



Finanziato
dall'Unione europea
NextGenerationEU



Ministero
dell'Università
e della Ricerca



Italiadomani
PIANO NAZIONALE
DI RIPRESA E RESILIENZA



Leveraging distributed resources through high throughput analysis platforms for enhancing HEP data analyses



Adelina D'Onofrio¹, Tommaso Diotallevi^{1,3}, Francesco Giuseppe Gravili^{1,5}, Salvatore Loffredo^{1,2}, Elvira Rossi^{1,2}, Federica Maria Simone^{1,4}, Bernardino Spisso¹

on behalf of the ATLAS and CMS Collaborations

CHEP2024, 19-25 Oct 2024, Krakow

1 INFN, 2 University Federico II, 3 University of Bologna, 4 Polytechnic Bari, 5 Università del Salento

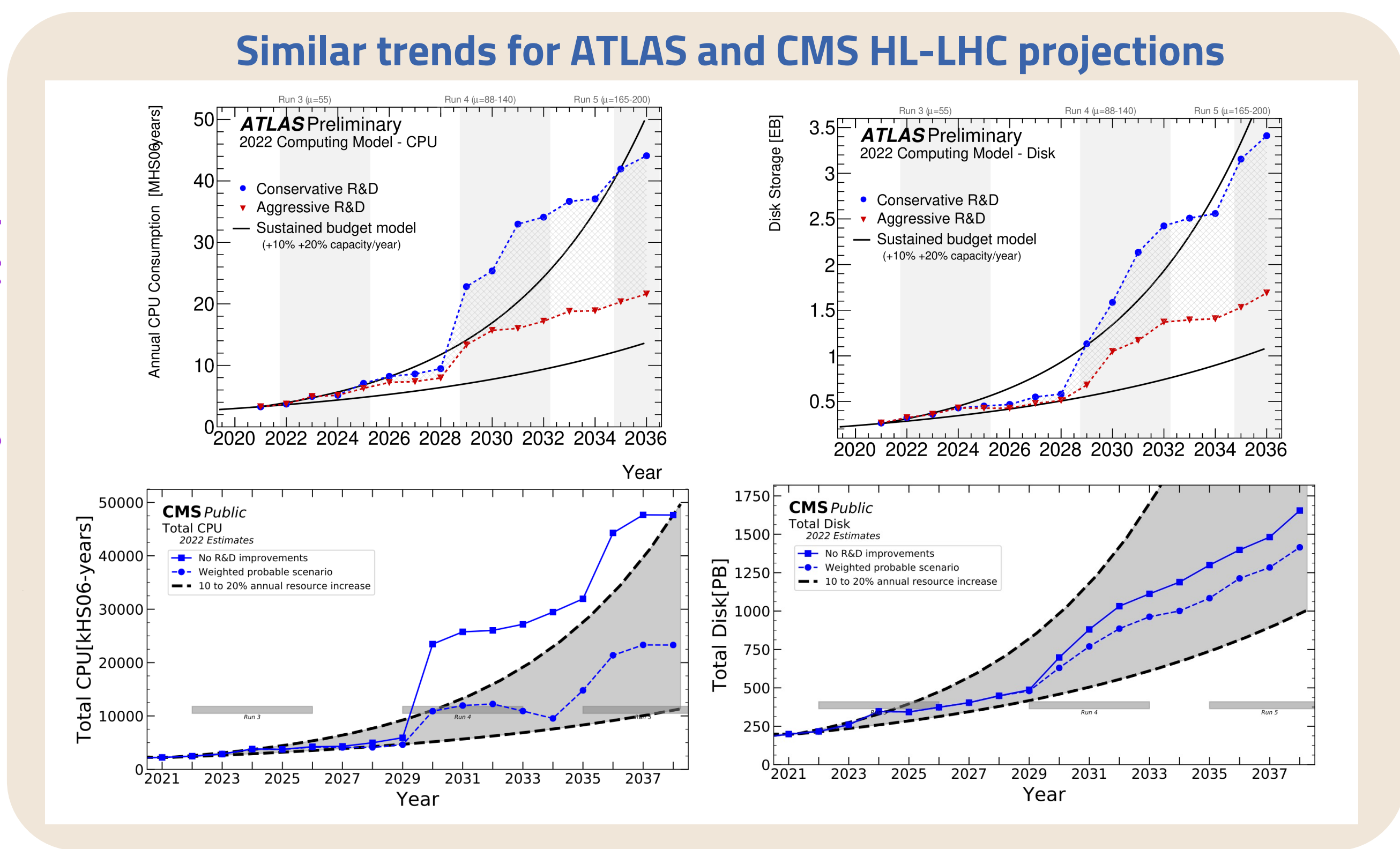
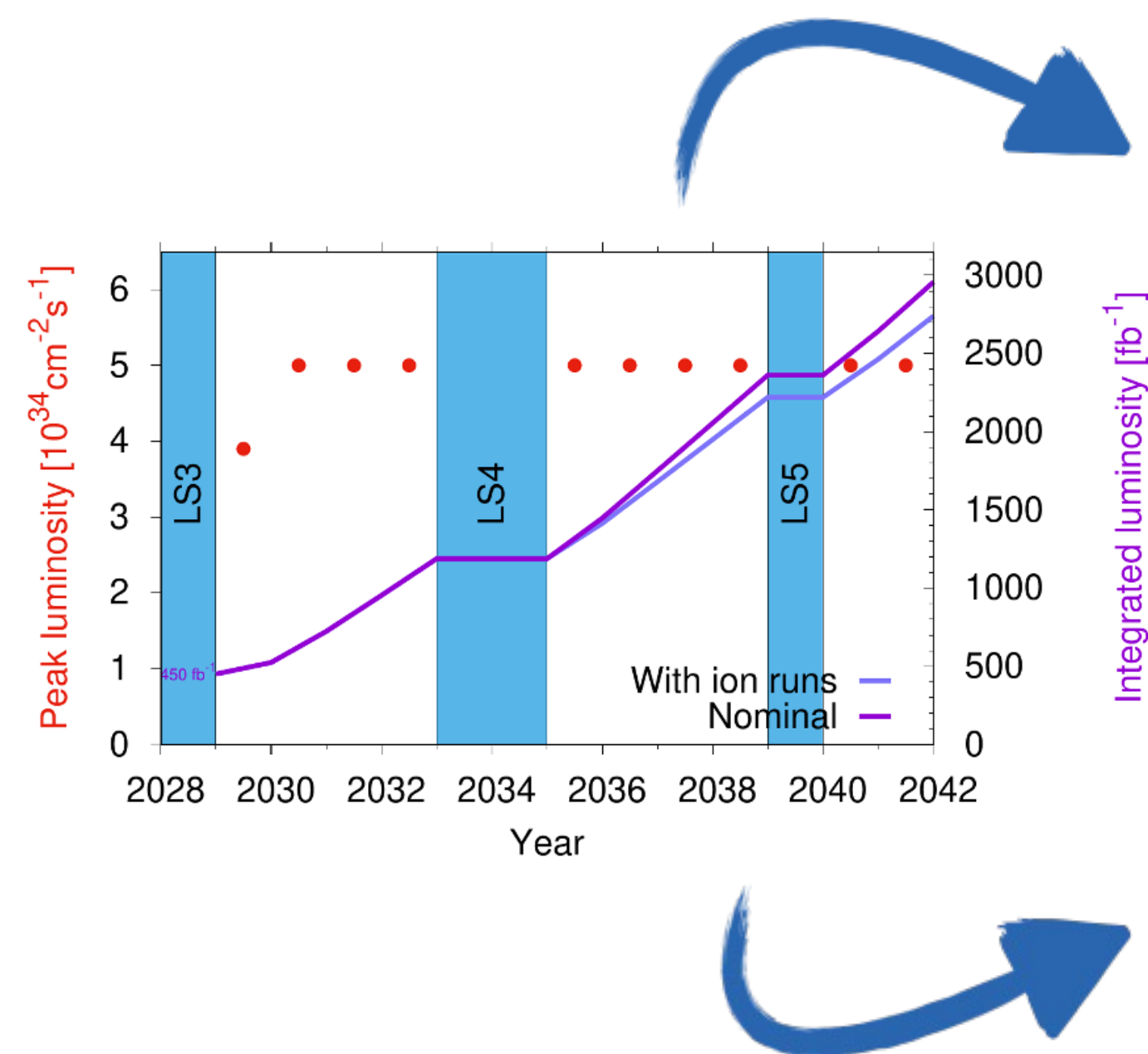
Outline

- Motivations
- Test infrastructure
- Analysis use-cases:
 - Search for SUSY signatures at ATLAS
 - Flavor physics and search for rare decays at CMS
 - Preliminary scalability results
- Conclusions

Motivations

- Challenges of LHC, and HL-LHC are pushing to **re-think the HEP computing models**

Impact on several aspects, from software to the computing infrastructure



Need to:

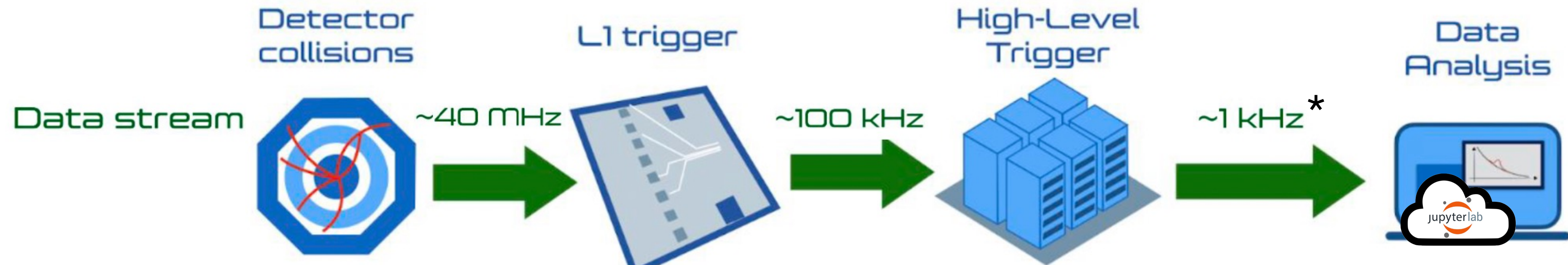
- Optimize the usage of CPU and storage
- Promote the usage of better data formats
- Develop new analysis paradigms!**
- New software based on declarative programming and interactive workflows
- Distribute on geographically separated resources

Higher rates of collision events

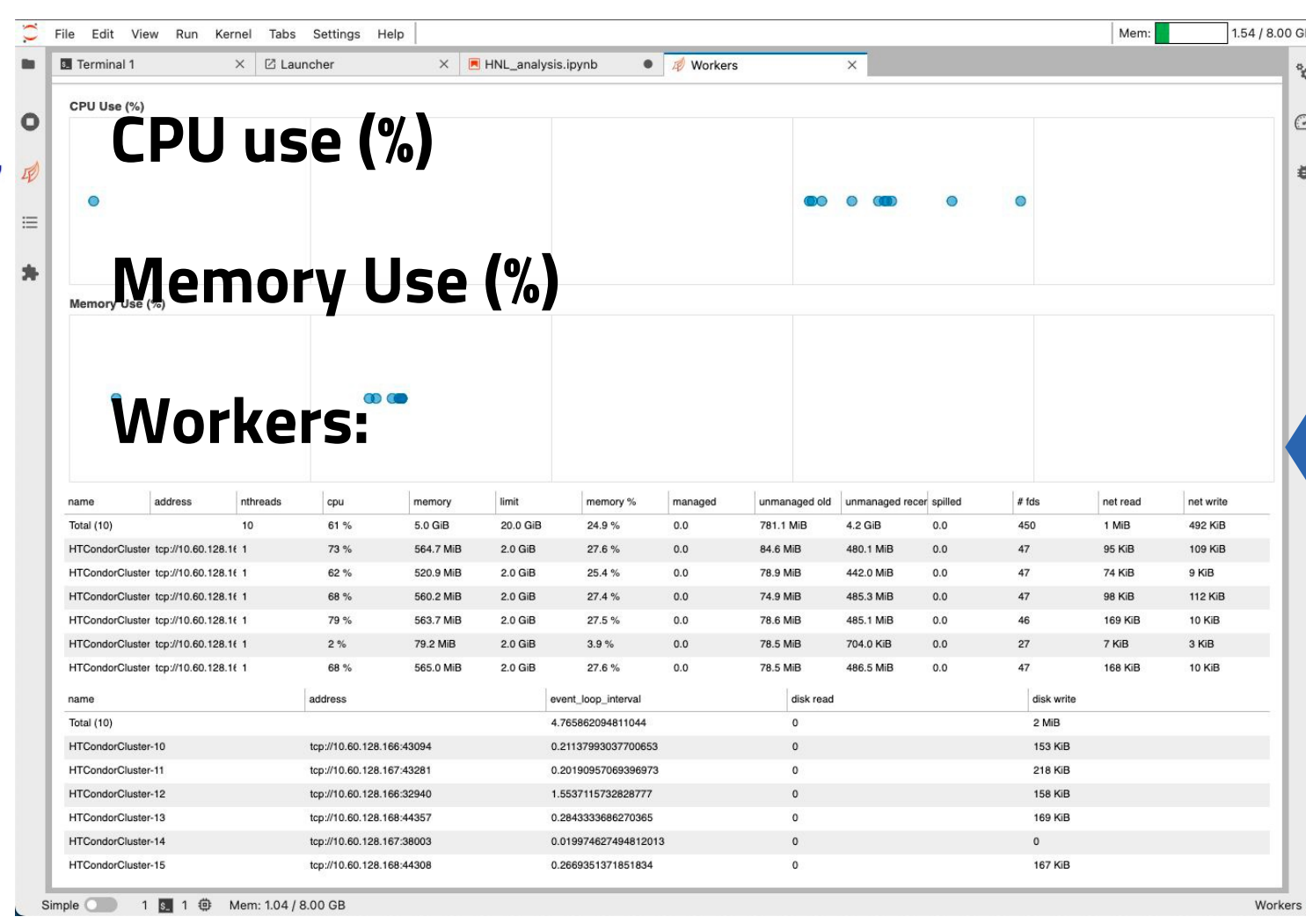


Higher demand for computing and storage resources

HEP data analysis with ICSC



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



The screenshot shows a JupyterLab terminal window with the following content:

HNL CMS Analysis

Code from Leonardo Lunelli

Data analysis code

Dask cluster configuration

NOTE: The cell below must be changed every time the Dask cluster is recreated

```
[1]: from dask.distributed import Client
client = Client("localhost:22631")
client

/usr/local/share/miniconda/lib/python3.10/site-packages/distributed/client.py:1309: VersionMismatchWarning: Mismatched versions found
| Package | Client | Scheduler | Workers |
| l24 | 4.0.0 | None | 4.0.0 |
| msgpack | 1.0.3 | 1.0.5 | 1.0.3 |
| python | 3.10.10.final.0 | 3.9.9.final.0 | 3.10.10.final.0 |
| toolz | 0.12.0 | 0.11.1 | 0.12.0 |

Notes:
- msgpack: Variation is ok, as long as everything is above 0.6
warnings.warn(version_module.VersionMismatchWarning(msg["warning"]))
```

Client: Client-ce7539b8-e288-11ed-81dd-7a36feca5287

Connection method: Direct

Dashboard: <http://localhost:31645/status>

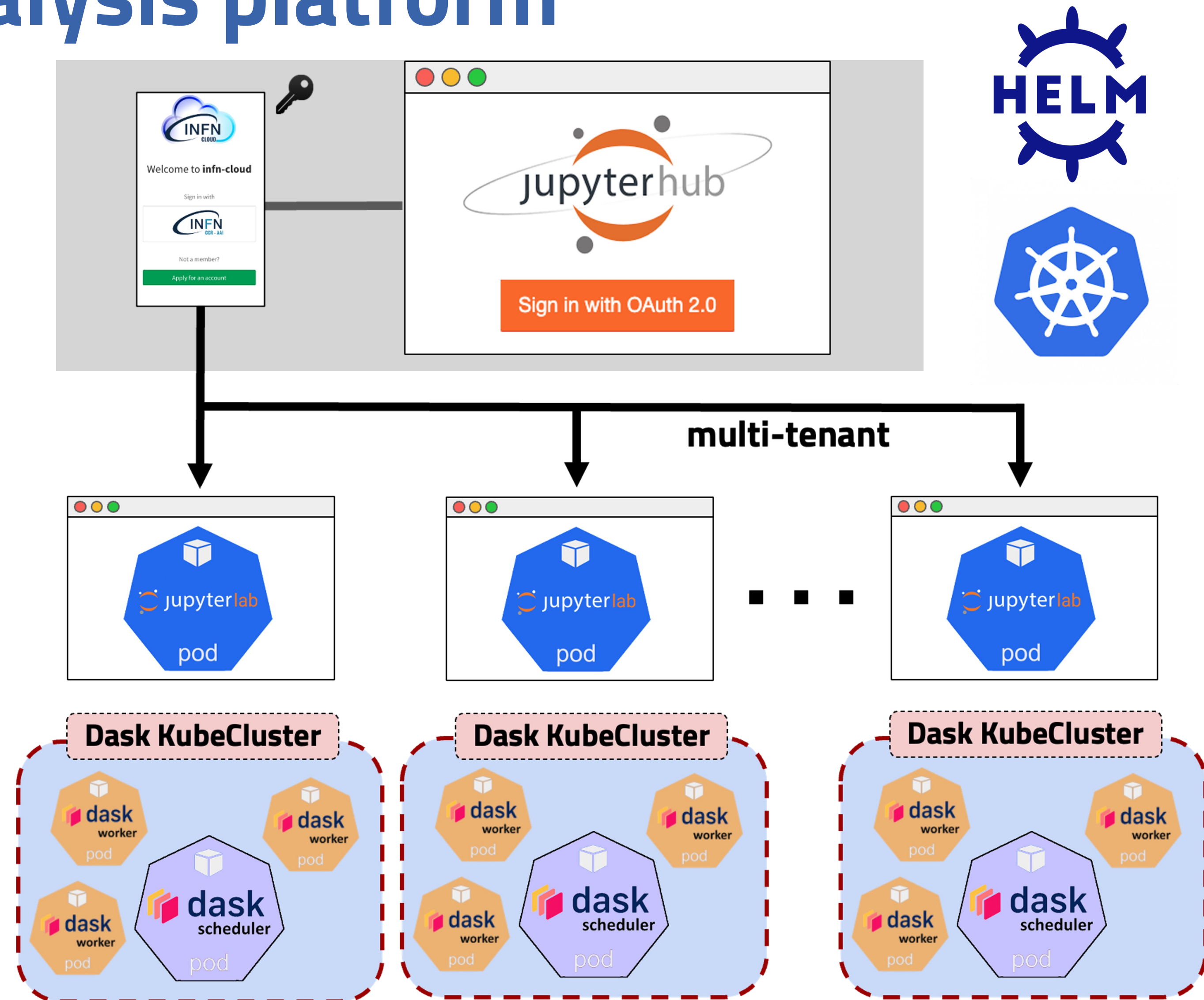
Scheduler Info

Scheduler

*trigger rates for previous Runs, now factor 3 ÷ 5 higher, will further scale in HL-LHC

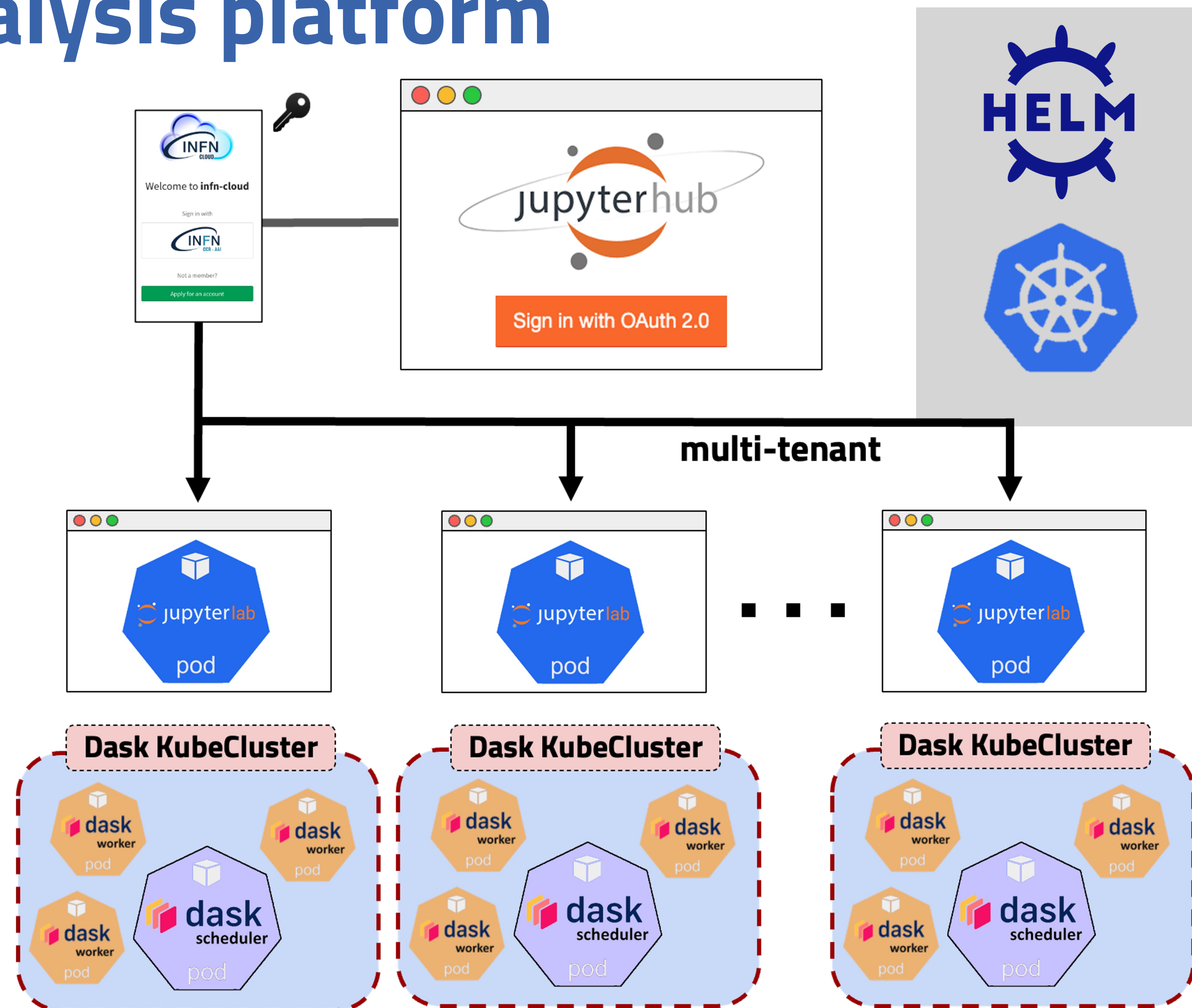
High throughput data analysis platform

- After connecting to an endpoint 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.
- The jupyterhub is deployed on a Kubernetes (k8s) cluster with **128 vCPUs and 258 GB**, divided into 8 nodes configured via [RKE2](#)



High throughput data analysis platform

- The deployment of the Kubernetes resources is handled via HELM charts in the official [Spoke2 Jhub HELM repo](#)
- This allows for a scalable and fault-tolerant deployment of the available resources



High throughput data analysis platform

- Jupyterlab interface is flexible and customizable:
 - Includes specific plugins (e.g. [Dask](#))
 - Working environment highly customizable using [Docker](#) containers allowing for experiment specific software

```

[4]: from dask.distributed import Client
      client = Client("tcp://dask-root-c1d75b3b-c-scheduler.jhub:8786")
      client
    
```

Client
Client-17e349cd-8980-11ef-90ac-4e37bcd44ce1
Connection method: Direct
Dashboard: <http://dask-root-c1d75b3b-c-scheduler.jhub:8787/status>

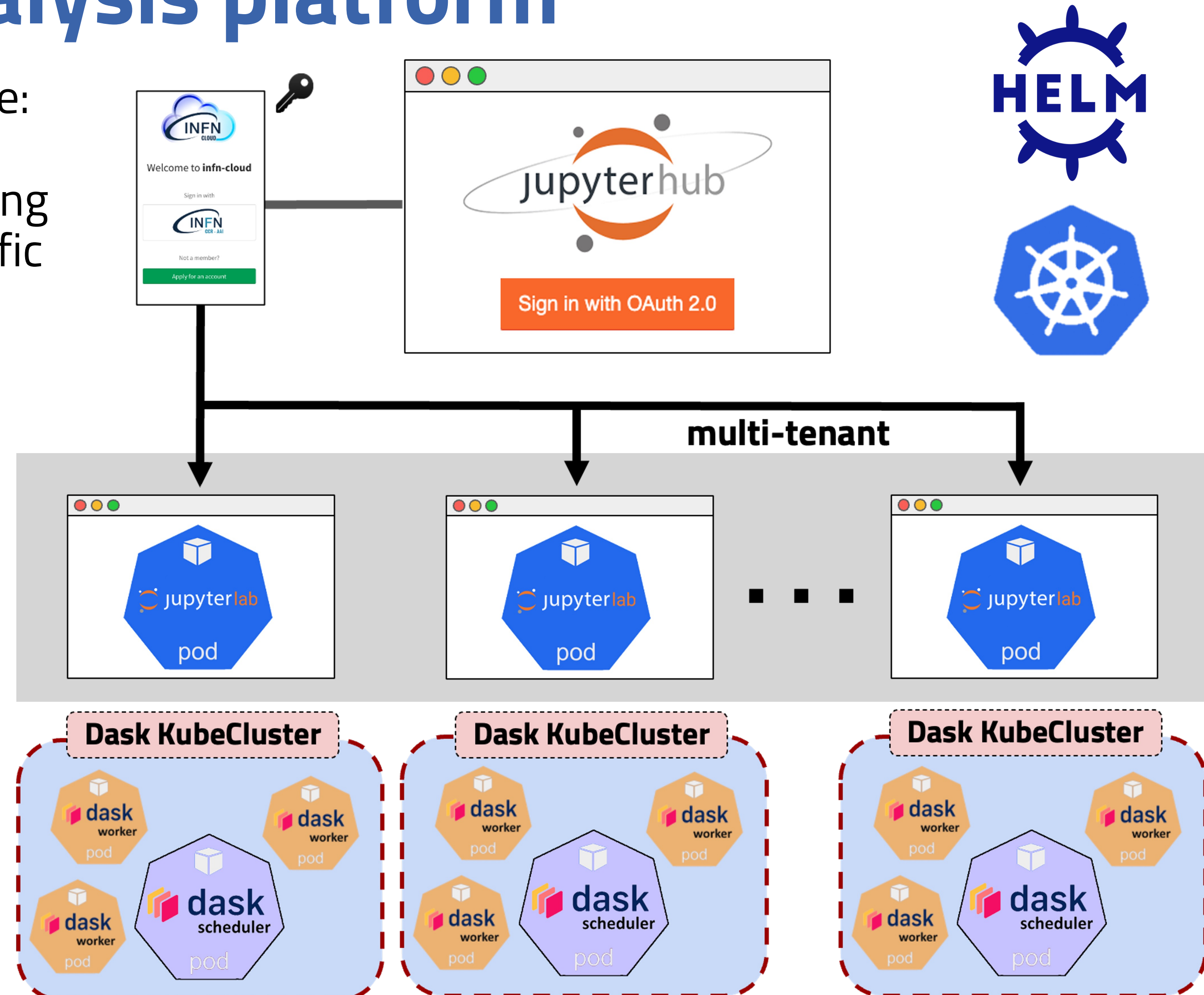
Scheduler Info

Scheduler
Scheduler-e626a51e-74cc-4c92-b481-6a821940c462
Comm: tcp://10.42.6.73:8786
Dashboard: <http://10.42.6.73:8787/status>
Started: 20 minutes ago

Workers: 40
Total threads: 40
Total memory: 80.00 GiB

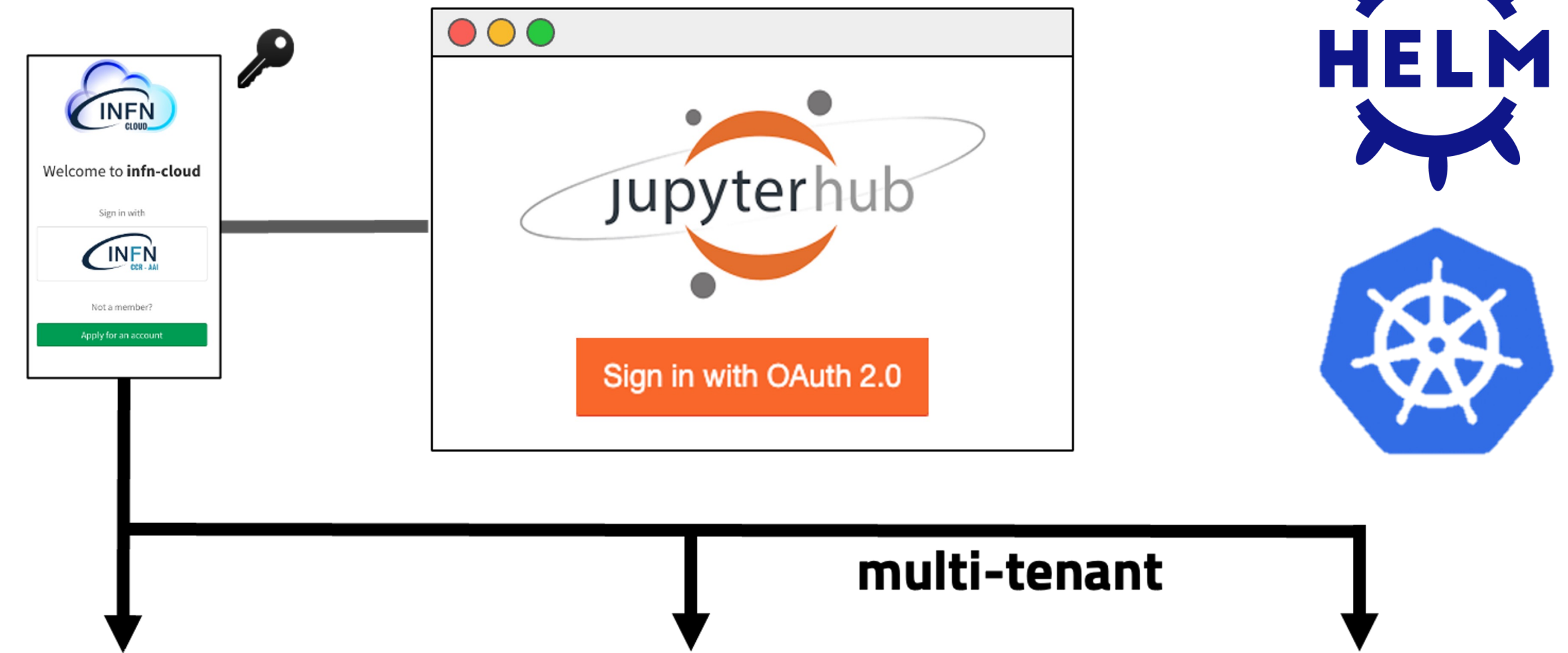
Workers

- Worker: dask-root-c1d75b3b-c-default-worker-00d0a343d6
- Worker: dask-root-c1d75b3b-c-default-worker-0435a98c83
- Worker: dask-root-c1d75b3b-c-default-worker-046ae5f895
- Worker: dask-root-c1d75b3b-c-default-worker-0739901384
- Worker: dask-root-c1d75b3b-c-default-worker-0bbcab7e3d
- Worker: dask-root-c1d75b3b-c-default-worker-16932c5f0b
- Worker: dask-root-c1d75b3b-c-default-worker-256fa4e72a
- Worker: dask-root-c1d75b3b-c-default-worker-3bf9267d55



High throughput data analysis platform

- Ideal environment for testing interactive analysis and validating new frameworks, e.g. the multi-threading features of ROOT RDataFrame
- The [Dask Labextension](#) provides a user-friendly monitoring dashboard
- More in the [official docs!](#)



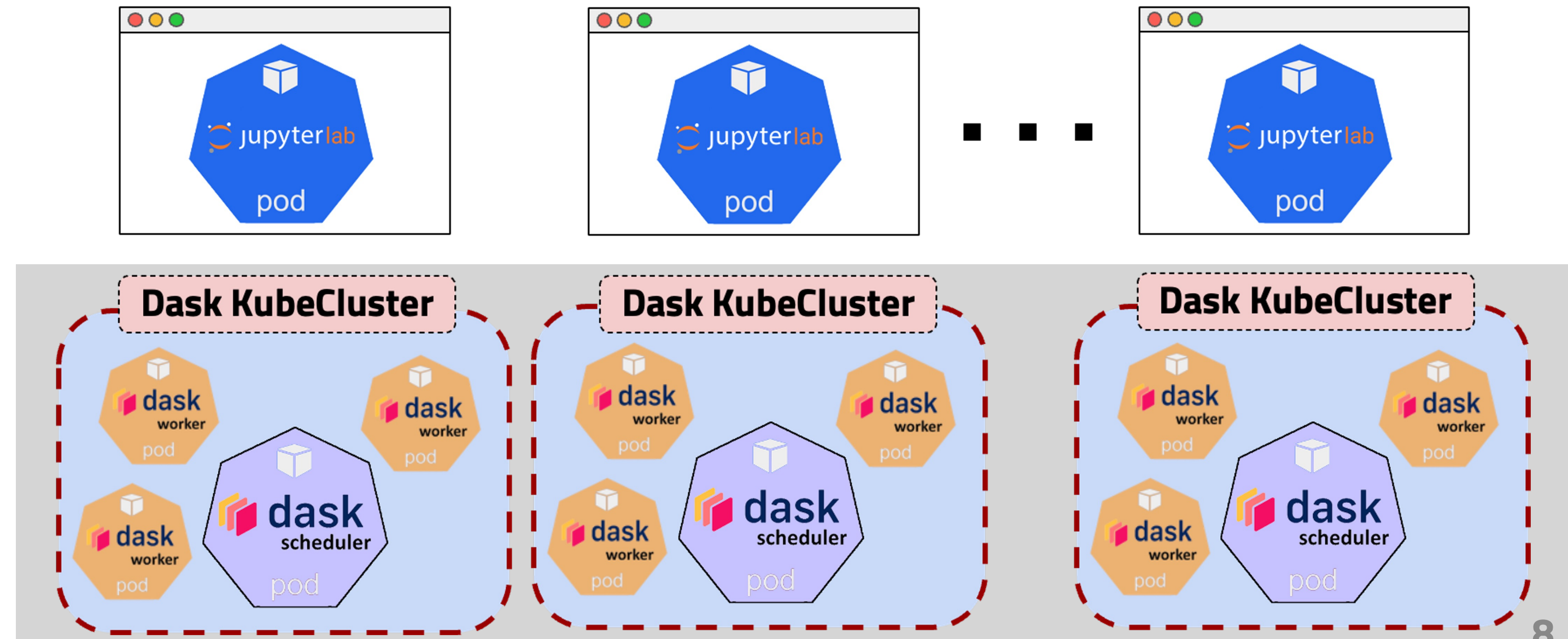
Dask Dashboard

1 **Monitoring workers**

2 **Cluster map**

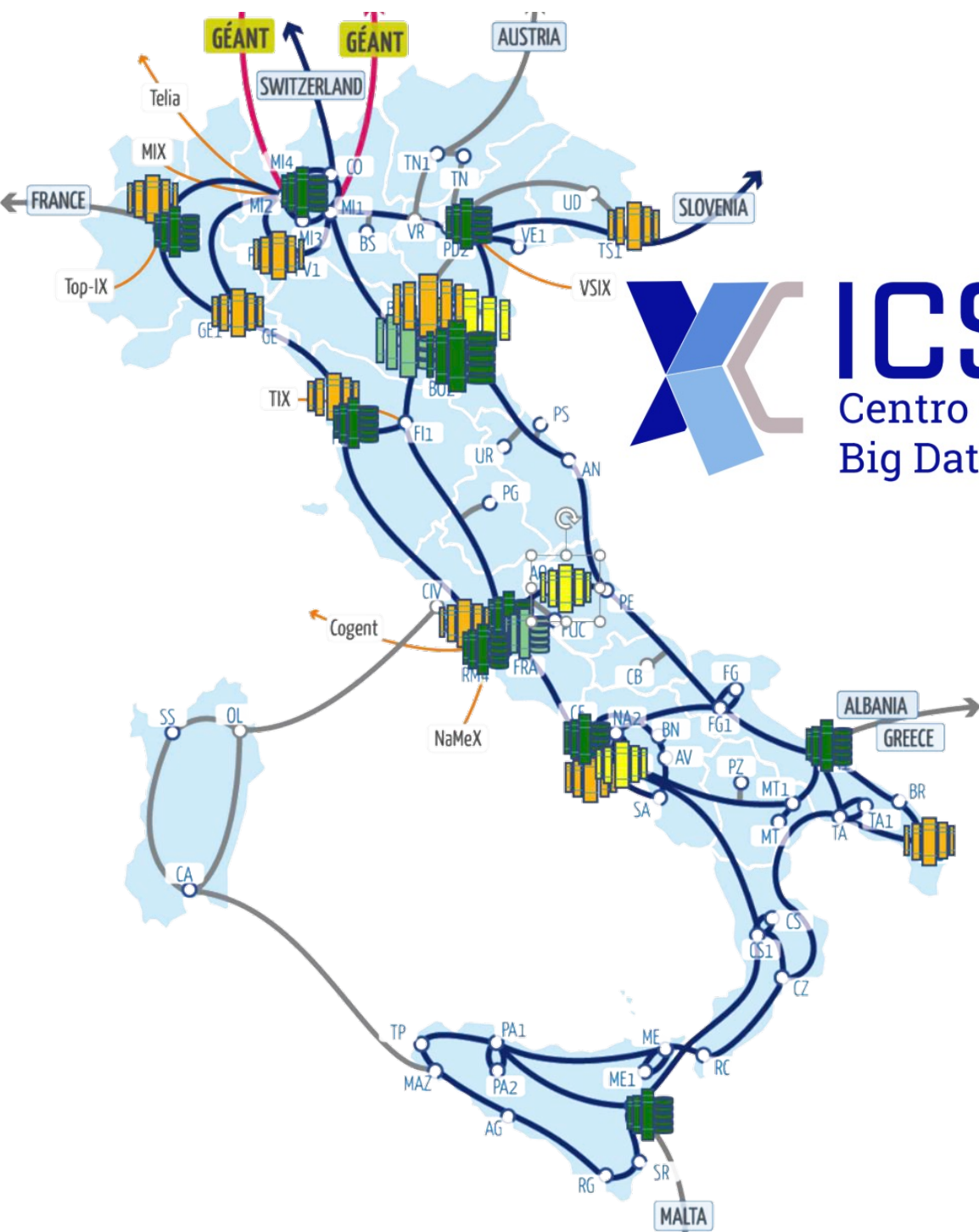
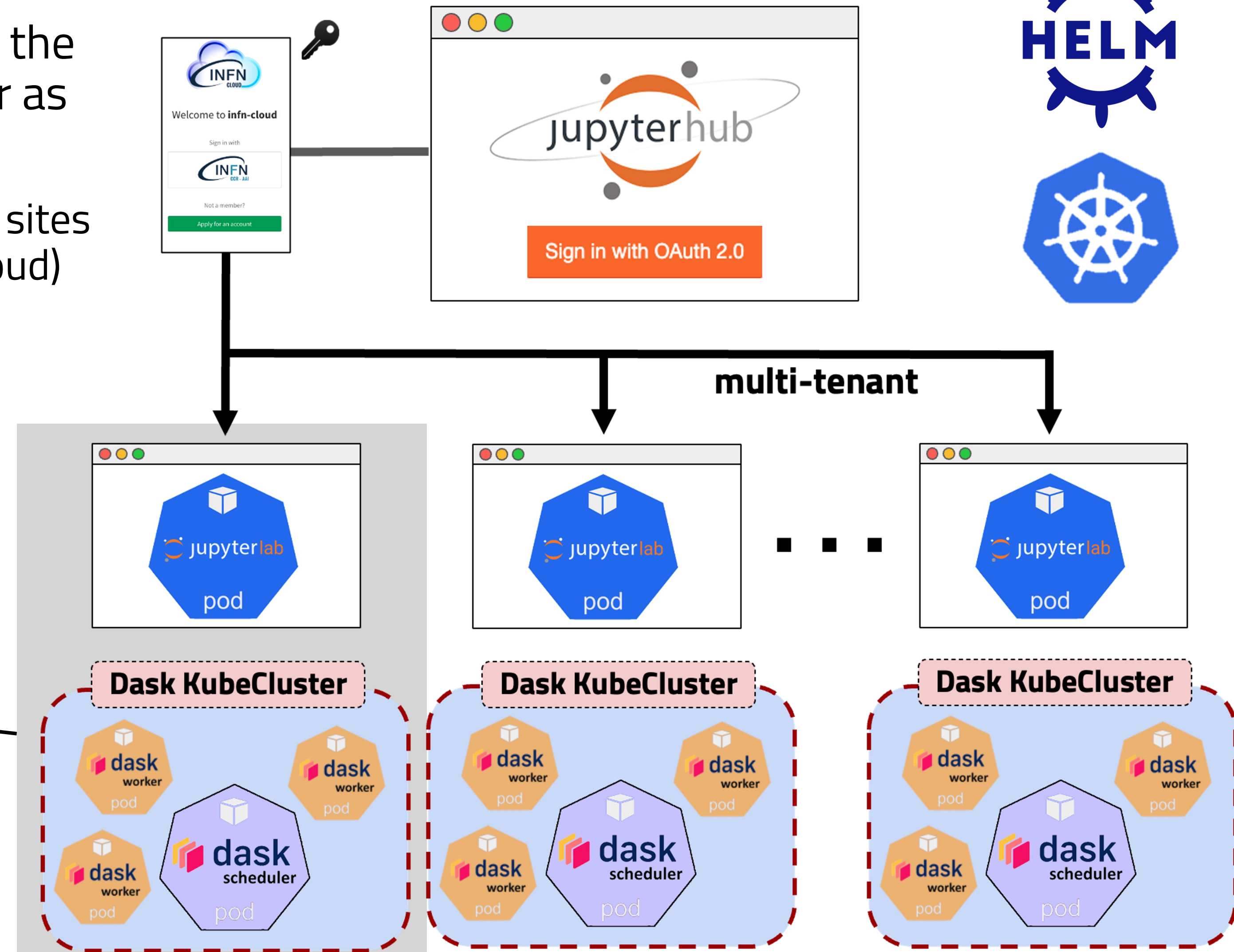
The screenshot shows the Dask Dashboard interface. On the left, there's a sidebar with various metrics like 'AGGREGATE TIME PER ACTION', 'BANDWIDTH TYPES', 'CPU', 'GPU MEMORY', etc. The main area is split into two panels: 'Monitoring workers' with a table of worker details and 'Cluster map' showing a visual representation of the cluster with 'worker' and 'scheduler' nodes.

name	address	nthreads	cpu	memory	limit	memory / managed	unmanaged	unmanaged / spilled	if fs	net read	net write	disk read	disk write
Total (0)	5	5%	731.7 MB	30.0 GB	7.1 %	0.0	730.9 MB	844.0 KB	0.0	110	1 KB	7 KB	0
dask-worker-knp123.v-1	1	6%	144.3 MB	2.0 GB	7.2%	0.0	144.3 MB	160.0 KB	0.0	22	286 B	1 KB	0
dask-worker-knp123.v-1	1	4%	145.3 MB	2.0 GB	7.0%	0.0	145.3 MB	160.0 KB	0.0	22	286 B	1 KB	0
dask-worker-knp123.v-1	1	6%	146.5 MB	2.0 GB	7.2%	0.0	146.5 MB	160.0 KB	0.0	22	286 B	1 KB	0
dask-worker-knp123.v-1	1	4%	148.7 MB	2.0 GB	7.3%	0.0	148.8 MB	160.0 KB	0.0	22	286 B	1 KB	0
dask-worker-knp123.v-1	1	6%	145.8 MB	2.0 GB	7.1%	0.0	145.7 MB	160.0 KB	0.0	22	286 B	1 KB	0



High throughput data analysis platform

- Offloading strategy: resources used to offload the computation are hosted in the same k8s cluster as the jupyter interface, via DASK KubeCluster
- **Under development:** spawning on multiple remote sites allowing for heterogeneous resources (HTC/HPC/Cloud) (see more in backup)



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

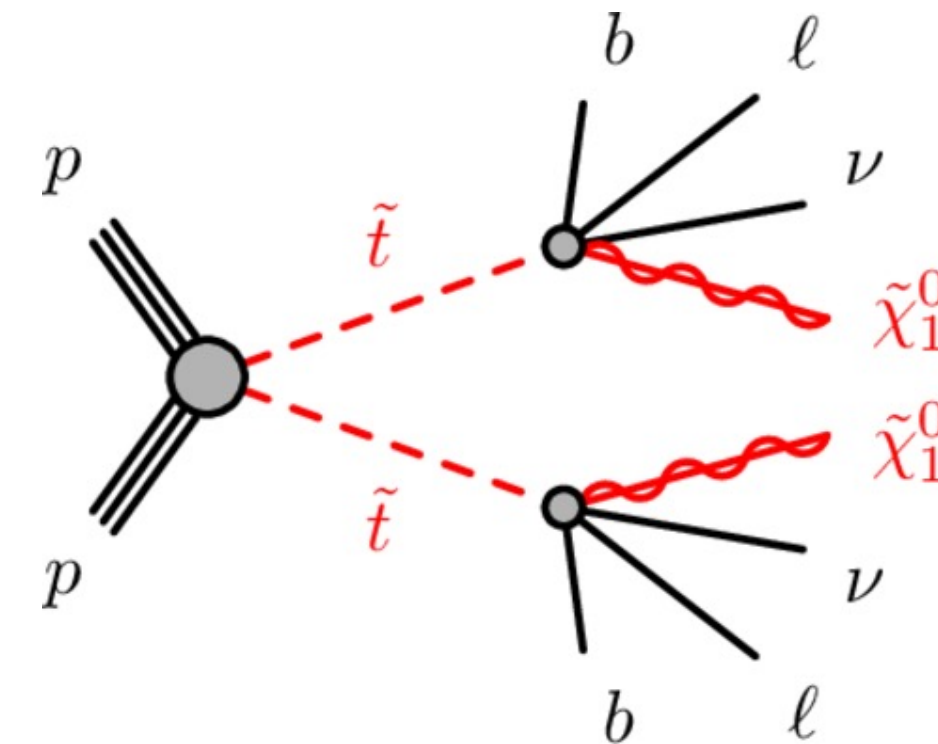
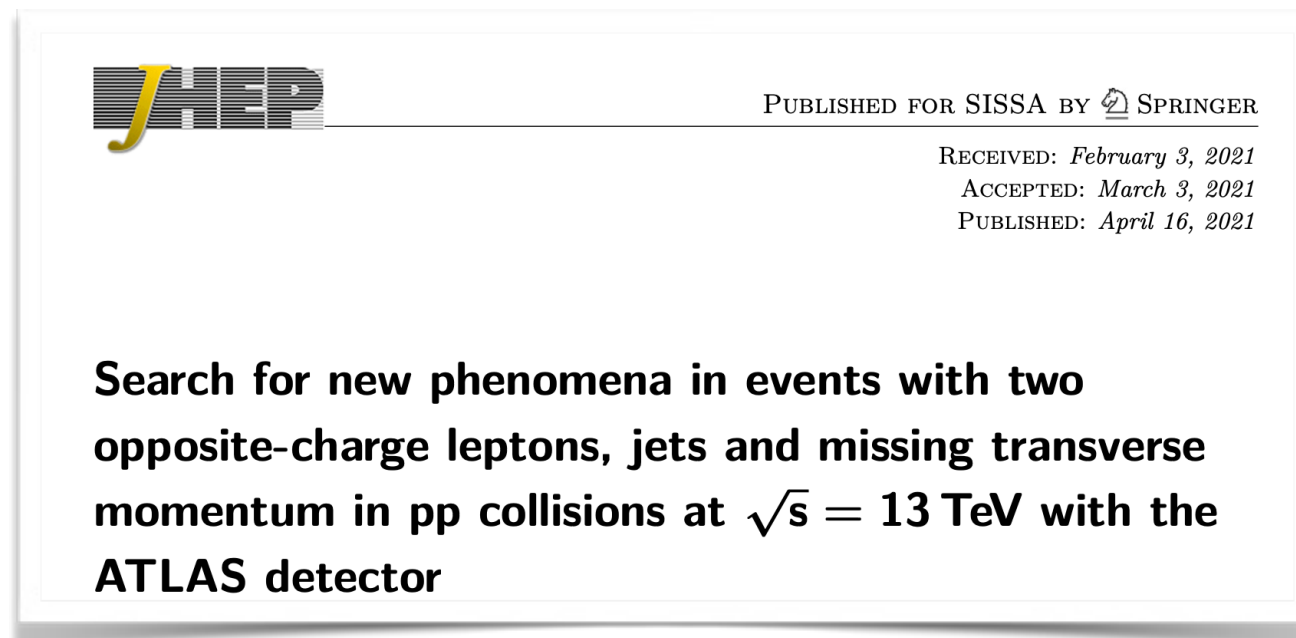
Offloading to..

The background is a deep blue gradient. On the left side, there is a complex pattern of light trails and dots. These trails are composed of many thin, parallel lines that curve and converge towards the center, creating a sense of depth and movement. The dots are small, bright blue spheres of varying sizes, some appearing as individual points and others as part of the light trails. The overall effect is reminiscent of a digital data stream or a futuristic tunnel.

Benchmark interactive analyses

ATLAS use-case

SUPERSymmetry: Beyond Standard Model (BSM) theory



Soft leptons coming from a virtual W^* boson decay

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

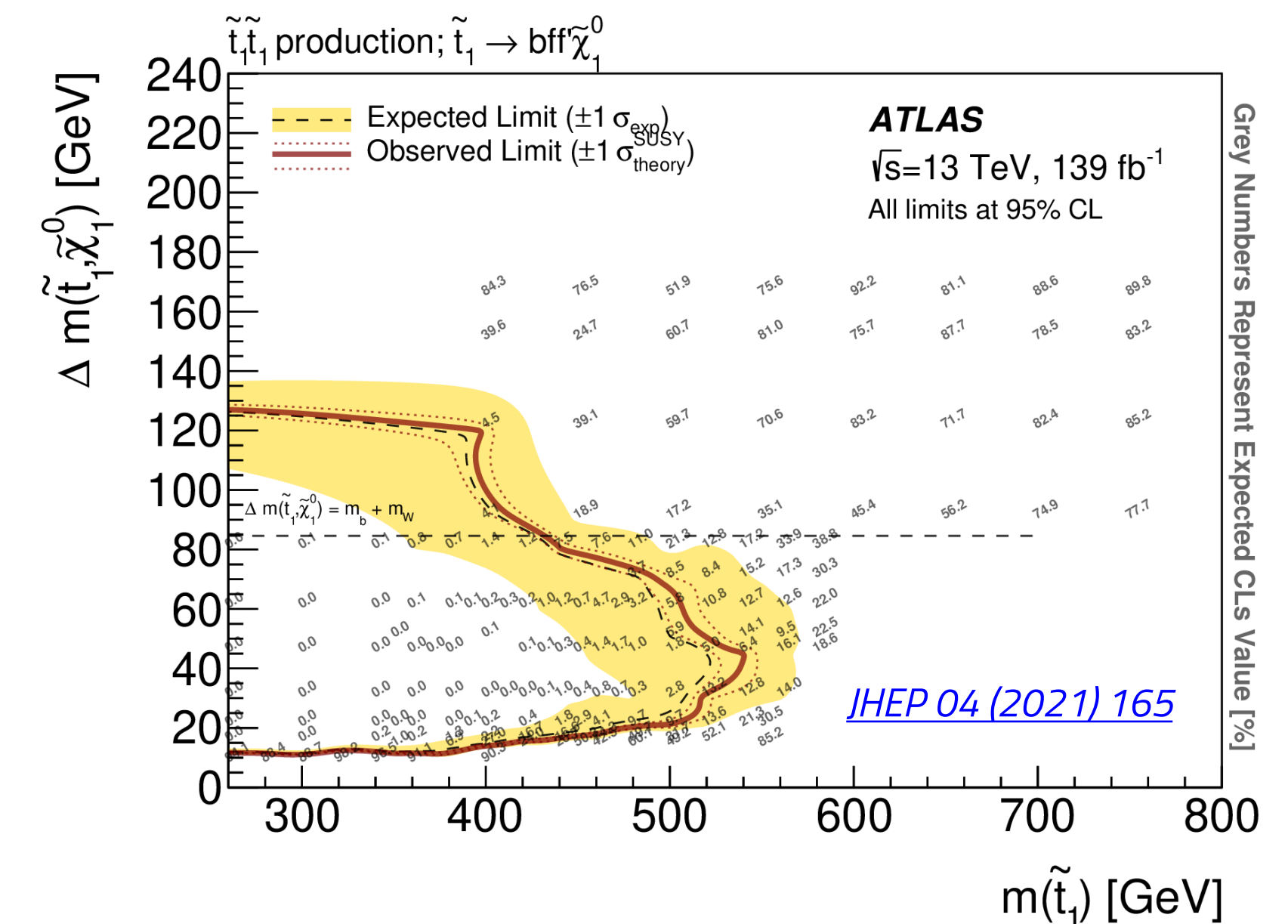
- Three different analysis in the [Run 2 paper](#), already published, according to mass splitting between *stop* (\tilde{t}_1) and *neutralino* ($\tilde{\chi}_1^0$), allowing different decay modes:

- 2 body $\rightarrow \Delta m > m_t$
- 3 body $\rightarrow m_W + m_b < \Delta m < m_t$

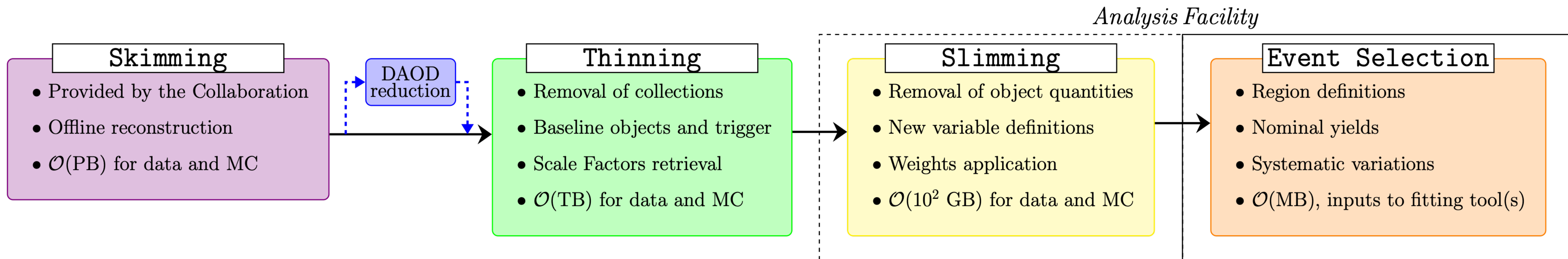
• 4 body $\rightarrow \Delta m < m_W + m_b$ **used as a benchmark**

- Common final state signature: 2 OS leptons from W^* decays, b-jets and missing transverse energy

- Cut-based analysis

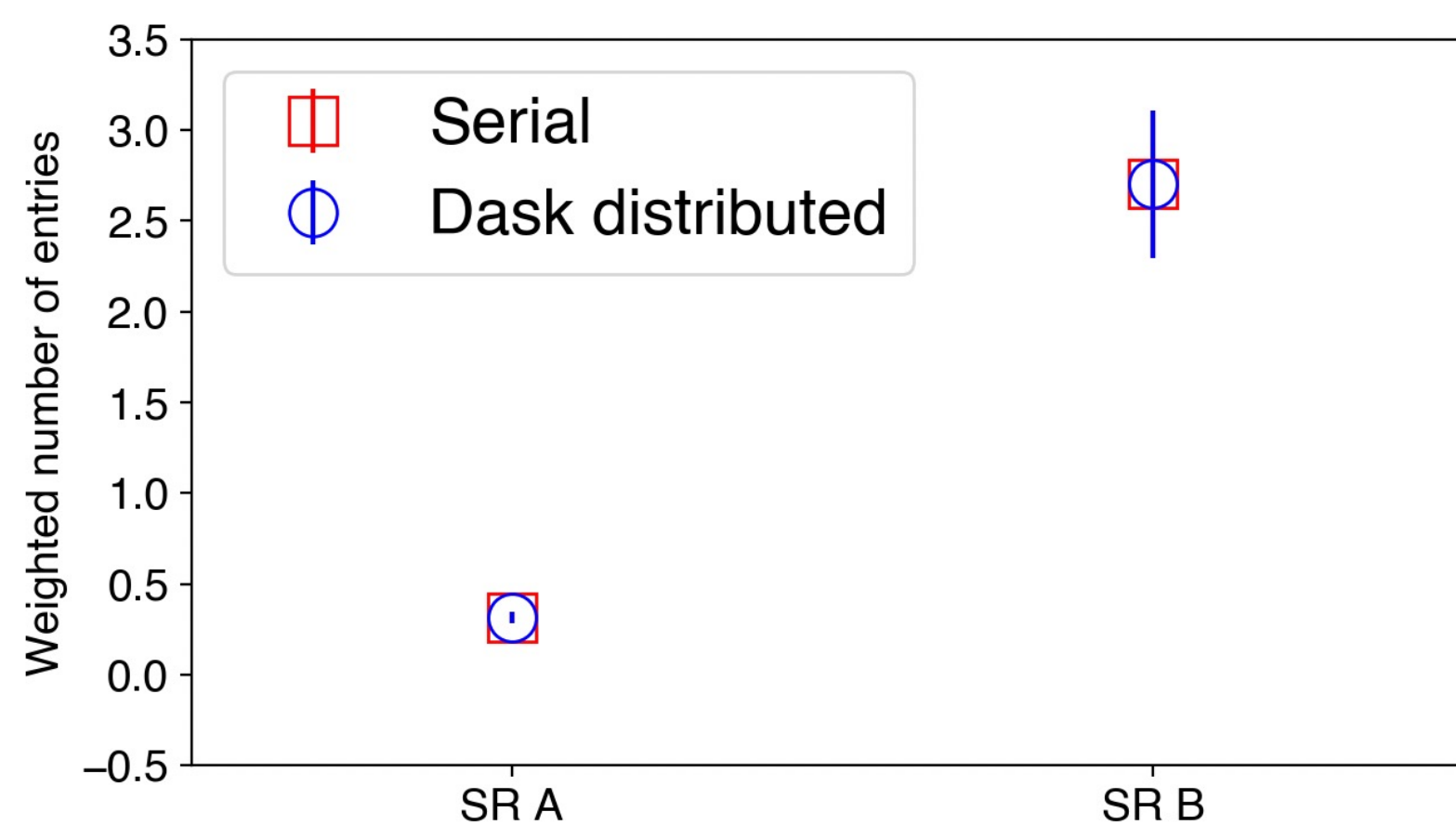


4-body search workflow



Sanity check

Weighted number of events in the Wt background sample, after the event selection cuts in signal regions A and B, nominal case
→ identical



Slimming

ATLAS slimming code already in RDataFrame, but entirely written and compiled in C++ → Dask distributed approach was not used

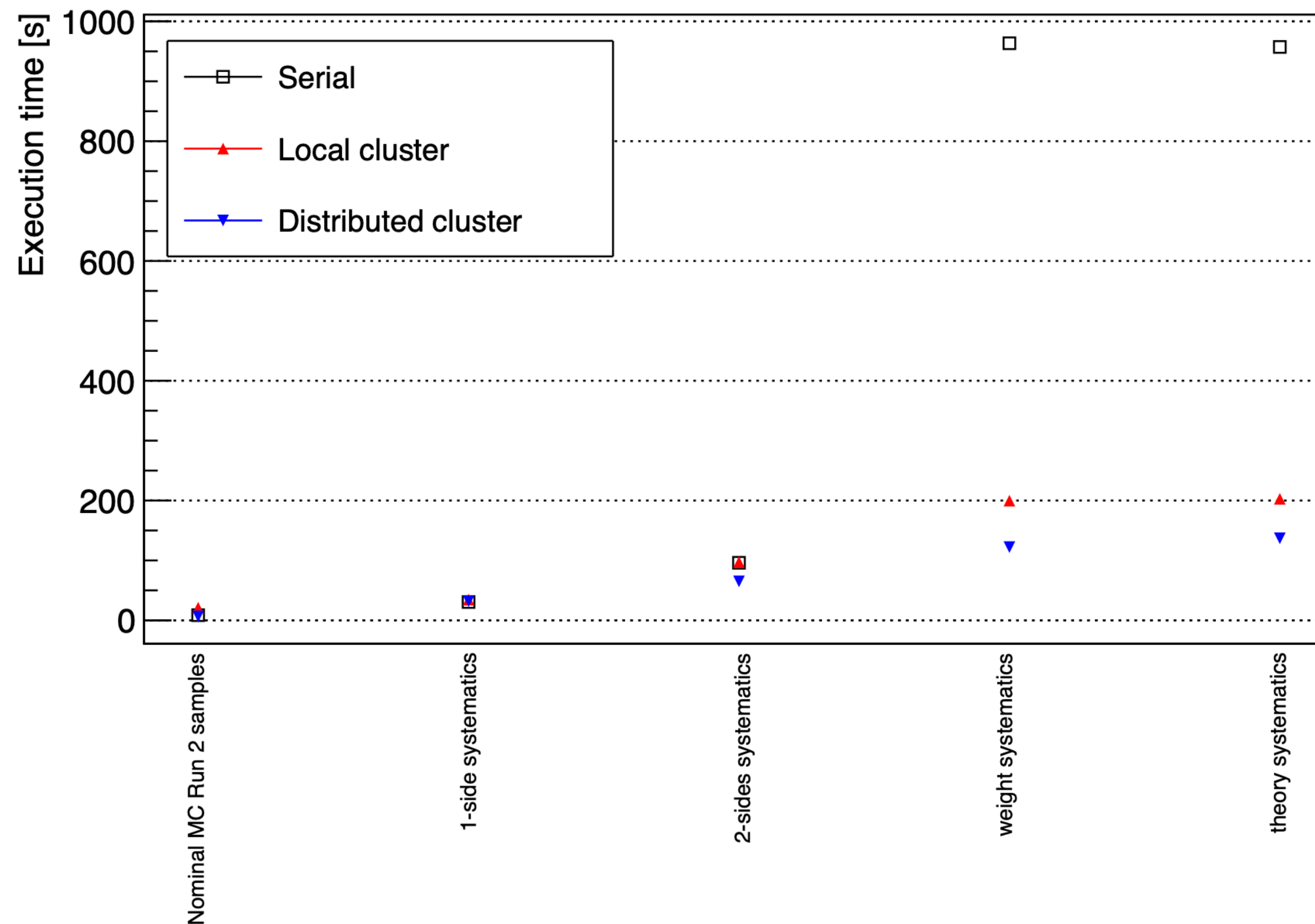
Event Selection

- Event selection for fitting tools
- RDataFrame + Dask applied to Wt bkg sample ~ 1.8 GB copied to the INFN workspace
- Playing with syst. variations
- 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

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



- LocalCluster: Dask multithread execution on local machine (max 8 cores, 16 GiB)
- Distributed: Dask distributed execution on remote workers

Scheduler and Working Nodes Reports

Distributed approach

Dask Performance Report

Select different tabs on the top for additional information

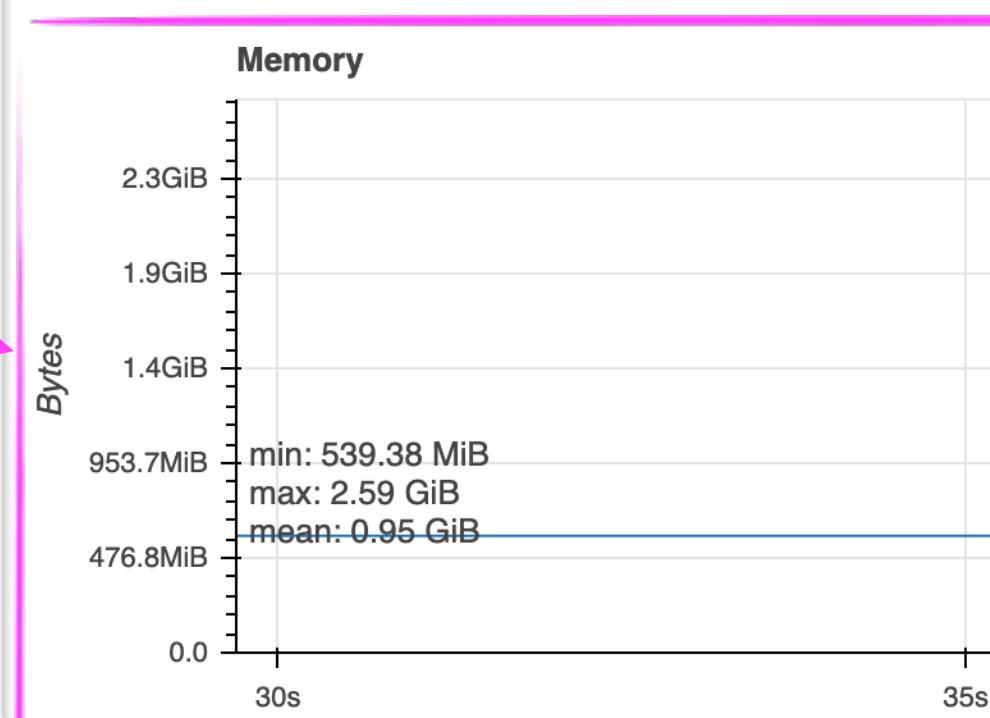
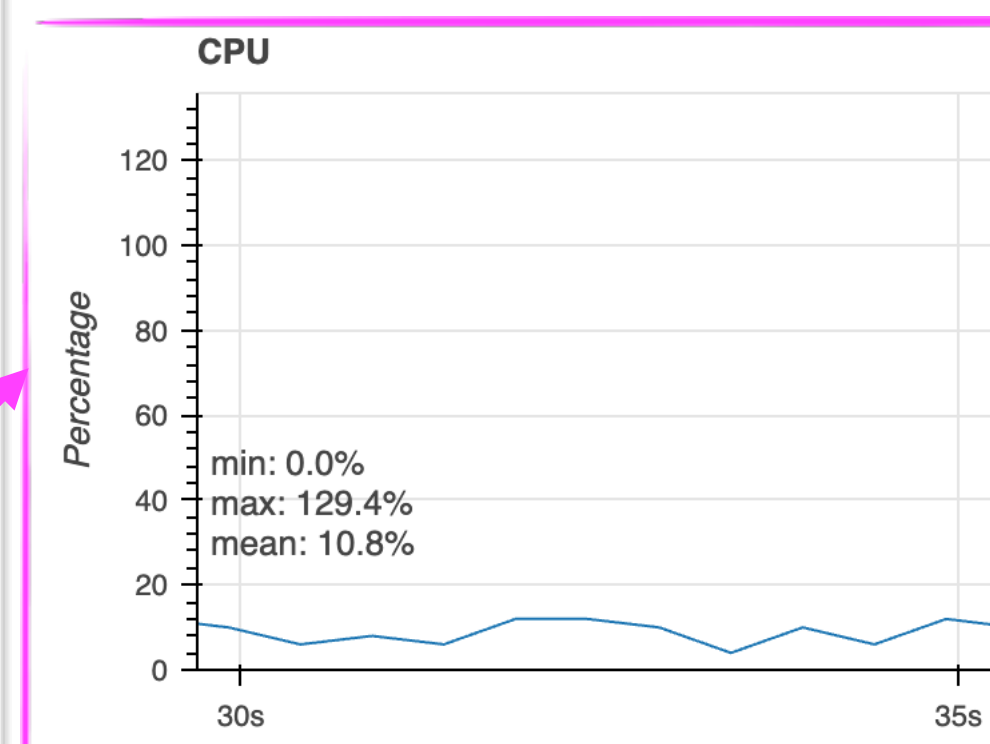
Duration: 252.87 s

Tasks Information

- number of tasks: 621
- compute time: 118.06 s
- deserialize time: 2.39 s

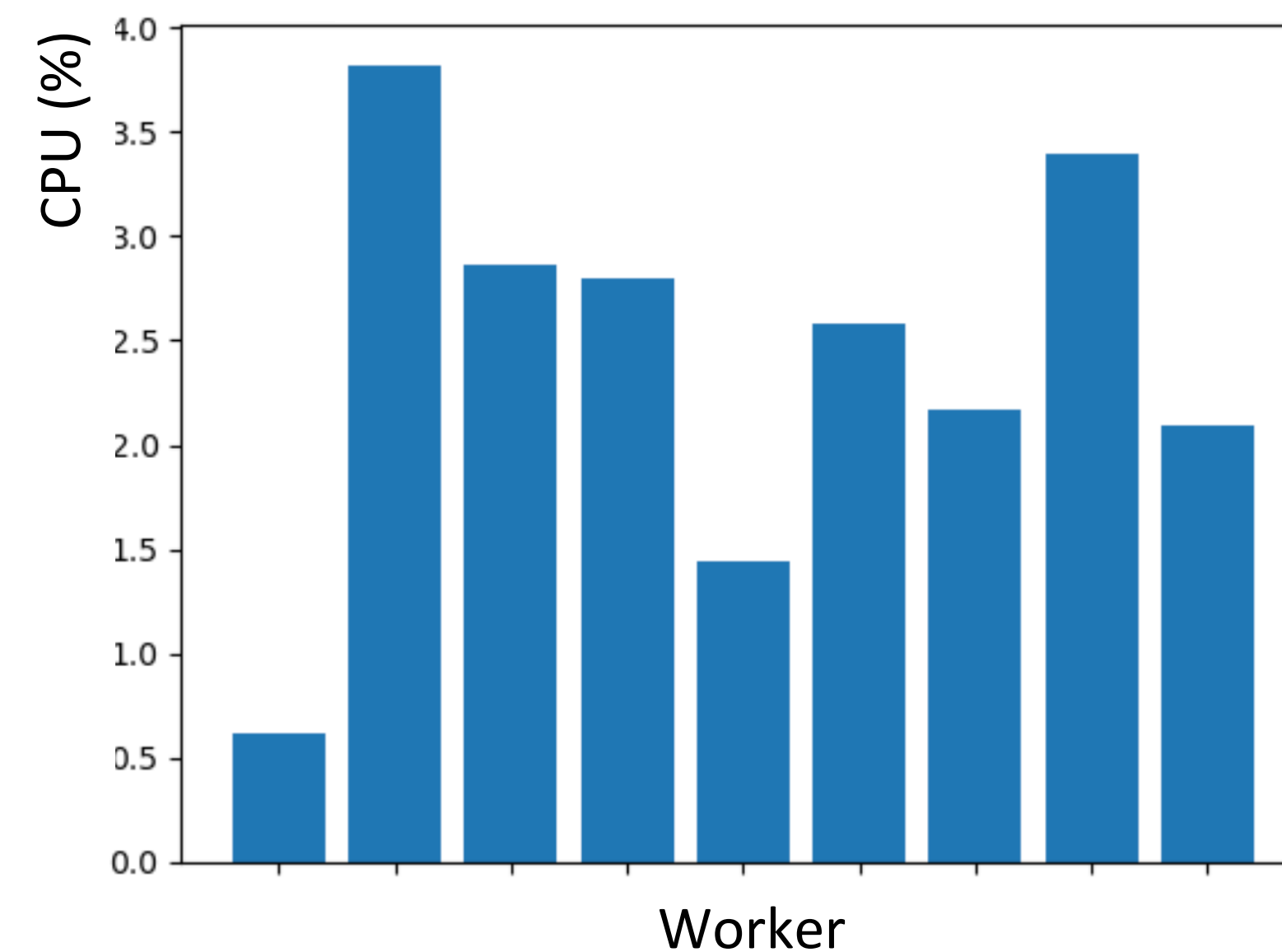
Scheduler Information

- Address: tcp://127.0.0.1:43821
- Workers: 2
- Threads: 2
- Memory: 4.39 GiB
- Dask Version: 2022.11.0
- Dask.Distributed Version: 2022.11.0



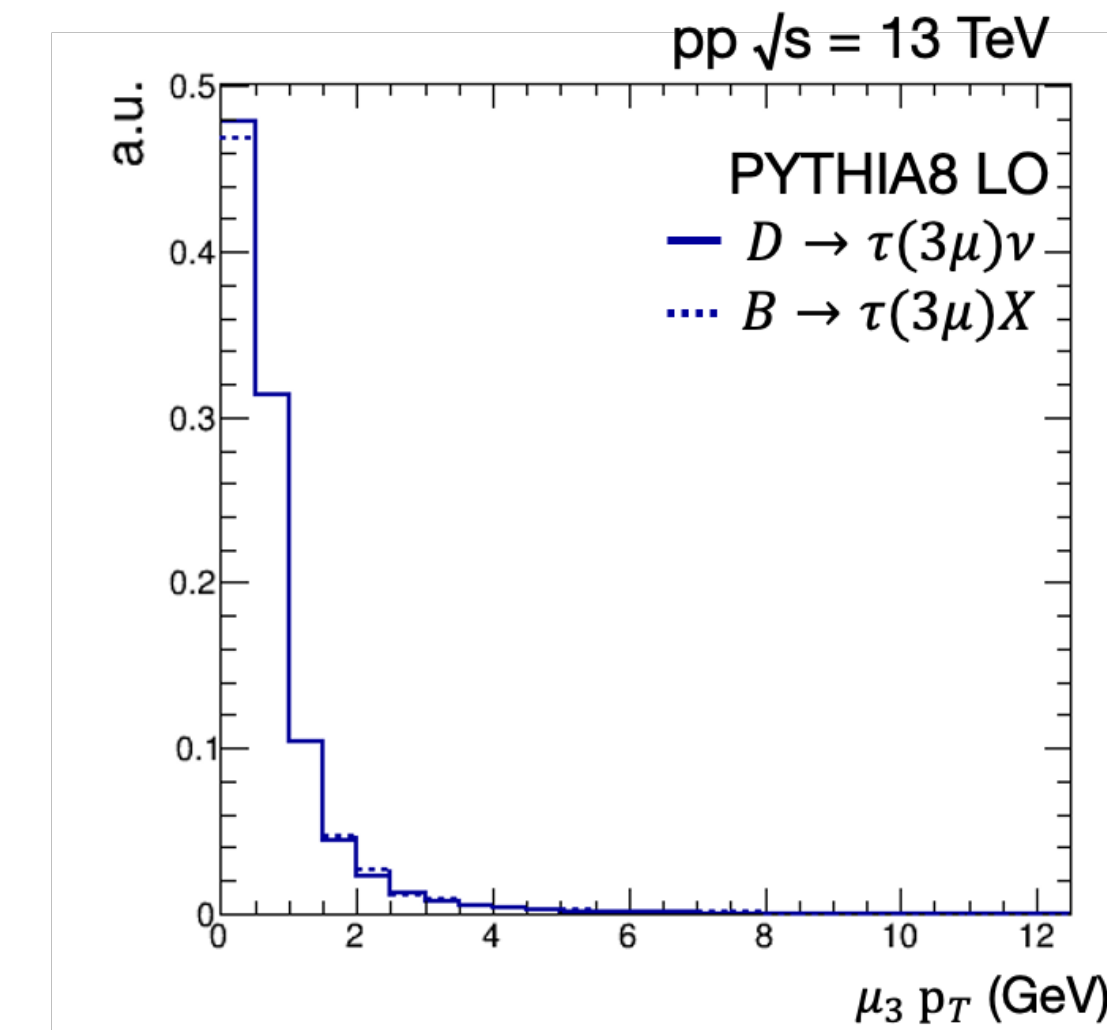
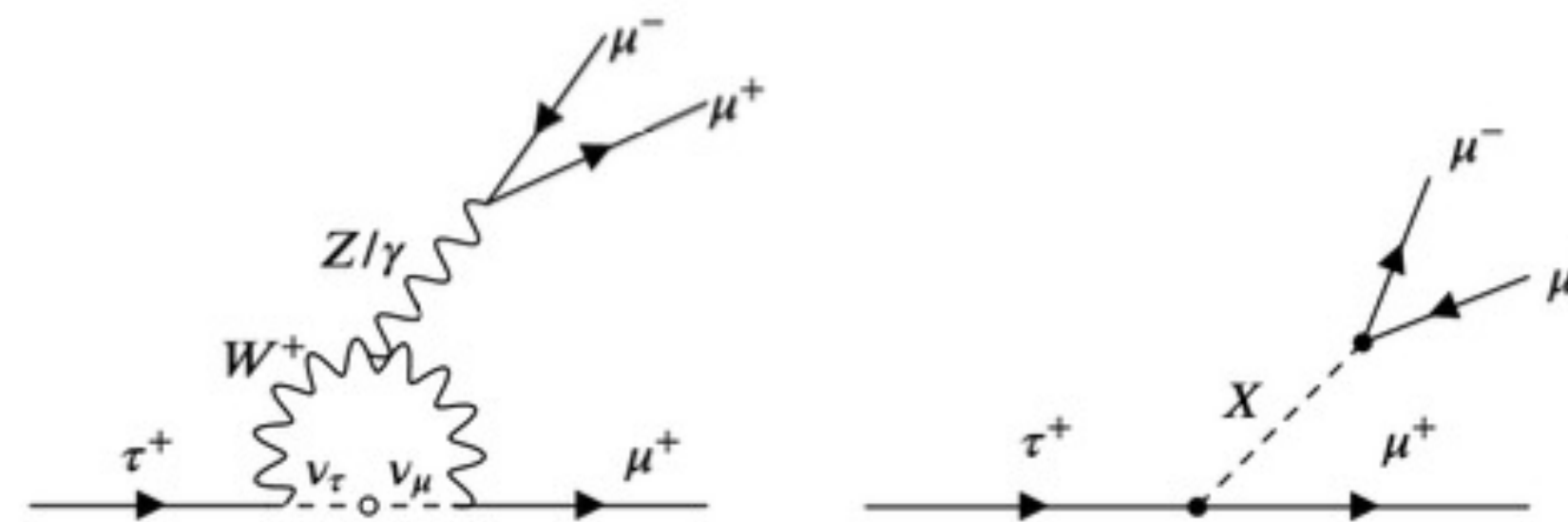
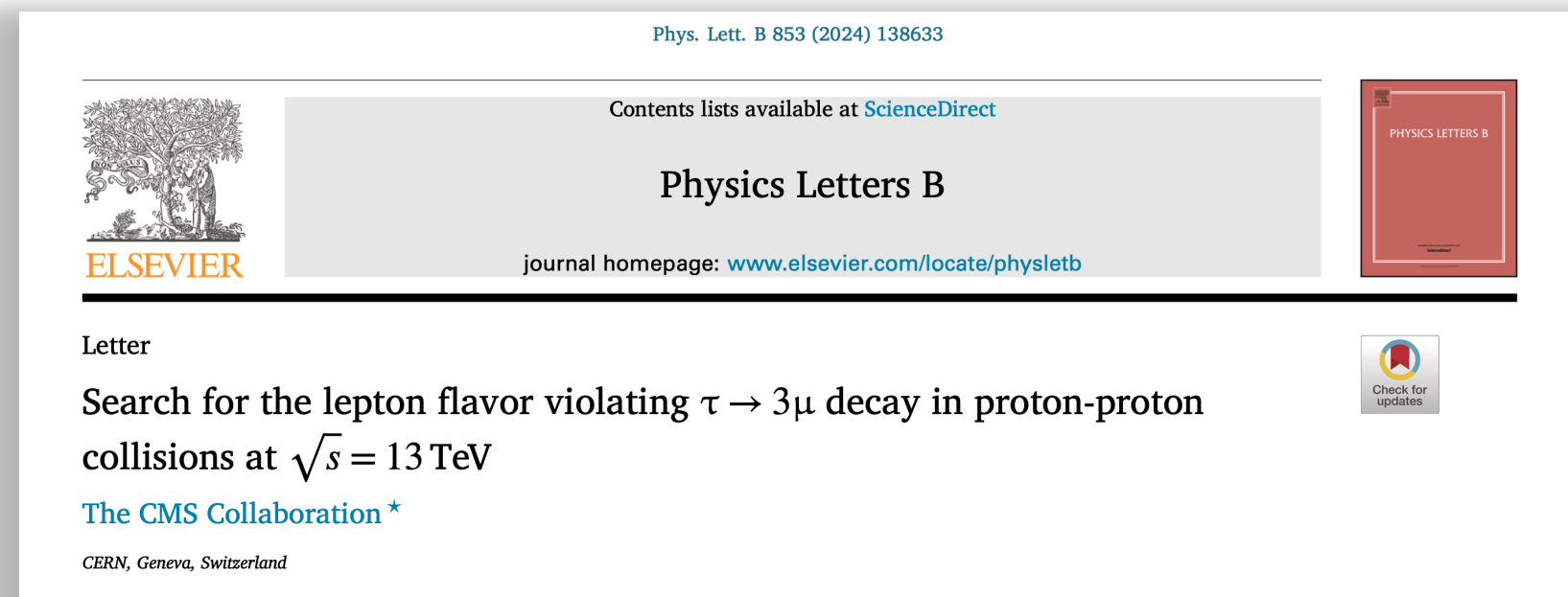
Connecting to working nodes

- Out of 9 worker nodes, we get about 4% average CPU occupancy on each worker node
- Limited CPU consumption due to the easy cut&count operations



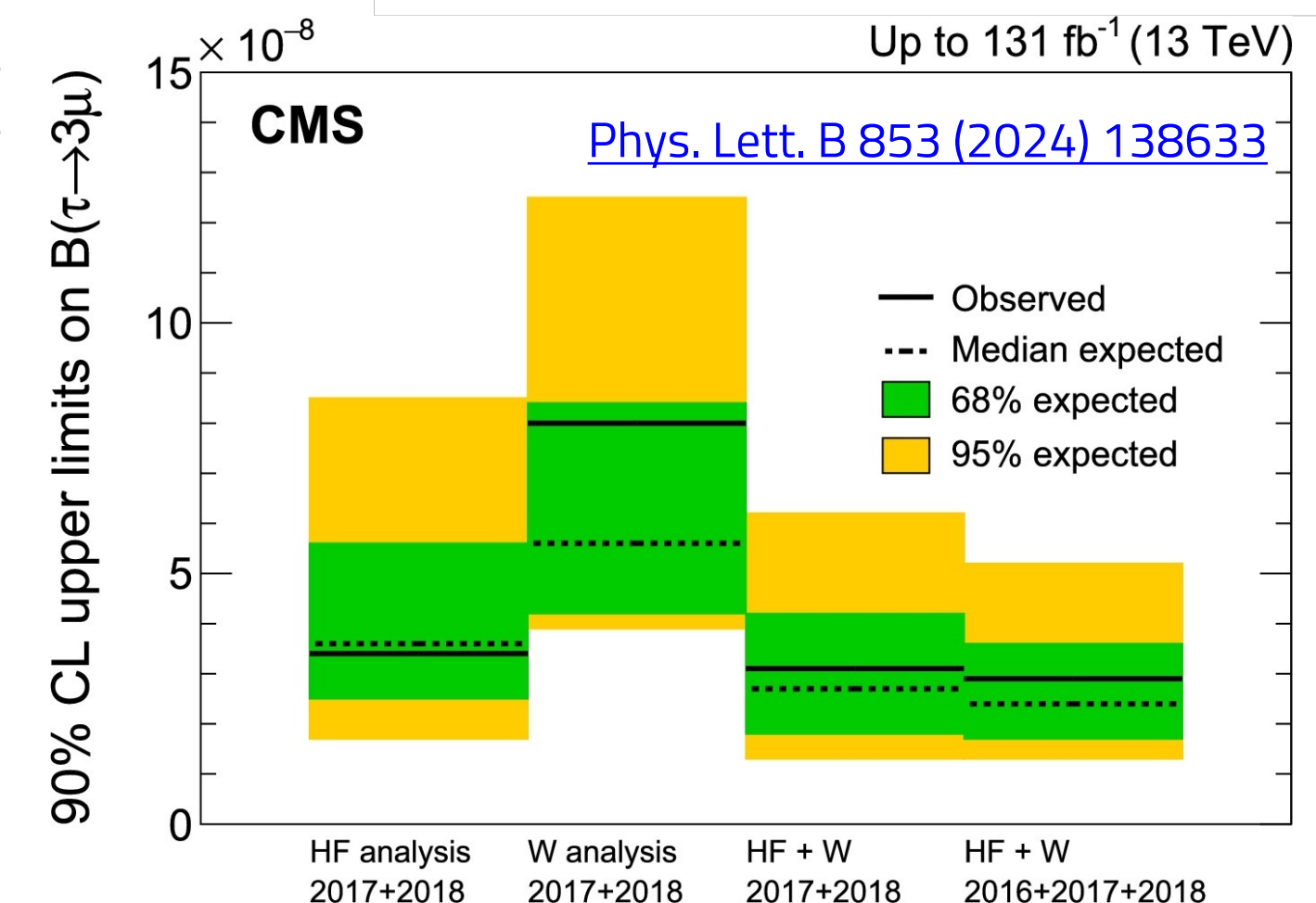
CMS use-case

Lepton Flavor Violation in the charged sector: $\tau \rightarrow 3\mu$

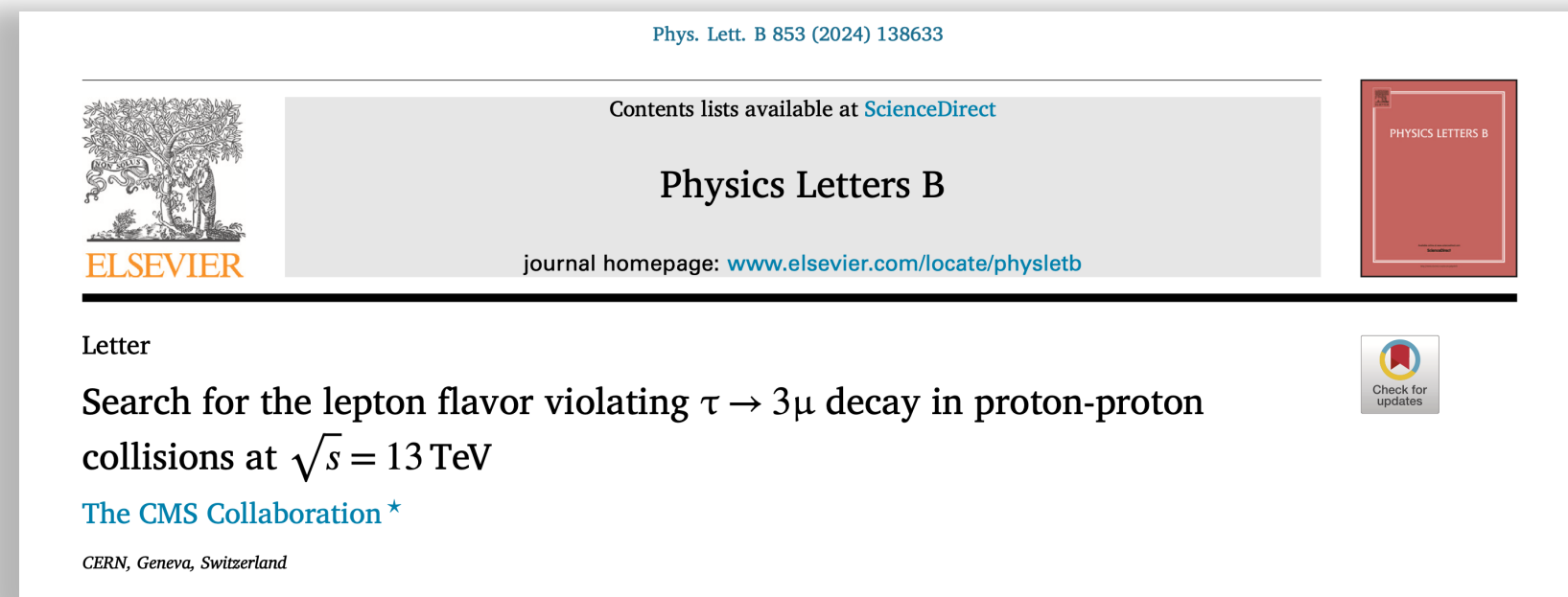


Search for $\tau \rightarrow 3\mu$ decays, which have very small SM branching fractions $BR_{SM} \sim \mathcal{O}(10^{-55})$, while being predicted with sizable BR in several BSM scenarios $BR_{BSM} \sim \mathcal{O}(10^{-10} \div 10^{-8})$

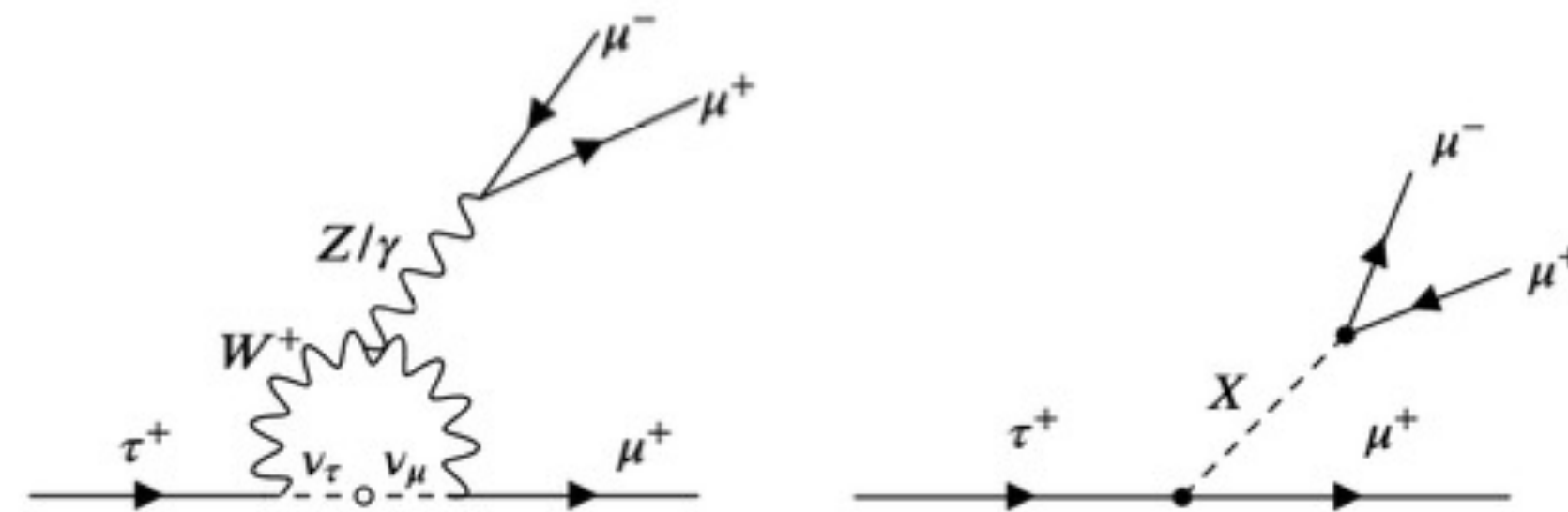
- τ leptons produced in D and B meson decays provide large statistics at LHC experiments, but are only accessible with **low- p_T muon triggers**
- Analysis of Run 2 data recently published, **stat. limited**
 - benefitting from inclusive low- p_T muon L1 trigger in **Run 3**
 - technical challenge: **new datasets are x3 times heavier**



CMS use-case



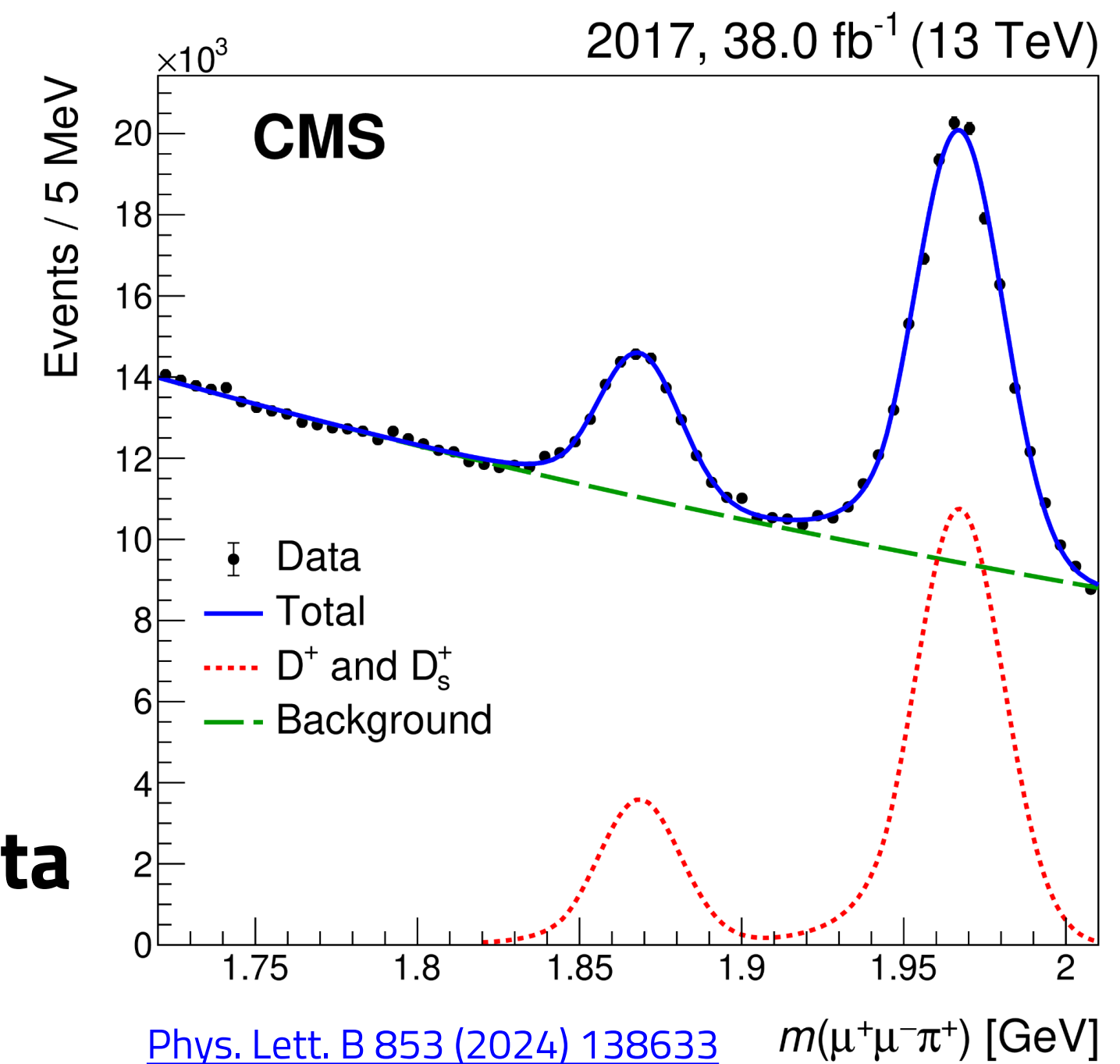
Lepton Flavor Violation in the charged sector: $\tau \rightarrow 3\mu$



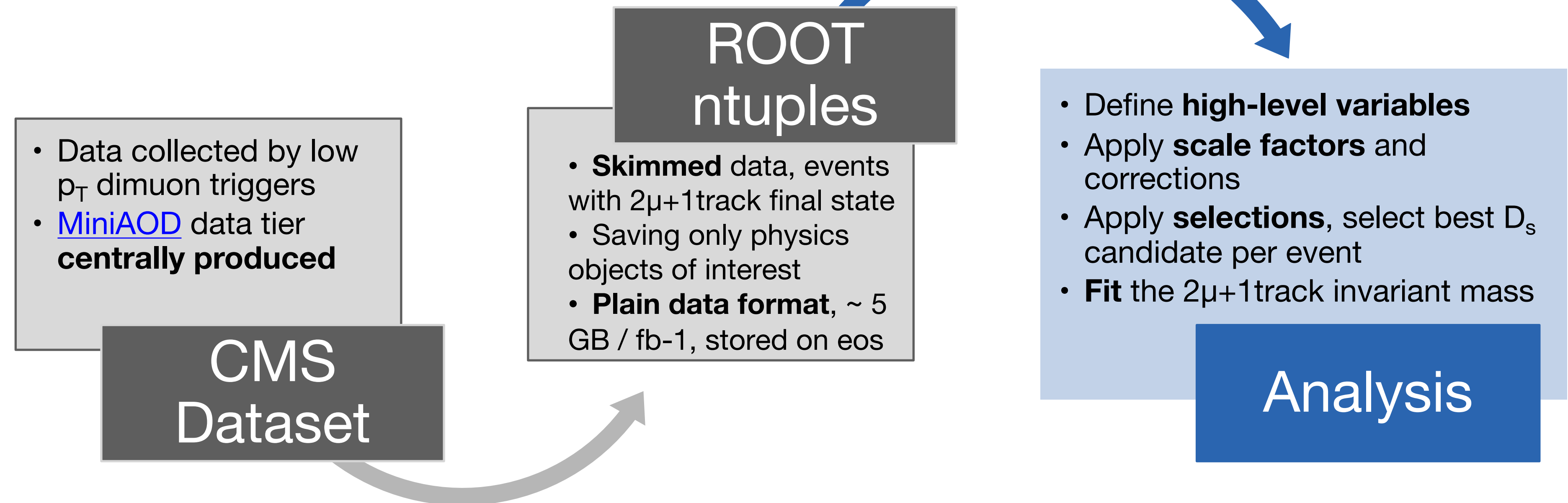
Search for $\tau \rightarrow 3\mu$ decays, which have very small SM branching fractions $BR_{SM} \sim \mathcal{O}(10^{-55})$, while being predicted with sizable BR in several BSM scenarios $BR_{BSM} \sim \mathcal{O}(10^{-10} \div 10^{-8})$

- τ leptons produced in D and B meson decays provide large statistics at LHC experiments, but are only accessible with **low- p_T muon triggers**

- The normalisation channel used as a benchmark: $D_s^+ \rightarrow \phi(\mu\mu)\pi^+$
→ **cut-based analysis + mass fit for measuring the D_s^+ yield in data**

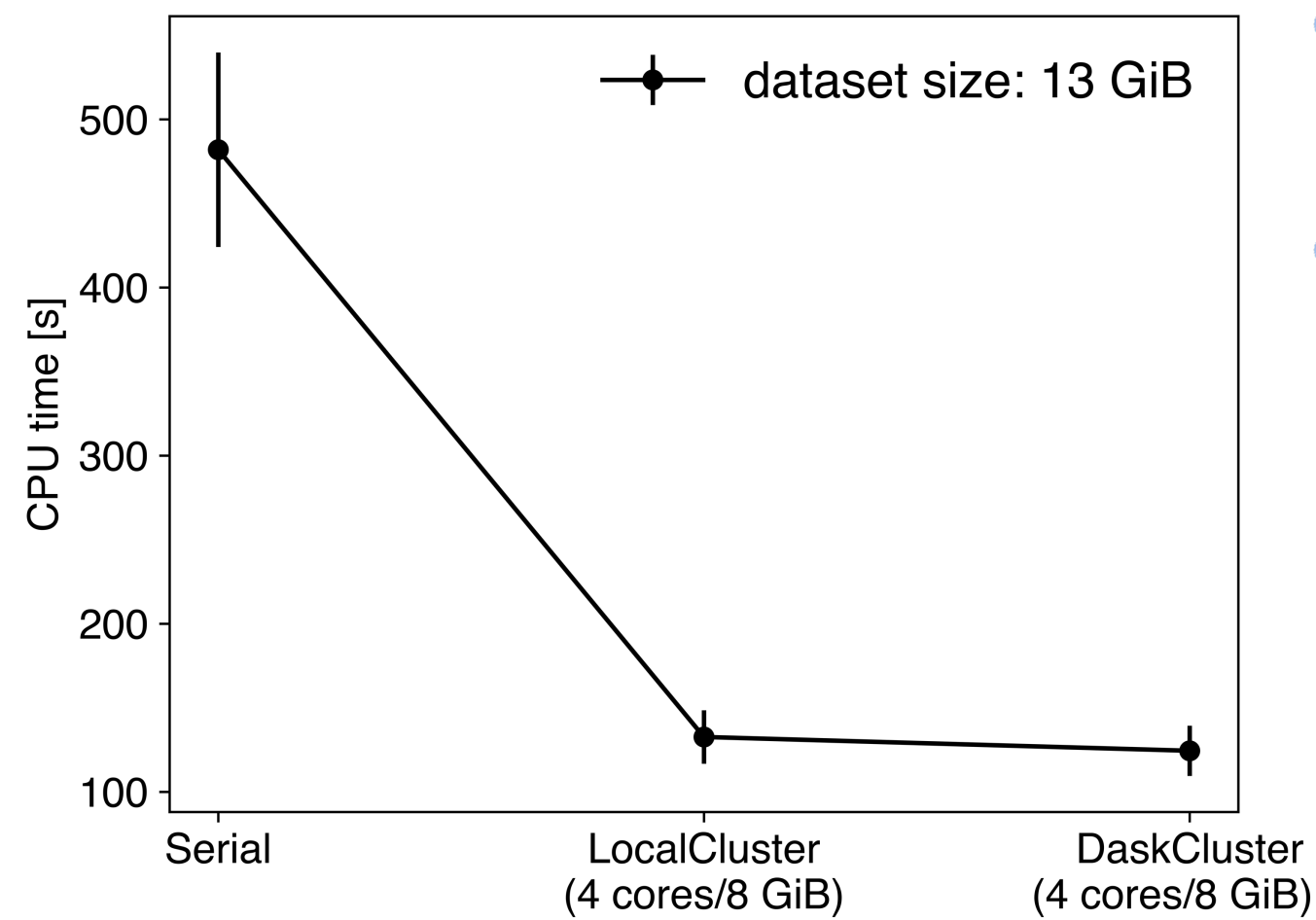


$D_s^+ \rightarrow \phi(\mu\mu)\pi^+$ analysis workflow



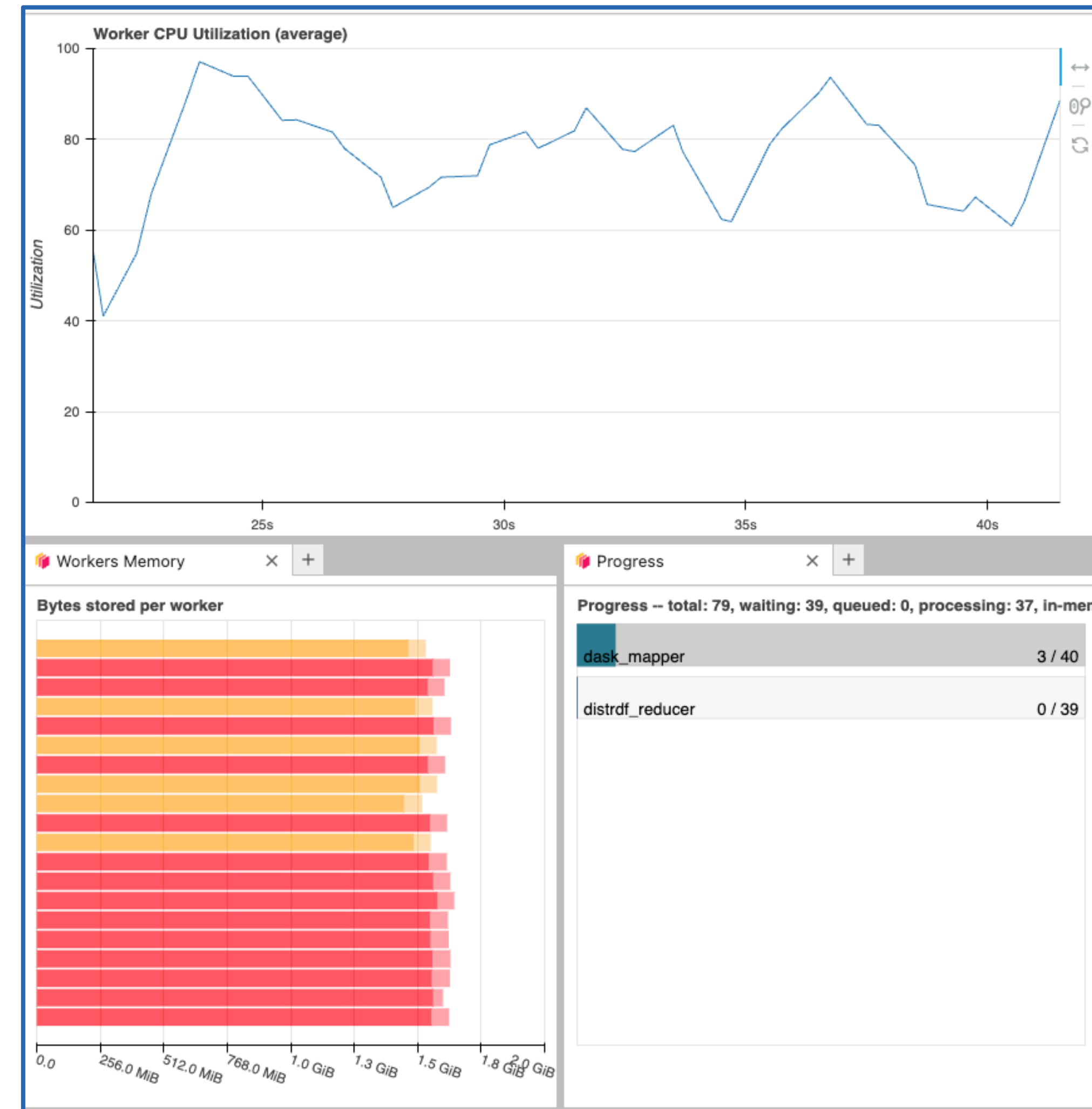
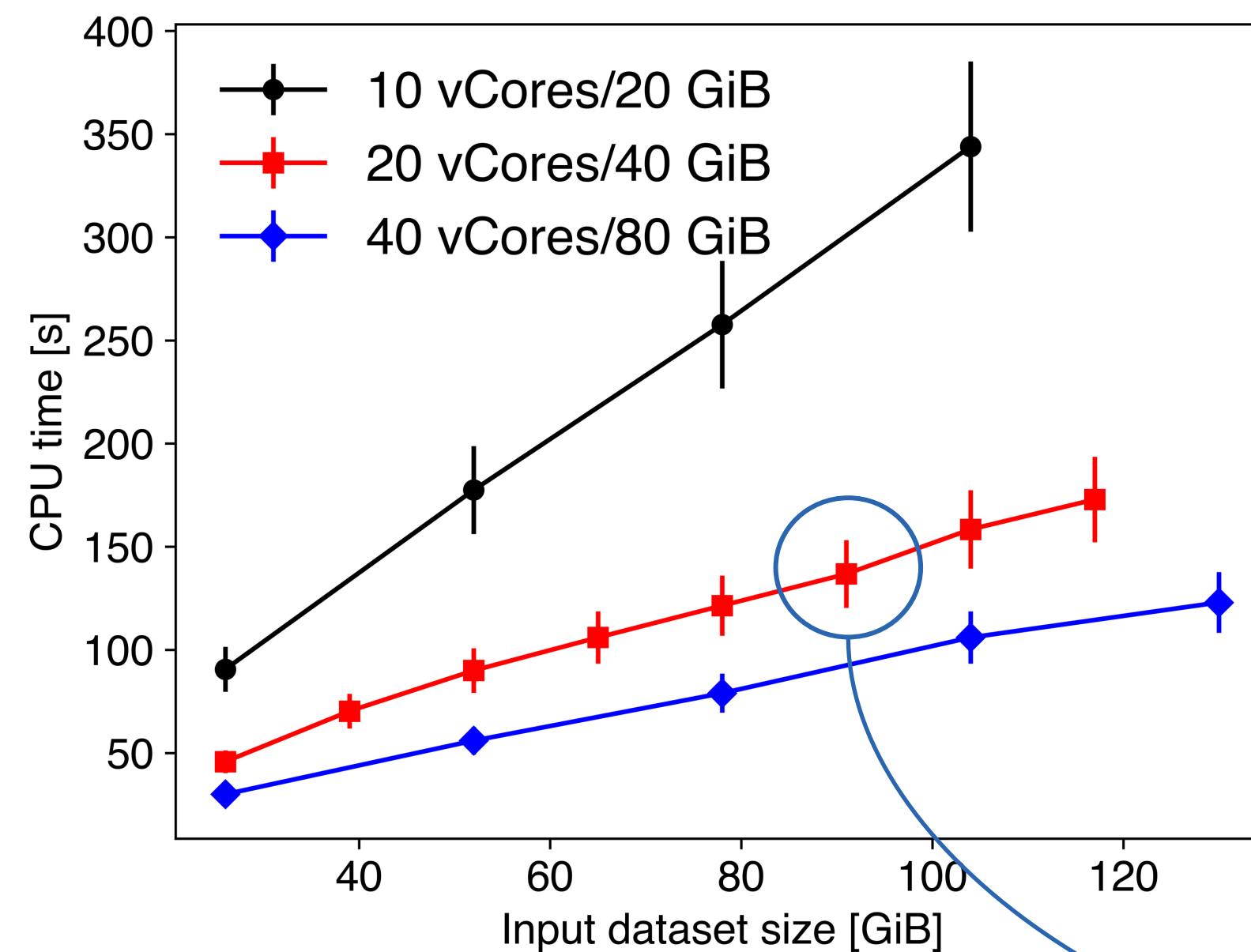
- **Legacy: approach** Loop-based analysis implemented using ROOT TTree:MakeClass
 - split computation in batches of input files, run separately as HTCondor jobs, gather the output rootfiles
- **New:** Ntuples read as RDataFrame, almost all operations “lazy” → no loop triggered till the end
 - going distributed using ROOT RDataFrame distributed features, with Dask backend.

Preliminary results



- Significant improvement in execution time *wrt* the standard/serial approach
- The facility allows for dynamically scaling the resources, here testing the performance at fixed #cores and memory, varying the dataset size

- Stress test at high CPU and memory occupancy
- Stable performance, linearly scaling with the input dataset size
- Dataset size ~ 100 GiB is representative of ~ 15 /fb of Run3 data for this specific analysis



Conclusions & Next Steps

- HL-LHC poses significant challenges to HEP experiments in terms of storage and computing resources
- An interactive high throughput platform has been developed in the framework of the “HPC, Big Data e Quantum Computing Research Centre” Italian National Center (ICSC)
 - offers users a modern interactive web interface based on JupyterLab
 - experiment-agnostic resources
 - based on a parallel and geographically distributed back-end
- Interactive analyses feasibility studies on INFN cloud succeeded
 - 📌 Performance evaluated using the high-rate platform
 - 📌 HEP analysis use-case explored from the CMS and ATLAS Collaborations

Medium-long term goals: Expand the current pool of resources by a factor of 5 in the upcoming months, to perform scale testing of the analysis workflows.

The background is a deep blue gradient. On the left side, there are numerous thin, glowing blue lines that curve and converge towards the center, creating a sense of depth and movement. Interspersed among these lines are small, bright blue particles or dots, some of which appear to be trailing off, giving the impression of light or data being processed.

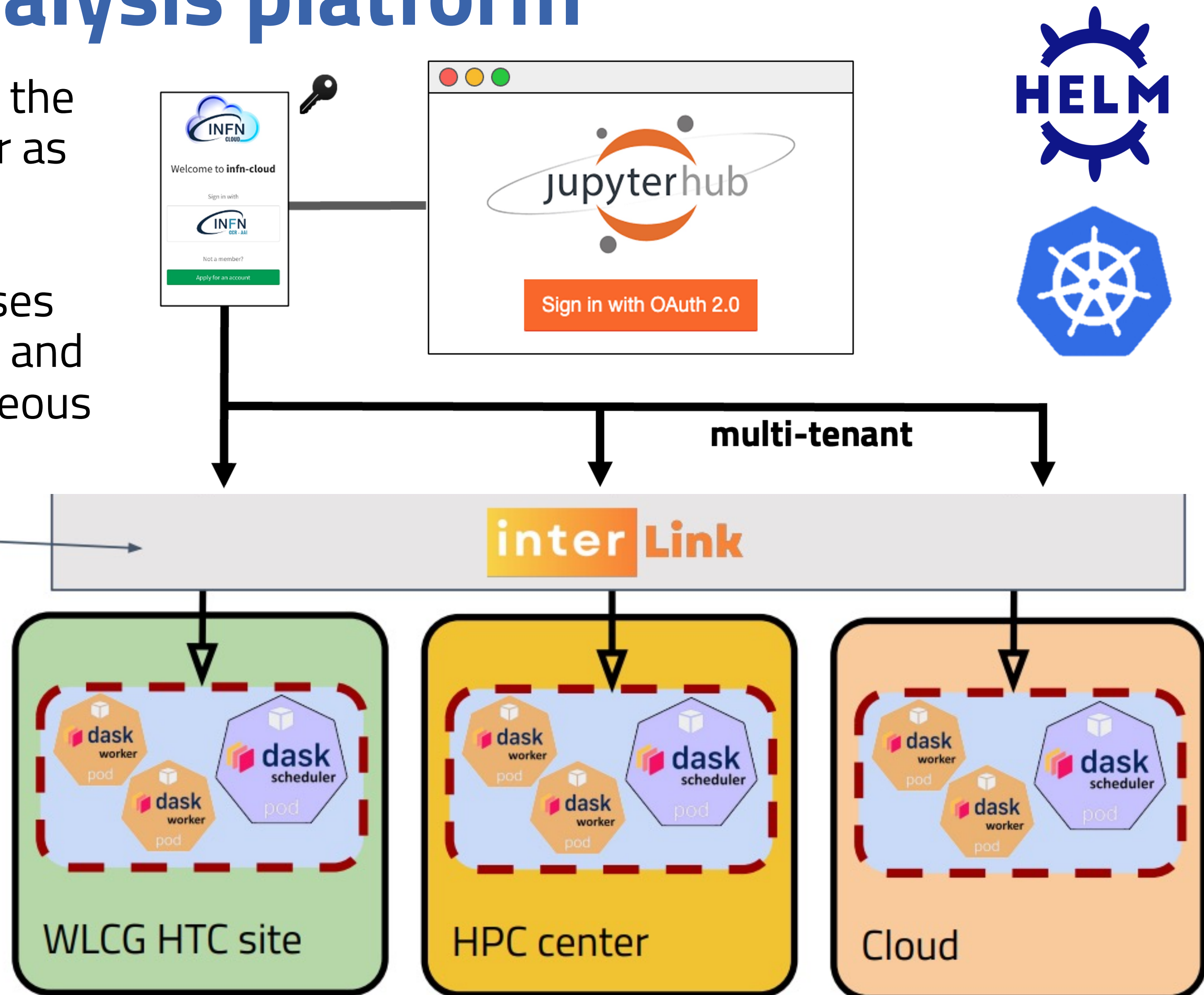
Thank you!

Back-up

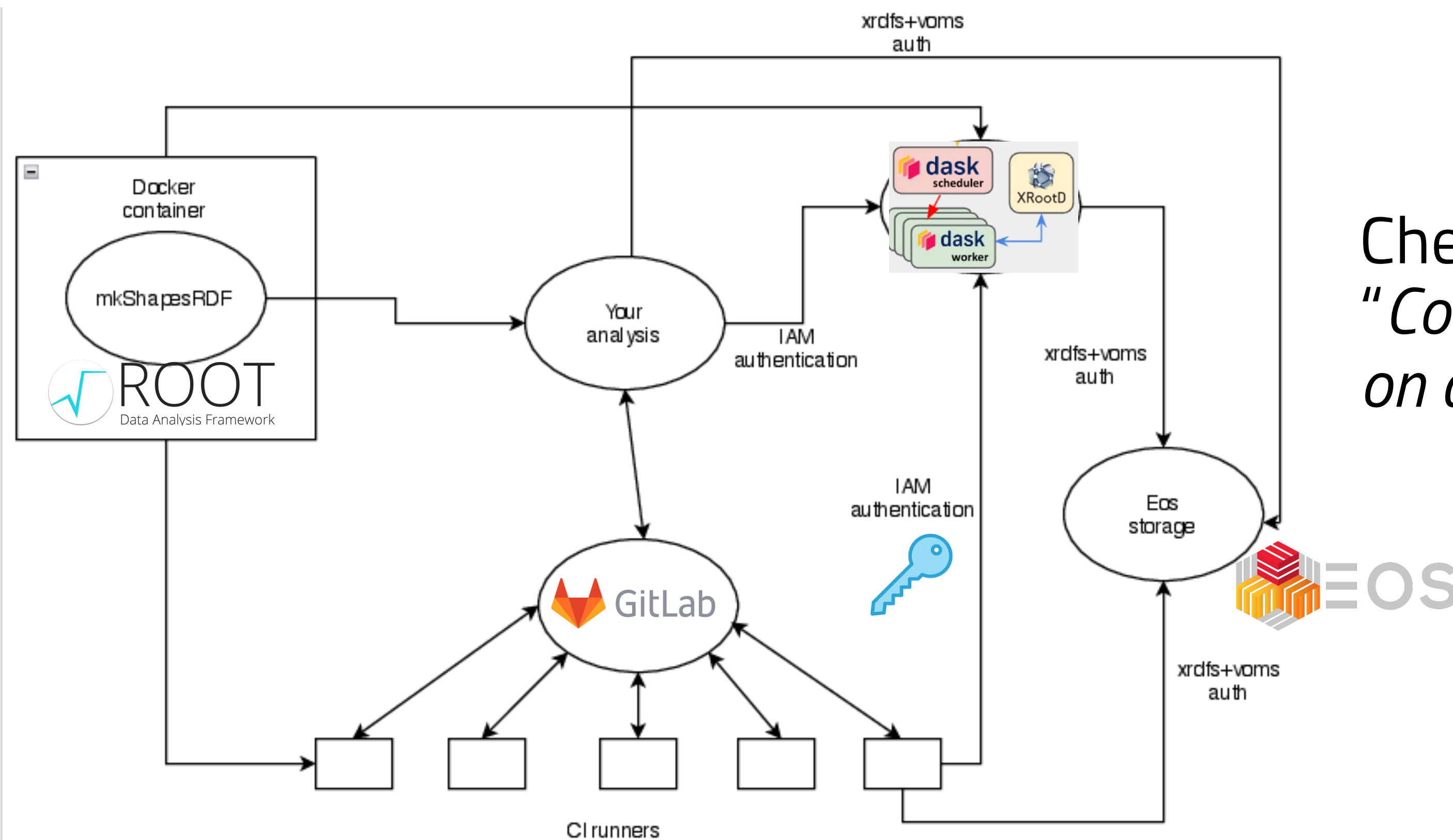
High throughput data analysis platform

- Offloading strategy: resources used to offload the computation are hosted in the same k8s cluster as the jupyter interface, via DASK KubeCluster
- **Under development:** schedule worker processes spawning on multiple remote sites dynamically and transparently → Implementation on heterogeneous resources (HTC/HPC/Cloud)

[InterLink](#) provides execution of a Kubernetes pod on almost any remote resource. Resources visible to the user thanks to an HTCondor overlay



CI triggered CMS analysis execution on the High Rate platform



Check out the [poster by Matteo Bartolini](#) "Continuous integration of analysis workflows on a distributed analysis facility"

Run3 CMS Luminosity

