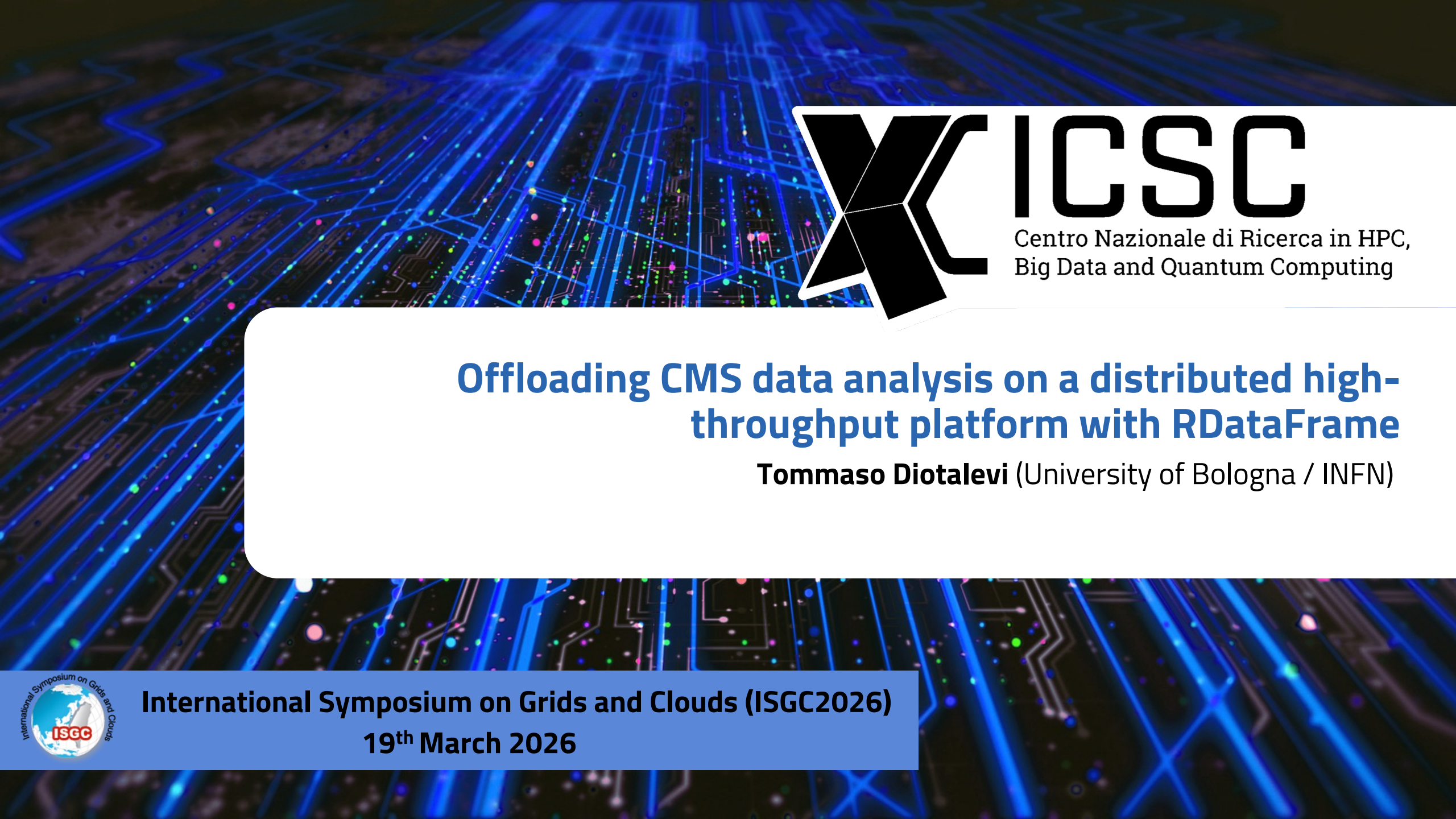


ICSC
Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

Offloading CMS data analysis on a distributed high-throughput platform with RDataFrame

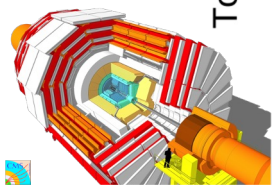
Tommaso Diotalevi (University of Bologna / INFN)

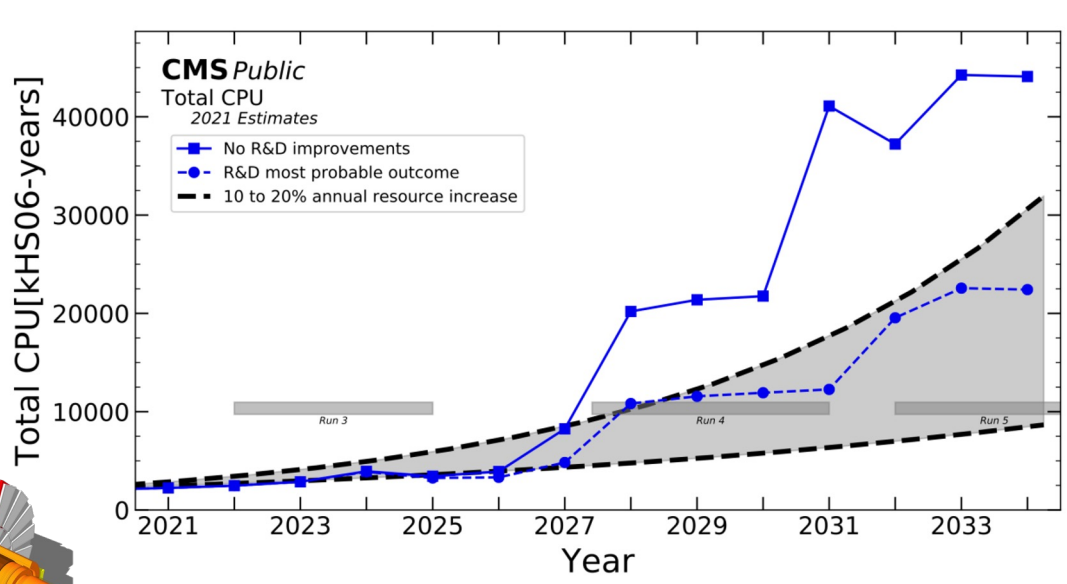
International Symposium on Grids and Clouds (ISGC2026)
19th March 2026

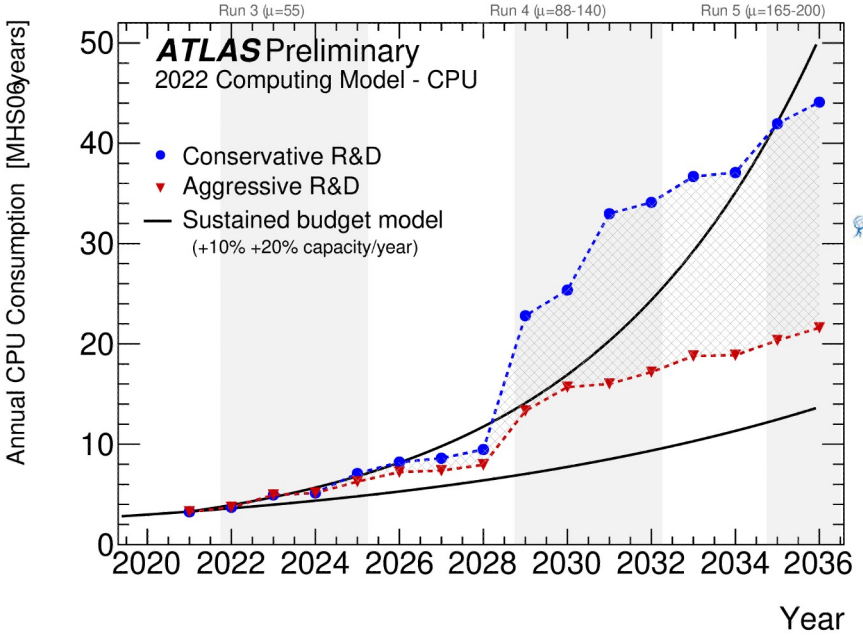
Introduction

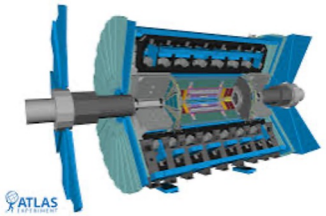
- Analysing large amounts of data efficiently, exploiting the available resources as much as possible, is a common challenge both for research and industry.

- From the beginning, the High Energy Physics (HEP) experiments at CERN, gave much attention to the computing and data management aspects. Nevertheless, the **next phases of the Large Hadron Collider** (HL-LHC) will require an even greater effort.









Introduction

Some estimate for the next 5-10 years of CMS operation:

- ~30 Billion collision events + 30 Billion simulation events;
- Each event: 2-4 kB;
- The last update of the CMS Computing model foresees this throughput:

Name	Length	% of the dataset	Data to process	Event, data rate
"A coffee"	< 5 min	1% (~0.6B evts)	~2 TB	~1.7MHz, ~7GB/s
"A lunch break"	1 hour	10% (~6B evts)	~20 TB	~1.5MHz, ~6GB/s
"A night"	12 hours	100% (60B evts)	~200 TB	~1.2MHz, ~5GB/s

- Difficult to get more than 100 Hz/CPU core → needs efficient distribution on a few tens of machines;

Not only concerning the HEP domain ("Data is data"):

- More scientific / industrial / societal domains have or will have soon needs like those from LHC:

DUNE DEEP UNDERGROUND NEUTRINO EXPERIMENT — ProtoDune: 2-3GB/s (like CMS); Real Dune: 80x

SKA SQUARE KILOMETRE ARRAY — SKA: up to 2 PB/day;

CTAO — CTA projects: up to 10PB/y

New analysis paradigm based on:

- Declarative programming and interactive workflows;
- Distributed computing on geographically separated resources.

Introduction

Some estimate for the next 5-10 years of CMS operation:

- ~30 Billion collision events + 30 Billion simulation events;
- Each event: 2-4 kB;
- The last update of the CMS Computing model foresees this throughput:

Name	Length	% of the dataset	Data to process	Event, data rate
"A coffee"	< 5 min	1% (~0.6B evts)	~2 TB	~1.7MHz, ~7GB/s
"A lunch break"	1 hour	10% (~6B evts)	~20 TB	~1.5MHz, ~6GB/s
"A night"	12 hours	100% (60B evts)	~200 TB	~1.2MHz, ~5GB/s

- Difficult to get more than 100 Hz/CPU core → needs efficient distribution on a few tens of machines;

Not only concerning the HEP domain ("Data is data"):

- More scientific / industrial / societal domains have or will have soon needs like those from LHC:

DUNE DEEP UNDERGROUND NEUTRINO EXPERIMENT — ProtoDune: 2-3GB/s (like CMS); Real Dune: 80x

SKA SQUARE KILOMETRE ARRAY — SKA: up to 2 PB/day;

CTAO — CTA projects: up to 10PB/y

New analysis paradigm based on:

High Throughput Platform

- Declarative programming and interactive workflows;
- Distributed computing on geographically separated resources;

ICSC: The National Center for HPC, Big Data and Quantum Computing

what is it?

- Italy has funded, with NRRP (pandemic recovery) funds, **5 large National Centers**, for a total of **1.6B€** over 3 years, on key future technologies.
- **ICSC**, coordinated by **INFN**, focuses on modern IT technologies, with the final goal of deploying a long-term distributed infrastructure (>> 3y) for national research and industrial development.
- The project started on September 2022, lasting until April 2026.

Want to know more?

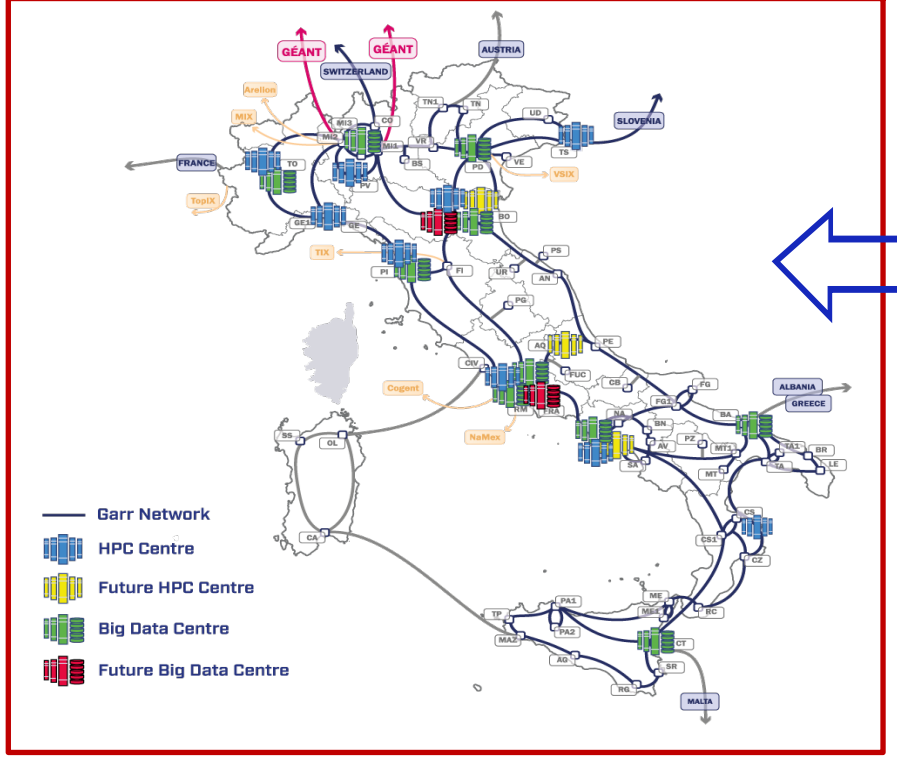
Take a look at yesterday's presentation

1. ICSC: Big Data, High Performance and Quantum Computing
2. Advanced Technologies in agricolture (Agritech)
3. Sustainable Mobility
4. Development of RNA drugs and gene therapies
5. Biodiversity

The structure of the ICSC National Center

The ICSC includes:
10 thematic Spokes and **1 Infrastructure Spoke**

0 SUPERCOMPUTING CLOUD INFRASTRUCTURE



EDUCATION AND TRAINING, ENTREPRENEURSHIP, KNOWLEDGE TRANSFER, POLICY MAKING, OUTREACH

1 FUTURE HPC & BIG DATA

2 FUNDAMENTAL RESEARCH & SPACE ECONOMY

3 ASTROPHYSICS & COSMOS OBSERVATIONS

4 EARTH & CLIMATE

5 ENVIRONMENT & NATURAL DISASTERS

6 MULTISCALE MODELING & ENGINEERING APPLICATIONS

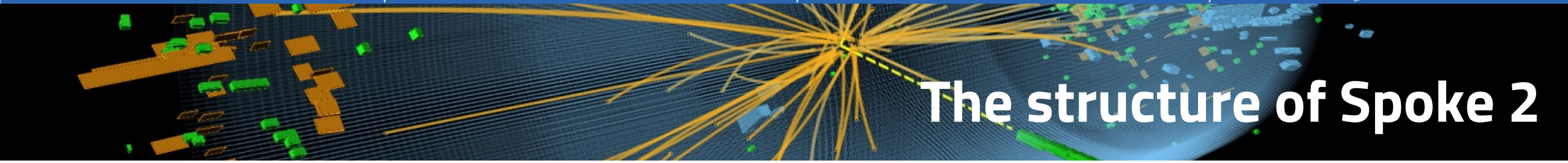
7 MATERIALS & MOLECULAR SCIENCES

8 IN-SILICO MEDICINE & OMICS DATA

9 DIGITAL SOCIETY & SMART CITIES

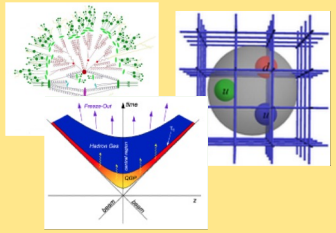
10 QUANTUM COMPUTING

- One Spoke dedicated to building the infrastructure
- Ten thematic Spokes, one of which dedicated to the HEP and Astroparticle research domains.

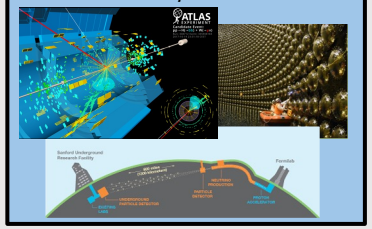


The structure of Spoke 2

WP1: tools and algorithms for Theoretical Physics

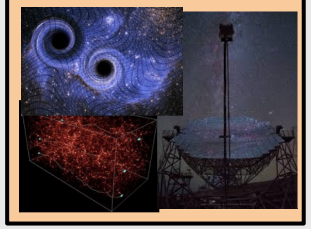


WP2: tools and algorithms for Experimental High Energy Physics



Scientific

WP3: tools and algorithms for Experimental Astroparticle Physics and Gravitational waves



ICSC-SPOKE2
Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

WP6: cross domain initiatives + space economy

WP5: Boosting computational performance on the distributed CN infrastructure

WP4: tools for porting/optimization on new architectures (low power, GPU, FPGA, ...)

Technologic

FUNDAMENTAL RESEARCH & SPACE ECONOMY

Istitution leader: INFN Istituto Nazionale di Fisica Nucleare

Istitution co-leader: INAF Istituto Nazionale di Astrofisica

Istitutions and Universities: Politecnico di Bari; Sapienza Università di Roma; Università degli Studi di Bari Aldo Moro; Alma Mater Studiorum Università di Bologna; Università della Calabria; Università di Catania; Università degli Studi di Ferrara; Università degli Studi Firenze; Università degli Studi di Milano Bicocca; Università degli Studi di Napoli Federico II; Università degli Studi di Padova; Università del Salento; Università degli Studi di Trieste

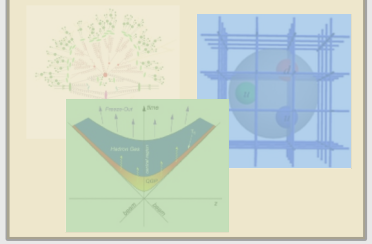
Companies: eni; Intesa Sanpaolo; Leonardo; ThalesAlenia Space; UnipolSai Assicurazioni; iFAB International Foundation Big Data and Artificial Intelligence for Human Development; sogei

Staff Researchers	195
(kEur)	6333
Rectruited researchers	28
(kEur)	5067
Phd positions	25
(kEur)	1992
Budget Innovation Grants (kEur)	1800
Bugdet Cascade Calls (kEur)	3200
Total Budget (kEur)	**18391**

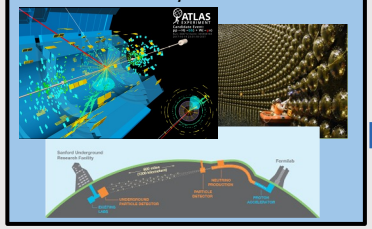
Quasi interactive analysis of Big Data with high throughput

Scientific

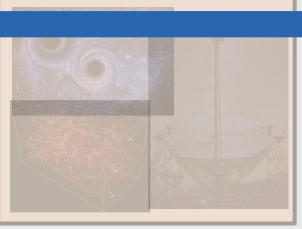
WP1: tools and algorithms for Theoretical Physics



WP2: tools and algorithms for Experimental High Energy Physics



WP3: tools and algorithms for Experimental Astroparticle Physics and Gravitational waves



ICSC-SPOKE2
Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

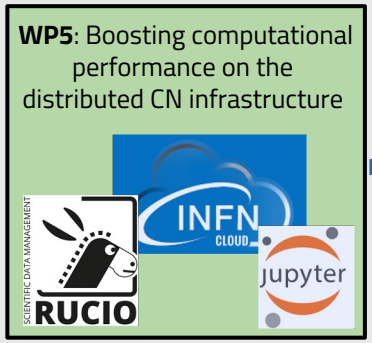
Technologic

WP4: tools for porting/optimization on new architectures (low power, GPU, FPGA, ...)

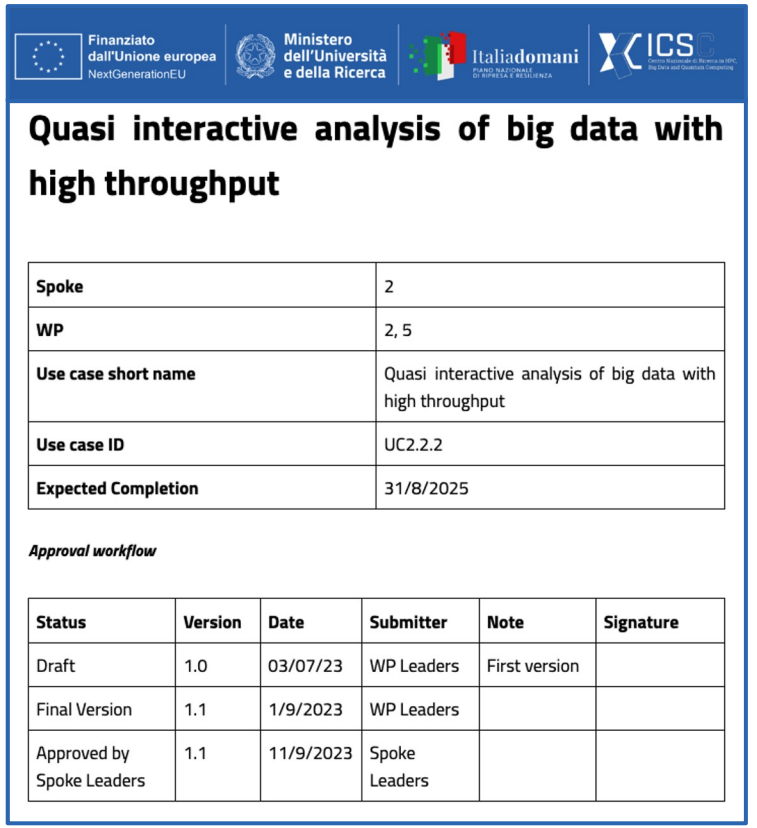
supported by

WP6: cross domain initiatives + space economy

WP5: Boosting computational performance on the distributed CN infrastructure



1 of the 19 Spoke2 flagship use cases:



Finanziato dall'Unione europea NextGenerationEU | Ministero dell'Università e della Ricerca | Italiadomani | ICSC

Quasi interactive analysis of big data with high throughput

Spoke	2
WP	2, 5
Use case short name	Quasi interactive analysis of big data with high throughput
Use case ID	UC2.2.2
Expected Completion	31/8/2025

Approval workflow

Status	Version	Date	Submitter	Note	Signature
Draft	1.0	03/07/23	WP Leaders	First version	
Final Version	1.1	1/9/2023	WP Leaders		
Approved by Spoke Leaders	1.1	11/9/2023	Spoke Leaders		

Principal Investigators:

Tommaso Diotalevi

Francesco G. Gravili

Re-thinking the analysis pipeline

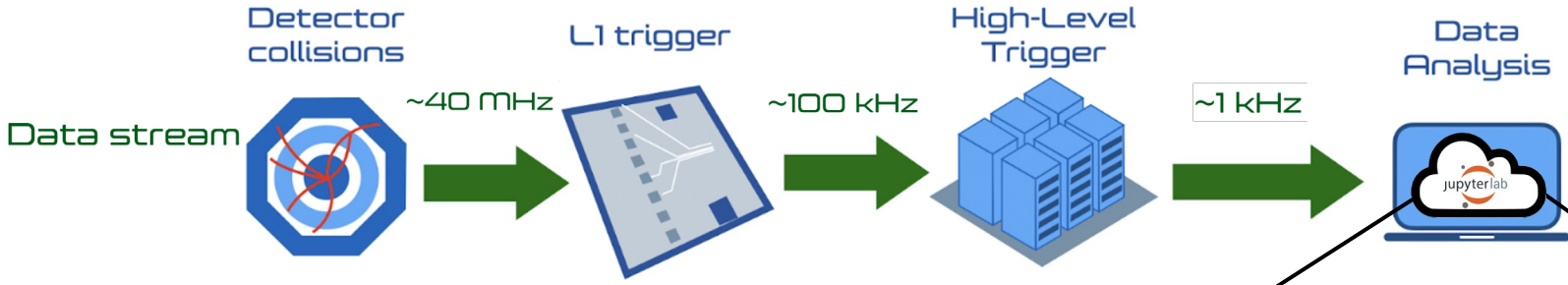
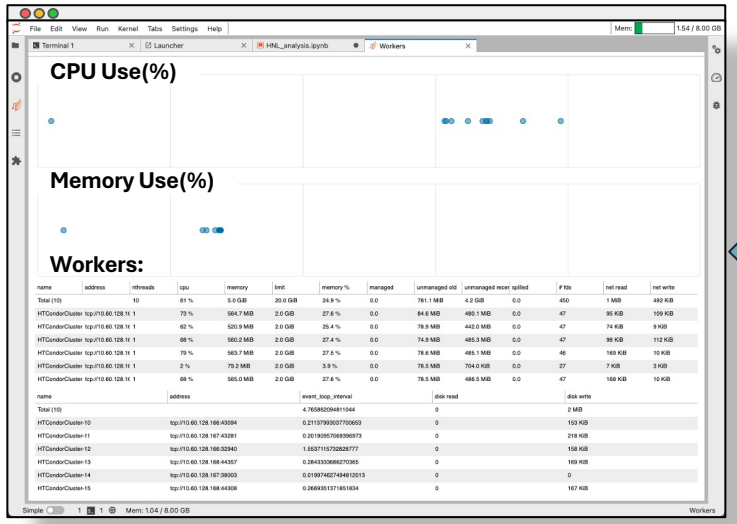
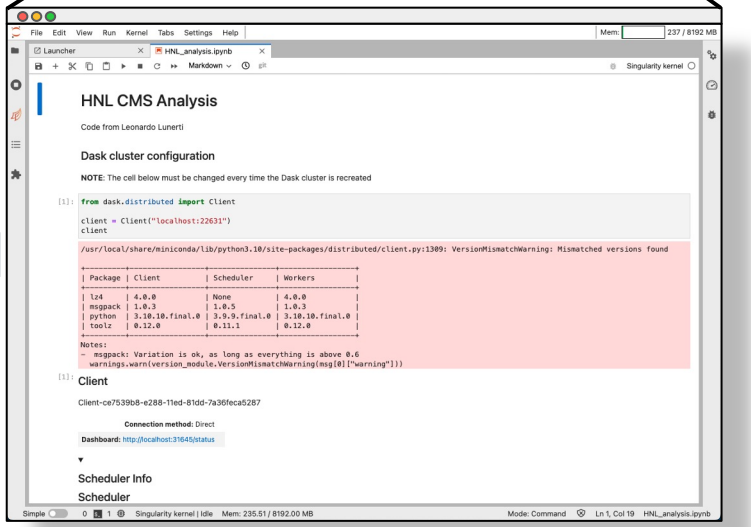


Image taken from: link

Resource monitoring dashboard



Analysis code



The high throughput platform

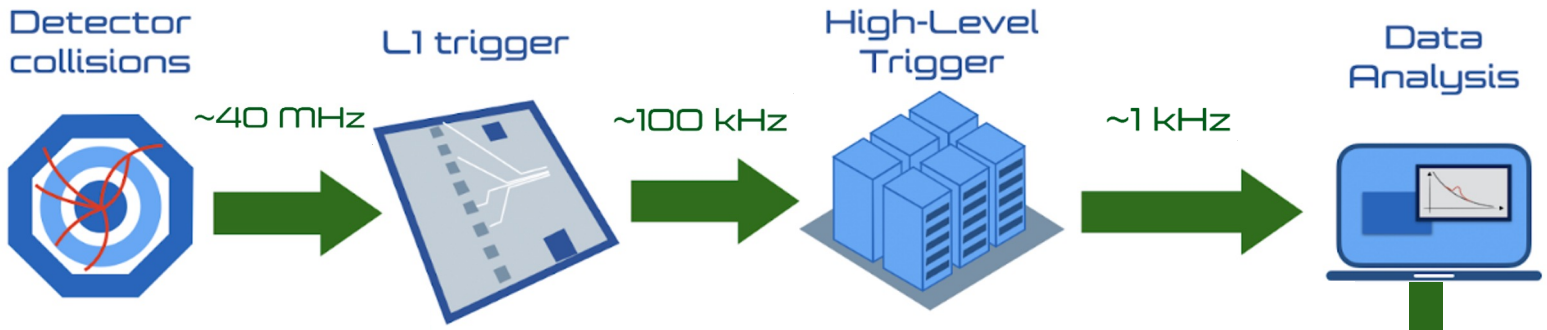


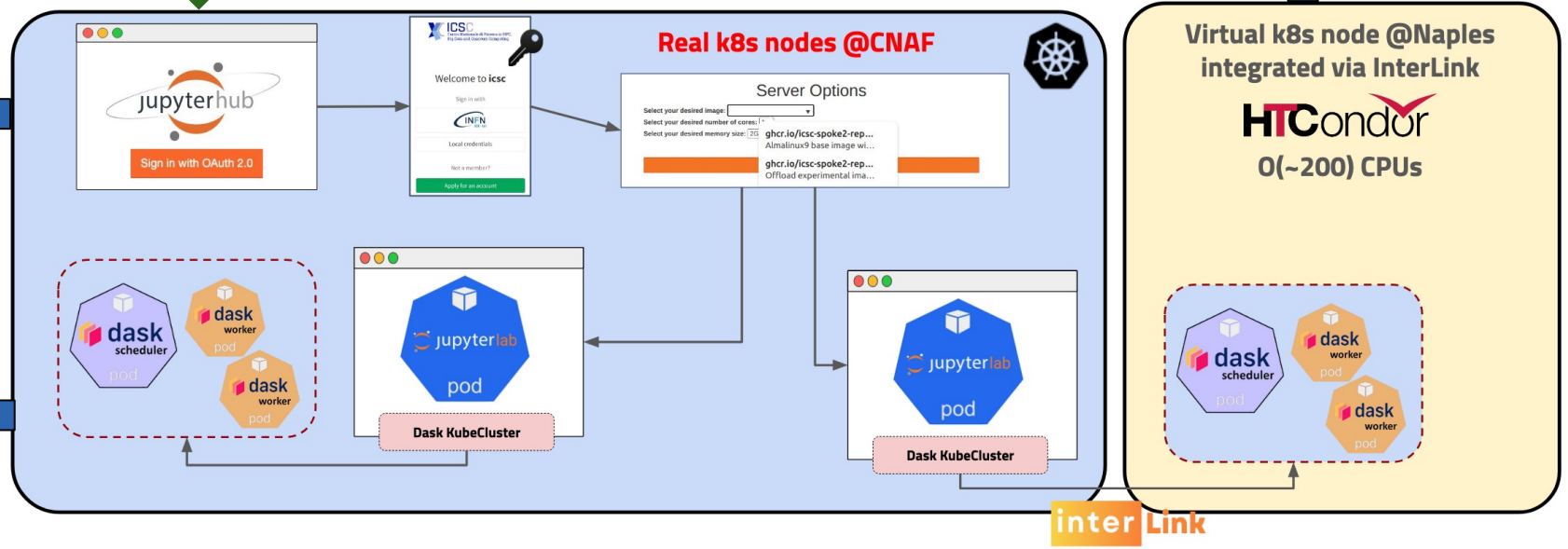
Image taken from: link

New!

- Offload capability on remote HTC resources, via **Interlink**;

Deployment of the Kubernetes resources handled via HELM Charts.

(Scalable on available resources)

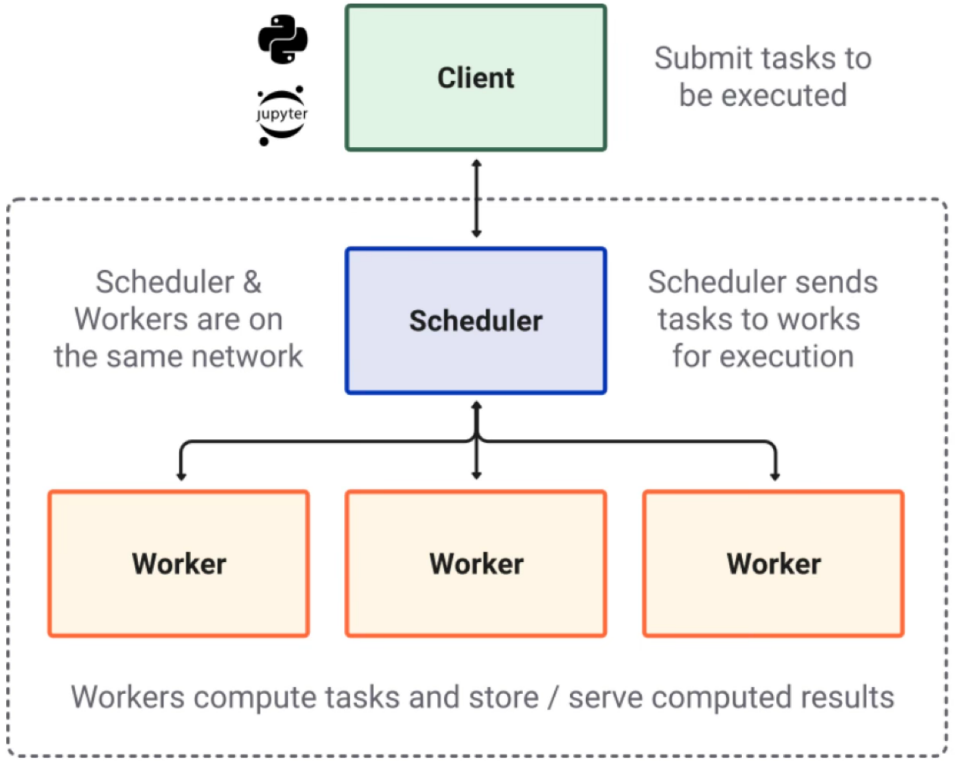


- The distributed execution happens in the **Dask Cluster**;
- Users choose on how many cores parallelly distribute the analysis.

Dask cluster

A **Dask cluster** is a distributed computing system that allows Python workloads to run in parallel across multiple machines or processes.

1. **Client**
   User-facing interface (Python session, notebook, script);
   Submits computations and tasks to the cluster.
2. **Scheduler**
   Central coordinator of the cluster;
   Builds the task graph and decides **which worker runs which task;**
   Tracks dependencies and task status.
3. **Workers**
   Processes that execute tasks;
   Each worker:
   - Has **multiple threads or processes**
   - Holds **data in memory**
   - Returns results to the scheduler/client

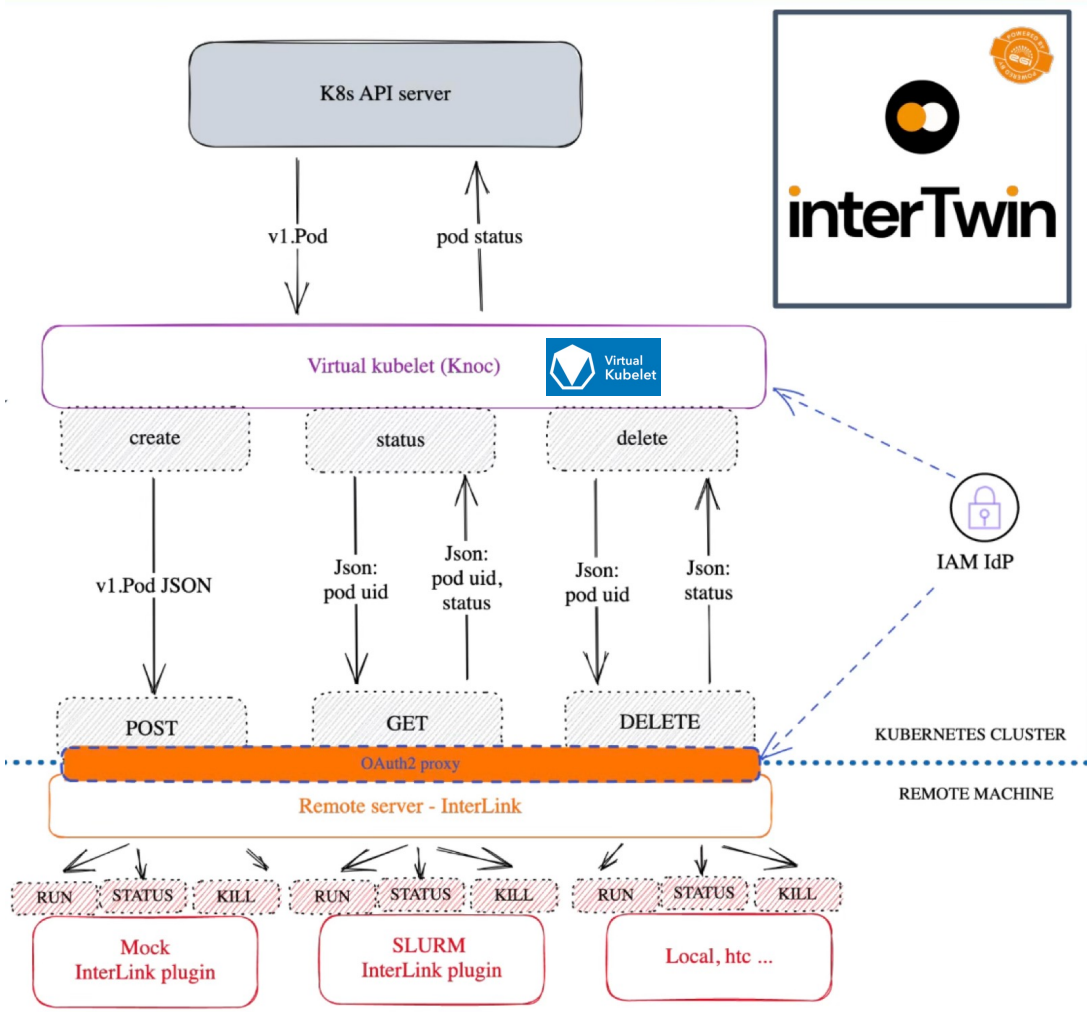


Dask Cluster

Offloading strategy with Interlink

Scheduling worker processes, spawning on multiple remote sites dynamically and transparently:

- **Virtual Kubelet** (open-source Kubernetes kubelet implementation that masquerades as a kubelet): registers as a virtual node and pulls work to run;
  - "It takes your pod and executes it wherever"
- InterLink **+ HTCondor Sidecar (Plugin)**: pods are translated into HTCondor jobs:
  - Translating interlink create/status/delete calls interacting with the proper HTCondor schedd via CLI
    - POST /create call -> condor_submit
    - GET /status call -> condor_q
    - POST /delete call -> condor_rm
- This strategy can be also applied to other job scheduling systems (e.g. Slurm/HPC)



Orbiting activities

Vector Boson Scattering ssWW analysis in hadronic tau and light lepton

Heavy Neutral Lepton search on heavy neutrinos in the D_S decays

Muon detector performance analysis

Continuous Integration pipeline, triggering analysis execution on HTP

di-Higgs decaying to two b quarks and two muons

Search of rare events in tau to 3 muons decay

Differential cross section measurement for ttbar inclusive production

Search for new phenomena in events with two opposite-charge leptons, jets and missing transverse momentum

Neural Network hyperparameter optimisation applied to future colliders (FCC-ee)

Declarative paradigms for analysis description and implementation

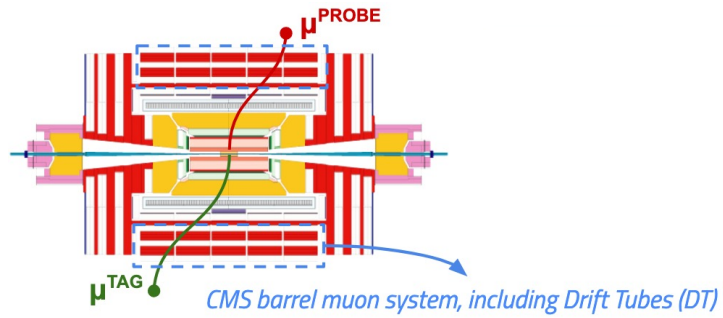
top quark+MET analysis

Benchmark interactive analysis for future colliders (FCC-ee)

Distributing the Simulated Annealing workload for Quantum Unfolding in HEP

Offloading a CMS analysis

Muon detector performance analysis @ CMS *[T.Diotalevi, C.Battilana]*



Typically, Detector Performance Group (DPG) analyses are run on a reduced amount of data (e.g. one run or fill), but processing of large dataset, at once, might be needed:

- To assess/improve systematics of high precision analyses, when they are dominated by the response of a specific detector;
- To reprocess multiple year data, e.g. for detector stability studies (ageing).

Use case

Porting of a well established Drift Tubes (DT) Tag-and-Probe analysis [CMS-DP-2023-049]

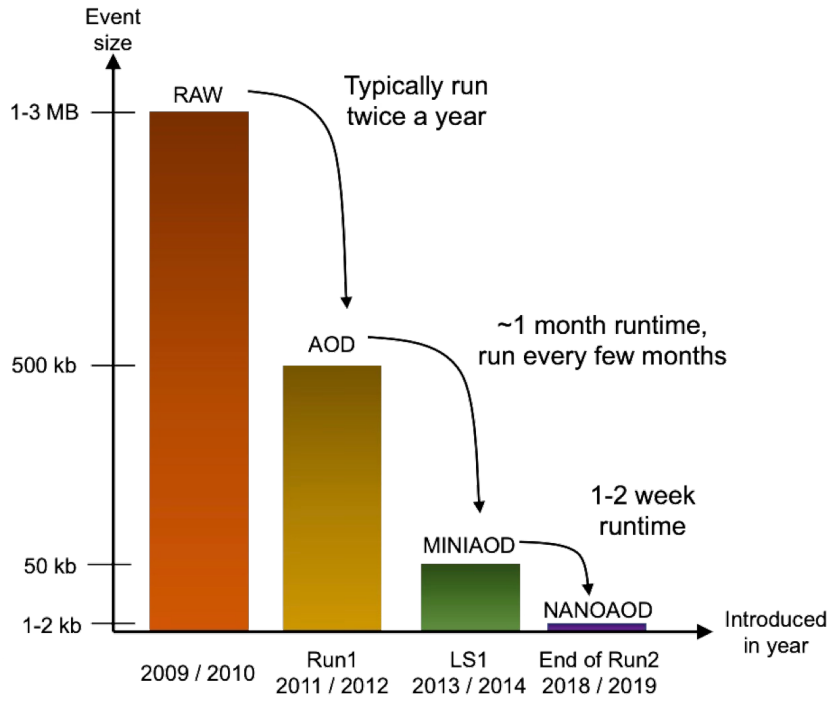
A data sample consisting in a skim of $Z \rightarrow \mu\mu$ decay candidates collected by CMS over 2023, corresponding to ~27fb^{-1} was explored for the study. **Size: 224GB**

- To evaluate the technical performance, the available statistics has been processed 3 times, mimicking roughly a entire year of data taking. **Size**: 224*3 = **672GB**

Data processing in CMS

The entirety of CMS data, centrally produced, are saved in ROOT files, in different data formats:

- **RAW**: A collection of the detector electronics output. Event size: 1-3 MB
- **AOD**: Analysis Object Data. Transforming RAW data in analysis objects (used by analysts) like jets, muons, electrons, etc… Event size: 500kB
- **MiniAOD**: Reducing the size of AODs, making them more compact with the downside of losing some information (e.g. zeroing floating-point numbers bit). Event size: 50kB
- **NanoAOD**: Further reduction of MiniAODs, saved as a columnar ROOT file. This new format, using fundamental data types (int, float), exits from the CMS ecosystem and requires a small number of dependencies to be analysed. Event size: 2-4 kB.



(image taken from: link to CHEP19 contribution)

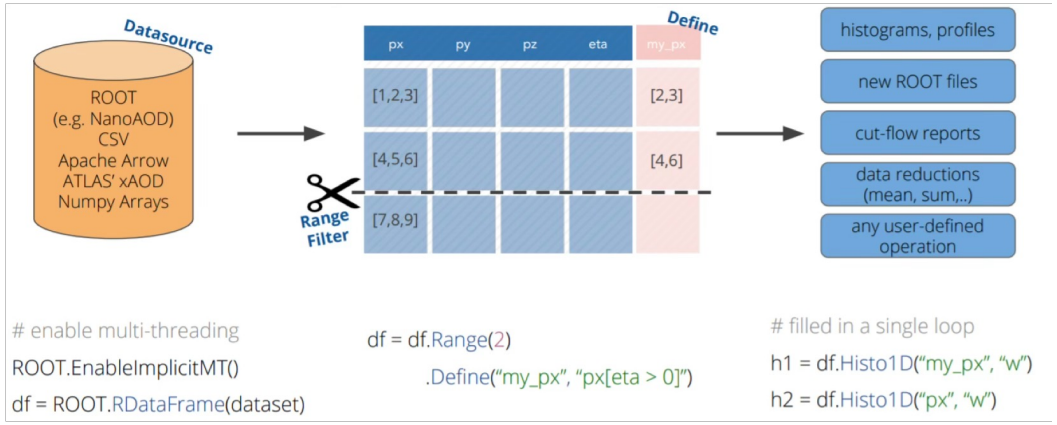
Dataset used, based on muon DPG common NANO flavour: a NanoAOD-like dataset, with 410 physical variables, tailored for DPG-based analyses.

ROOT Data Analysis Framework

ROOT RDataFrame

RDataFrame (RDF) is the high-level interface of ROOT for the data analysis saved in TTree, CSV and many other data formats. It is based on:

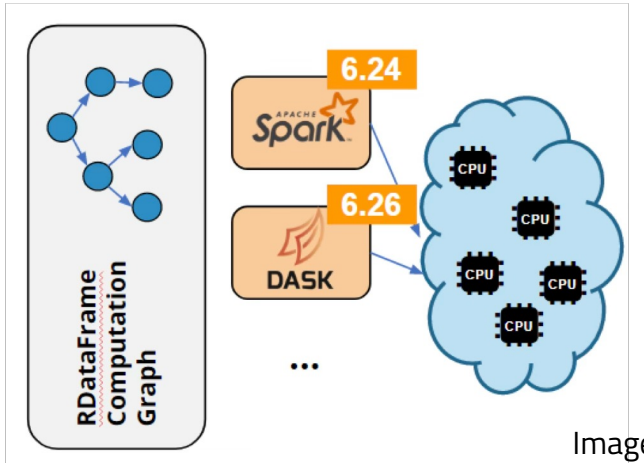
- multi-threading;
- low level optimisations (parallelism and caching).



Computations are expressed in terms of actions and transformations chain, constituting a computational graph.

The execution of such graph can be made also distributed, exploiting backends such as Dask and Spark.

Thanks to the "distributed" extension of RDF, available experimentally.



Images taken from:
PyHEP 2021

Porting results

The original code running mainly on C++, for the base histograms and computing the segment efficiencies.

- The ported code is running on Jupyter notebook (in Python), using **ROOT RDataFrame**. The Tag-and-Probe libraries are stored in a dedicated header file.

```
#include "RDataFrame.hxx"
#include "RVec.hxx

using namespace ROOT::VecOps;
RVec<RVec<T>> setProbeVars(...) {
   RVec<RVec<T>> probeVars;
   for (std::size_t i{0}; i!=nMuon; ++i){
      …
      }
return probeVars;
}
```

Example of header structure for data collections

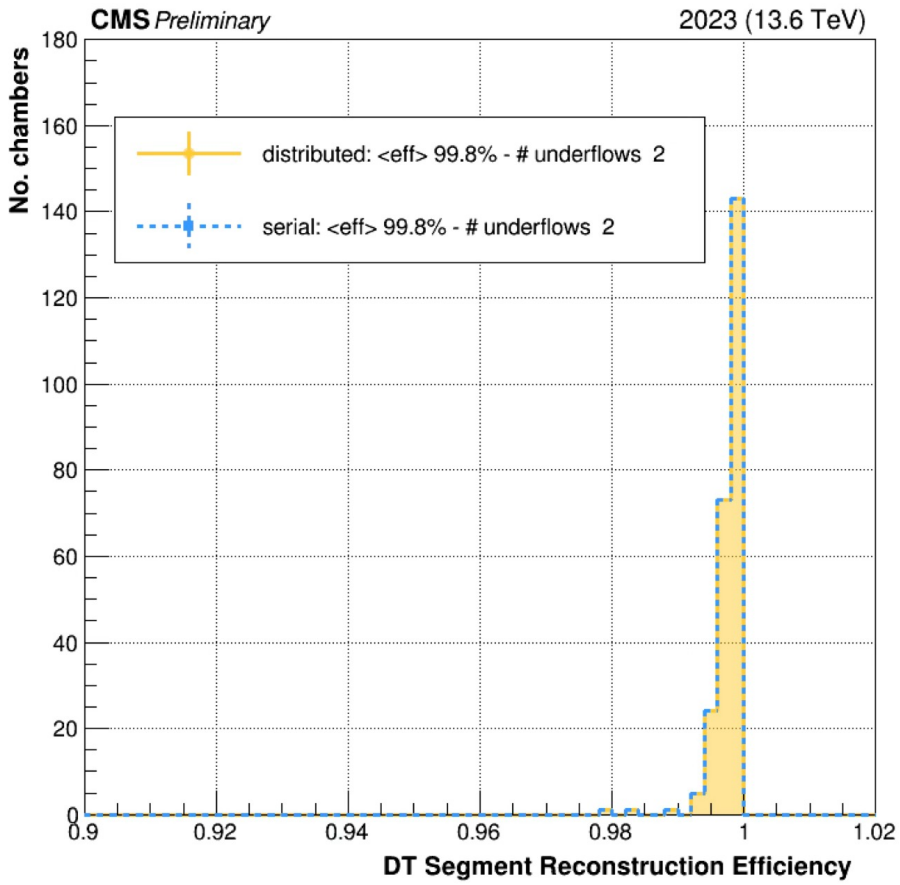
```
# Load header with:
ROOT.RDF.Experimental.Distributed.initialize(...)
# Point RDF calls to Dask-specific RDF
df =
ROOT.RDF.Experimental.Distributed.Dask.RDataFrame
(treeName, inputFiles,...)

df.Define("probeVars", setProbeVars<T>(...))
df.Histo1D(("var0", "var0", 10, 0, 10),
"probeVars[0]")
...
```

Example of RDF implementation

- The changes applied to the Tag-and-Probe program to run on the High Throughput Platform (distributed) do not affect performance, which is consistent with the original program (serial).

(one entry per chamber)



Porting results (cont'd)

The original code running mainly on C++, for the base histograms and computing the segment efficiencies.

- The ported code is running on Jupyter notebook (in Python), using **ROOT RDataFrame**. The Tag-and-Probe libraries are stored in a dedicated header file.

```
#include "RDataFrame.hxx"
#include "RVec.hxx

using namespace ROOT::VecOps;
RVec<RVec<T>> setProbeVars(...) {
   RVec<RVec<T>> probeVars;
   for (std::size_t i{0}; i!=nMuon; ++i){
      …
      }
return probeVars;
}
```

Example of header structure for data collections

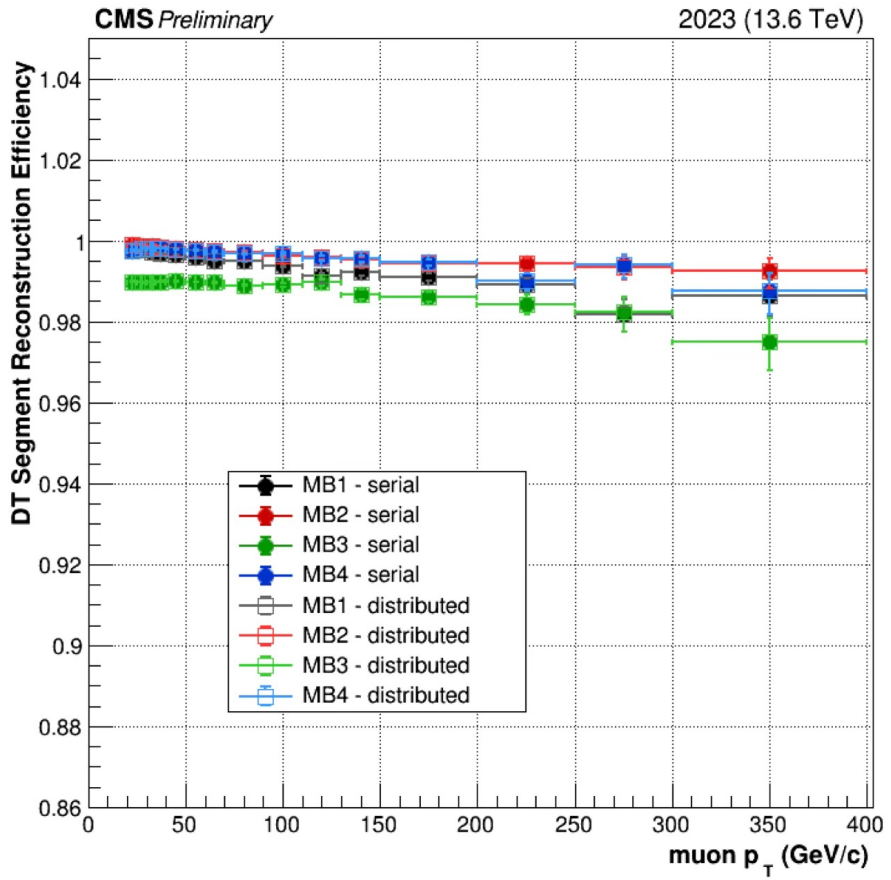
```
# Load header with:
ROOT.RDF.Experimental.Distributed.initialize(...)
# Point RDF calls to Dask-specific RDF
df =
ROOT.RDF.Experimental.Distributed.Dask.RDataFrame
(treeName, inputFiles,...)

df.Define("probeVars", setProbeVars<T>(...))
df.Histo1D(("var0", "var0", 10, 0, 10),
"probeVars[0]")
...
```

Example of RDF implementation

- The changes applied to the Tag-and-Probe program to run on the High Throughput Platform (distributed) do not affect performance, which is consistent with the original program (serial).

(measurement as a function of muon p_T)



Porting results (cont'd)

The original code running mainly on C++, for the base histograms and computing the segment efficiencies.

- The ported code is running on Jupyter notebook (in Python), using **ROOT RDataFrame**. The Tag-and-Probe libraries are stored in a dedicated header file.

```
#include "RDataFrame.hxx"
#include "RVec.hxx

using namespace ROOT::VecOps;
RVec<RVec<T>> setProbeVars(...) {
   RVec<RVec<T>> probeVars;
   for (std::size_t i{0}; i!=nMuon; ++i){
      …
      }
return probeVars;
}
```

Example of header structure for data collections

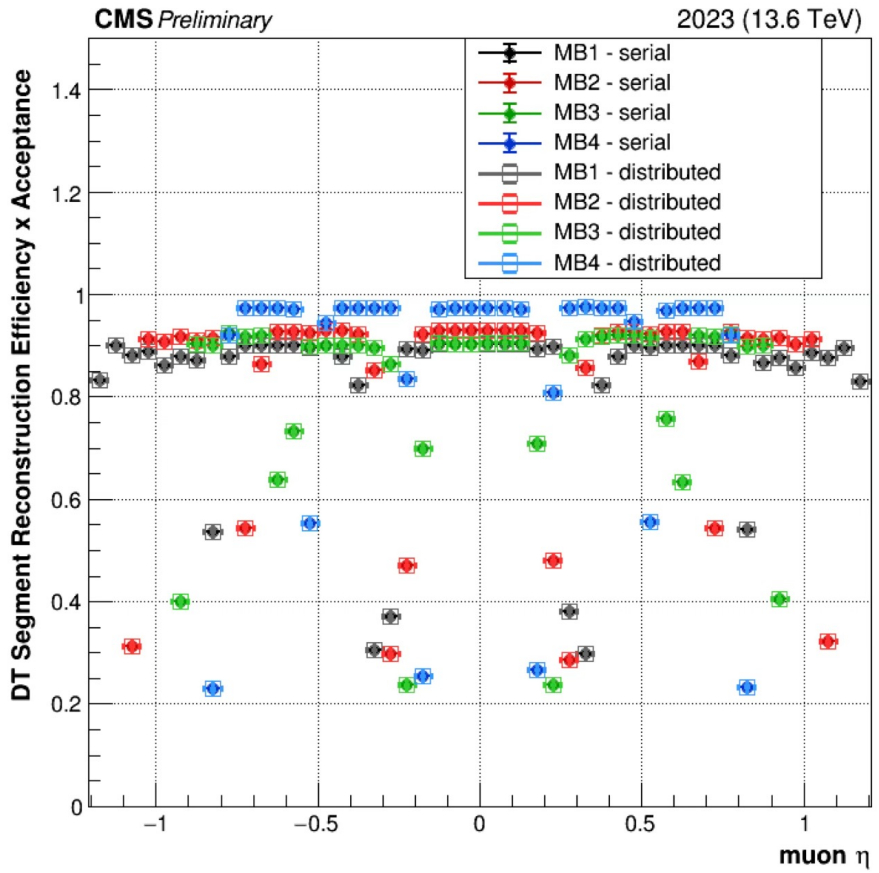
```
# Load header with:
ROOT.RDF.Experimental.Distributed.initialize(...)
# Point RDF calls to Dask-specific RDF
df =
ROOT.RDF.Experimental.Distributed.Dask.RDataFrame
(treeName, inputFiles,...)

df.Define("probeVars", setProbeVars<T>(...))
df.Histo1D(("var0", "var0", 10, 0, 10),
"probeVars[0]")
...
```

Example of RDF implementation

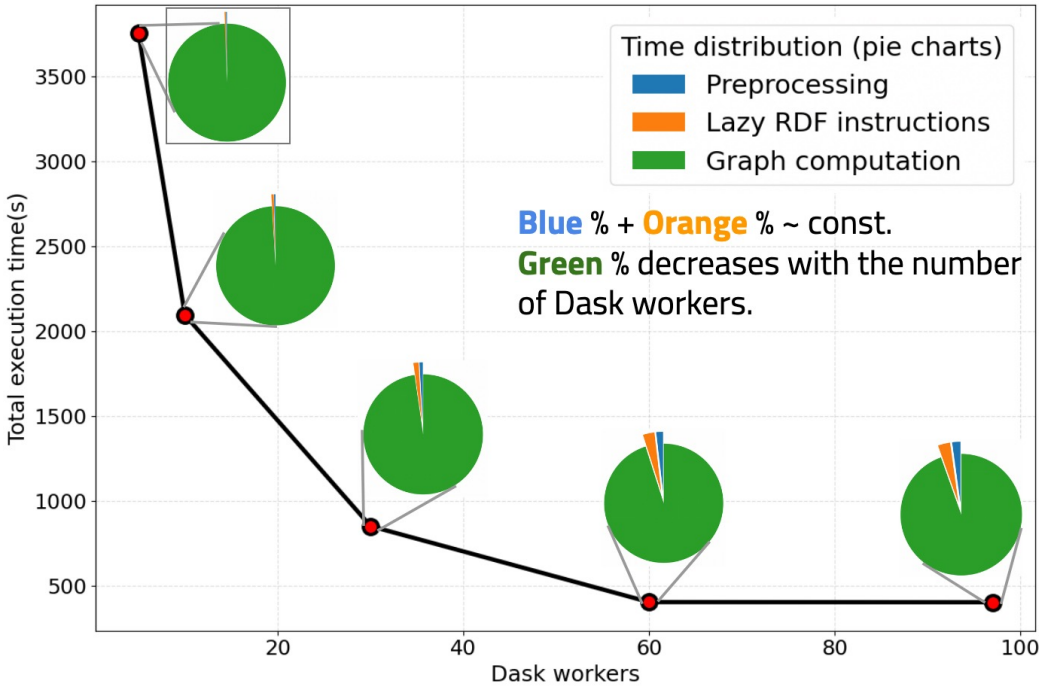
- The changes applied to the Tag-and-Probe program to run on the High Throughput Platform (distributed) do not affect performance, which is consistent with the original program (serial).

(efficiency x acceptance vs muon η)



Technical performance

- **Quasi-interactivity is now reached:**
  - Every time a re-execution of the analysis is needed (e.g. tweaking some thresholds or using different selection criteria), running a few Jupyter Notebook cells will do the trick (transparently accessing more resources)!!
  - This can result in a **great improvement** for any detector performance analysis application.



- Serial processing (as a single job on HTCondor)

Wall time: ~120 minutes

1 CPU on a AMD EPYC 7302 16-Core Processor, with 2GB memory

- Distributed processing on the platform:

Wall time: ~7 minutes

Up to 92 CPUs (46 physical), on **offloaded ICSC Naples HTCondor resources**

New result!

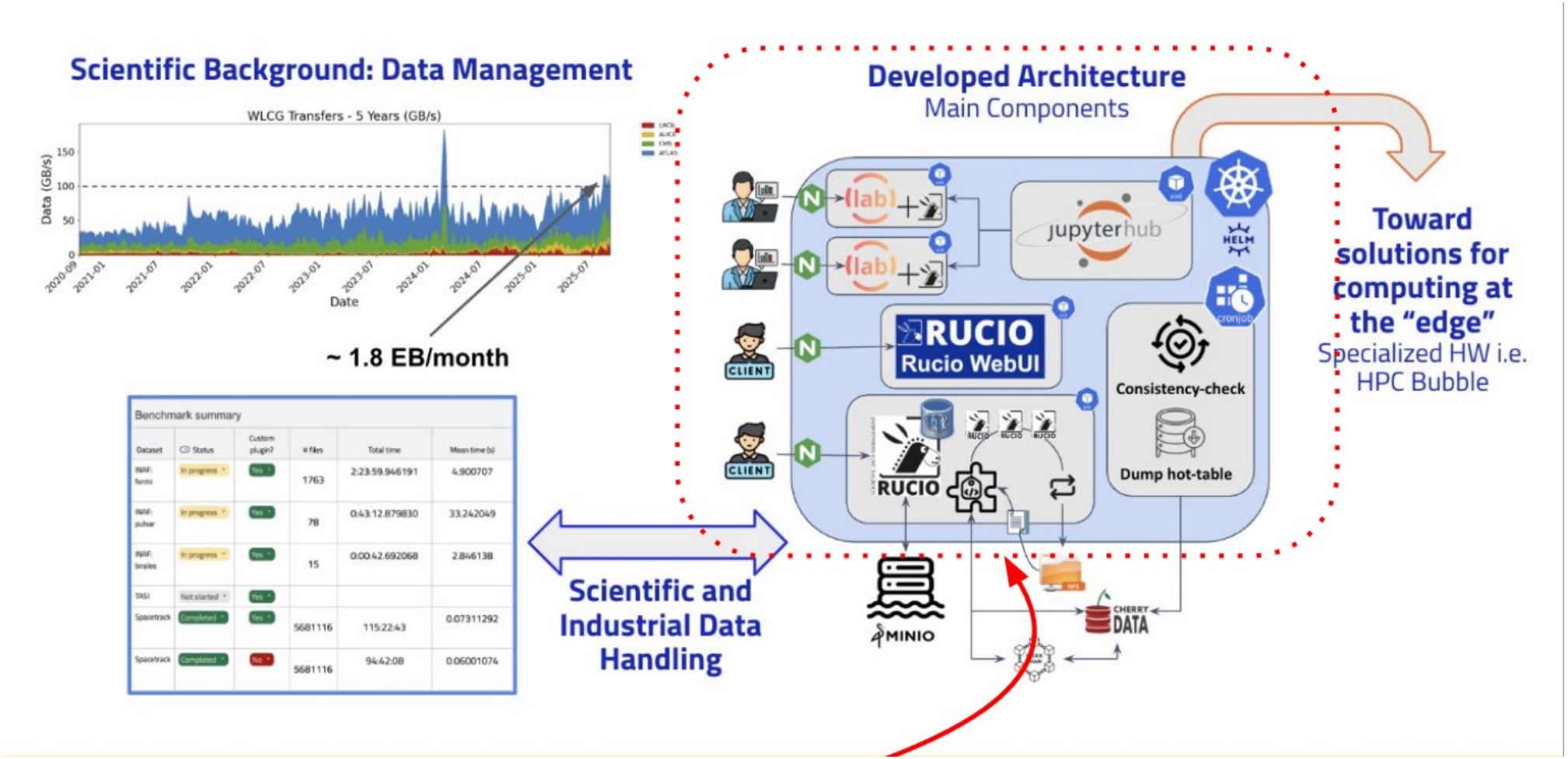
Conclusions

- The challenge presented by the next LHC phases requires a strong development effort of new tools, for making data analysis as efficient and as modern as possible;
- The «National Center for HPC, Big Data and Quantum Computing» is a unique opportunity for the creation of a modern infrastructure for research and industry in Italy;
- Aligned with CERN's R&D roadmap, a new **High Throughput Platform** has been developed:
  - Based on *interactive workflows* and on *declarative programming*;
  - Running on distributed resources (and heterogeneous).
- A CMS Detector Performance analysis has been used to benchmark such platform, obtaining identical results in terms of physics results.
  - From the technical point of view, a significant speed up is observed, thanks also to the new offloading capabilities granted by Interlink and ICSC resources;
  - This enables multiple analysis execution, critical aspect to assess detector stability during data taking.

This work is supported by ICSC – Centro Nazionale di Ricerca in High Performance Computing, Big Data and Quantum Computing, funded by European Union – NextGenerationEU.

BACKUP

Datalake, data management



Full stack running on ICSC resources hosted in INFN Cloud

DT local recontruction efficiency

The DT efficiency to reconstruct a local track segment was defined and measured using a Tag & Probe method.

Events were selected to contain a pair of oppositely charged reconstructed muons.

Muon tracks were required to be well reconstructed in the tracker detector (≥ 6 hits in the strip detector and ≥ 1 hit in the pixel detector) and to be well isolated in η and ϕ from other tracks. Moreover the muon tracks were required to have a separation between each other $\Delta R = \sqrt{\Delta\eta^2 + \Delta\phi^2} > 0.3$.

- To ensure that they come from the same interaction vertex, their distance at the point of closest approach to the interaction point should be $\Delta z < 0.1$ cm.
- Their invariant mass should be within 10 GeV of the Z_0 mass.

The track used as tag is also required to be well reconstructed in the muon detector, by satisfying the Tight-ID criteria described in JINST 13 (2018) P06015. Furthermore, it is required to have a transverse momentum $p_T > 27$GeV and also to pass the High-Level Trigger selection of isolated muons with $p_T > 27$GeV.

The inner component of track used as probe is propagated inside-out to each station of the DT detector and must have segments matched in ≥ 2 muon stations different from the one under study. It also must have $p_T > 20$GeV.

A DT chamber crossed by a probe track is considered efficient if a reconstructed segment is found within 15 cm distance of the extrapolated track in the R-ϕ plane.

The DT Segment Reconstruction Efficiency can be computed:

- within the full solid angle, in this case it also includes detector acceptance
- within fiducial regions i.e. discarding probes that cross a chamber within 15 cm of its edges.

LHC roadmap for Analysis Facilities

- Analysis facilities (AF) are one of the topics of the 2023 WLCG strategy document.
- All the R&D activities have been already packed in a comprehensive white paper (D. Ciangottini 🇮🇹)
- Since last year, LHCC has requested experiments to take part of such strategy and to provide a working direction:
  - Provide a list of questions that defines the expectations of AF from experiments.
  - LHCC will then ask all experiments to answer those questions (will happen ~November).
- In CMS, the O&C coordination area took part to this survey. The final proposal was refined and closed during the WLCG/HSF 2024 Workshop.

1. How does the experiment expect the Analysis Model will evolve for Run4 (and Run5) compared to Run3, considering both evolutionary advancements and potentially disruptive revolutions?
2. Please describe the data formats used today and in 2030 for analysis?
3. How much compute power is used today for analysis?
4. What are the main pain points that users experience today in analysis and how does the experiment plan to improve them in the coming years?
5. What is an Analysis Facility from the experiment point of view?
6. What are the capabilities and support models that you would need from the AF for them to be effective?
7. What is the motivation for deploying AF?
8. Describe the testing and benchmarking approach for the AF of the type(s) defined in Q5? How will their effectiveness and reliability be ensured? If benchmark analyses will be used, which ones?
9. List R&D that is being done to help address your concerns in questions Q1 and Q4?