  - Terminal
  - Notebook implementation
    - Completely exportable and replicable

Gianluca's presentation *link*