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

High Rate Analysis benchmarks

Tommaso Tedeschi on behalf of WP5

ICSC & Spoke2 Where are we now? - 10/12 Dec 2024 - Catania

A recap - the context

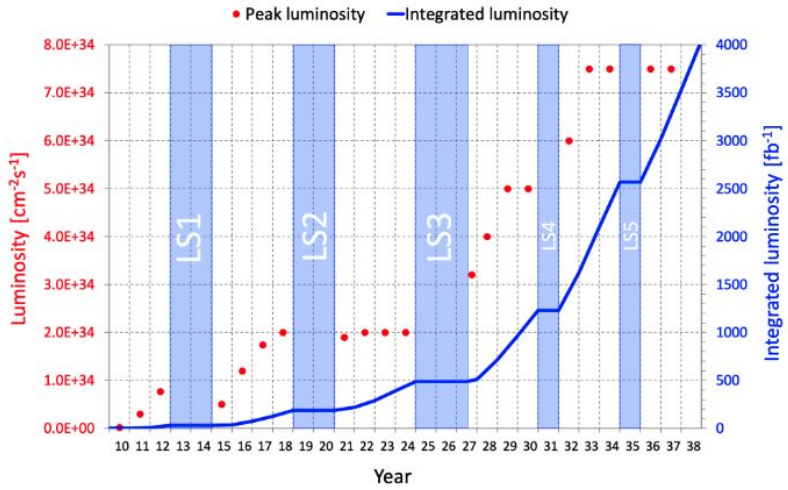
From 2029/30 onwards: huge increase in HEP experiments computing resources requests (HL-LHC above all)

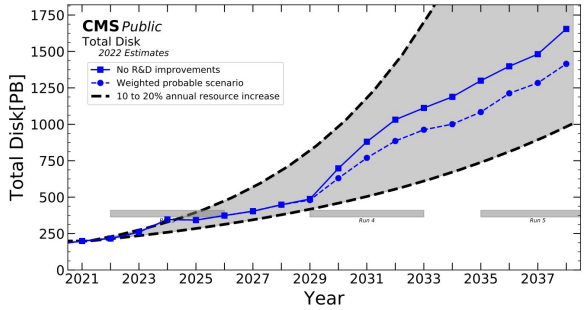
New analysis paradigm is arising: **high-rate, declarative, interactive or quasi-interactive data analysis approach**

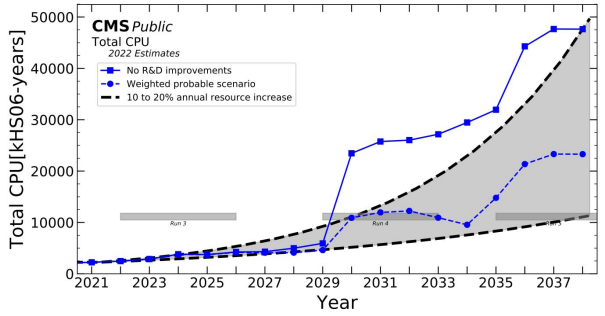
- enabled by the usage of slimmed (flat) data formats
- based on cutting-edge analysis tools (ROOT's RDataFrame, Coffea, …) which scale up thanks to industry-standard data science backends (Dask)

The development of infrastructural solutions (high rate platform) to implement such a new model is done **inside WP2 and WP5:**

- use case-driven approach and tests with real-world analyses
- **and synergically with Spoke0**:
  - adopting/proposing infrastructural solutions







A recap - Spoke2 high rate platform

As Spoke2 WP5 we are ready, using ICSC resources as per RAC allocation

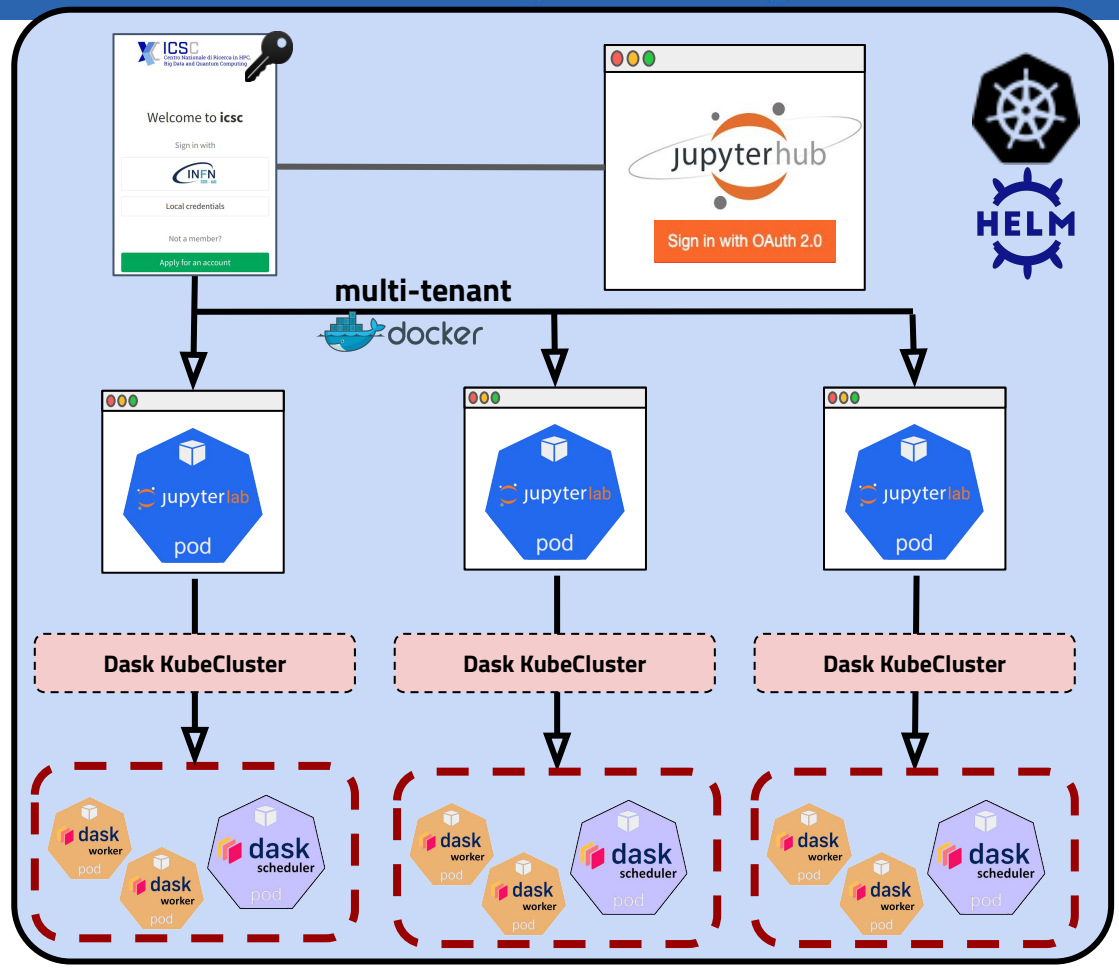
A jupyterhub deployed on a k8s cluster (128 vCPUs and 258 GB) via official Spoke2 JHub Helm repo

- endpoint is here and Indigo-IAM is used

Users choose JupyterLab image to deploy and they get access to a full IDE

Users can then scale up using Dask library, **within the cluster that possibly scale within the provider:**

- Dask Jupyterlab extension + Dask KubeCluster
- Interface with WLCG storage sites



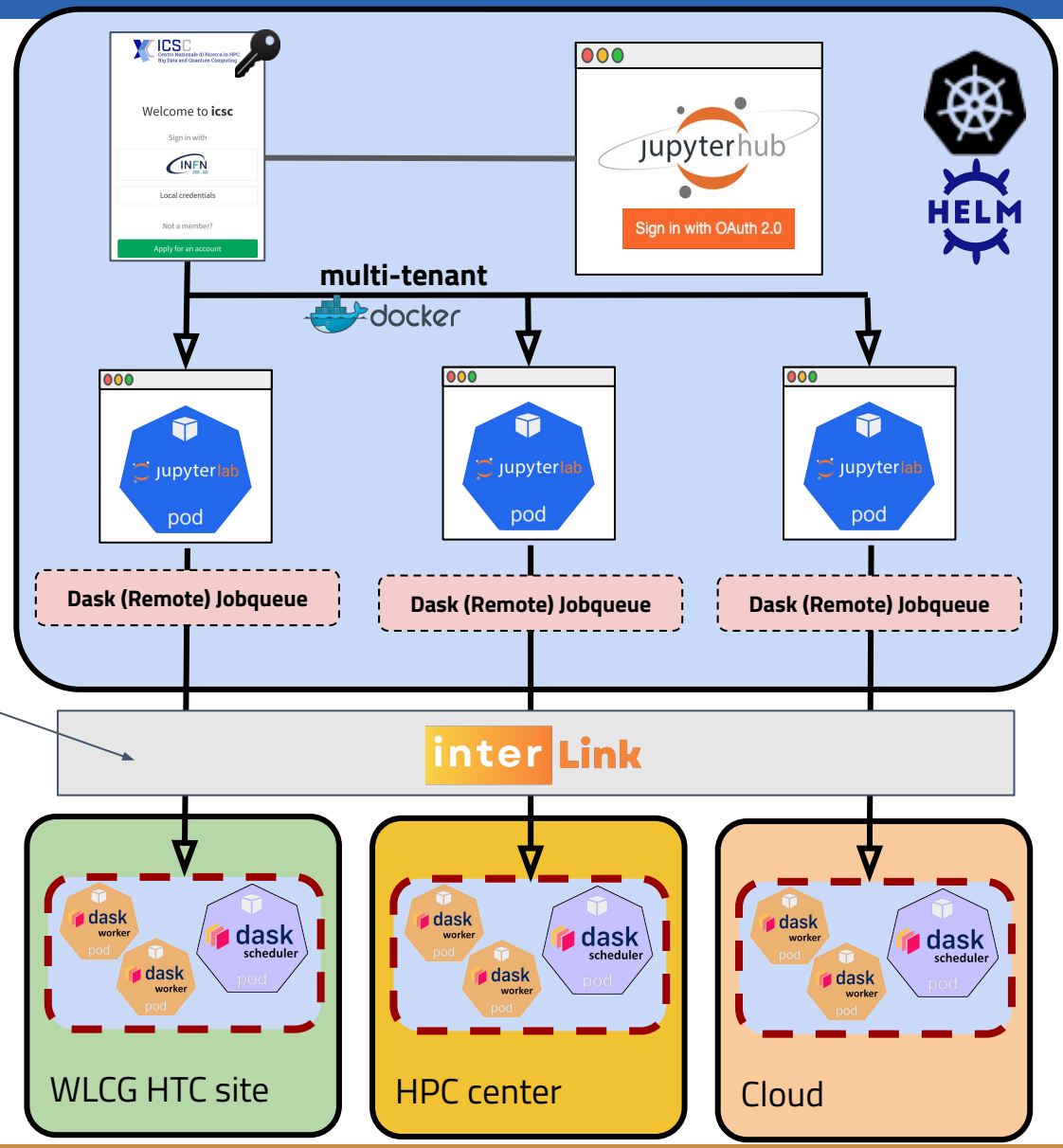
Potentially, **a huge amount of users with diverse use cases may join.. how to scale up to external resources?**

A recap - extend current high rate platform

Aim: enable the platform to **dynamically exploit all kinds of resources (HTC, HPC, Cloud)** transparently for the user

- looking for a synergy with active developments in this context, to delegate container execution on remote resources while keeping the very same user interface
- Possible solution: InterLink, which provides execution of a Kubernetes pod on almost any remote resource
  - Resources visible to the user thanks to an HTCondor overlay

CMS INFN Analysis Facility already implements this solution to integrate italian grid sites



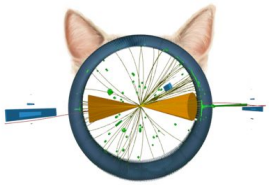
High rate analysis in CMS

INFN

Tools for achieving high throughput data analysis in the CMS experiment are discussed, developed and supported in the Common Analysis Tools (CAT) group

- it was established in 2022 in order to deal with a multitude of diverging tools, providing support and documentation, while moving towards efficiency, interactivity, and reusability of analyses
- It is charged with two main tasks
  - Take ownership of the development, maintenance and documentation of analysis tools of common interest
  - provide a forum to discuss developments of new analysis tools, offering guidance

CMS

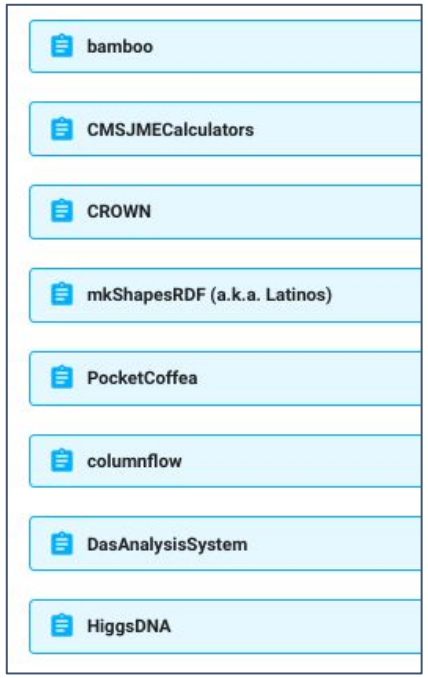


High rate analysis in CMS

INFN

CAT through its DPROC and WFLOWS subgroups is progressively taking ownership of already-existing **analysis frameworks** which are based on

- cutting edge tools:
  - **ROOT's RDataFrame (RDF)** - modern, high-level interface for analysis of data stored in TTree , CSV and other data formats, in C++ or Python
  - **HSF's Coffea** (or bare awkward-array) - makes use of uproot and awkward-array to provide an array-based syntax for manipulating HEP event data in an efficient and numpythonic way
- **NanoAOD** data format, most reduced CMS data format



Taken from CAT official docs

This ensures declarativeness, efficiency and (quasi-)interactivity of analysis, reducing time-to-insight

In collaboration with O&C, CAT represents the software needs of physics analyses in the context of the development of **analysis facilities** and other innovative computing infrastructures

In addition to CERN interactive logon services such as the LXPLUS service (Linux Public Logon User Service) and the SWAN (Service for Web based ANalysis) platform, other CMS institutions also provide access to computing resources by providing so-called Analysis Facilities to users with a CERN account that is associated with CMS. These are summarised here.

- Coffea Casa
- Purdue Analysis Facility
- INFN Analysis Facility

RDataFrame's 2017 SSWW VBS with tau

The feasibility of the interactive approach has been **tested and benchmarked on the CMS INFN Analysis Facility**

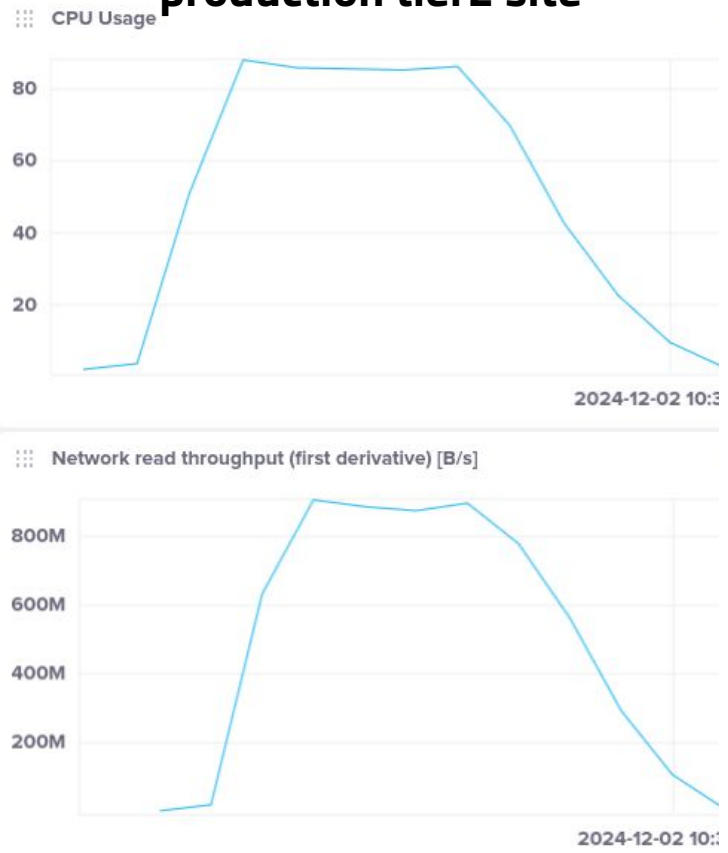
- The analysis with CMS detector of the scattering (VBS) of two same-sign W bosons decaying to a hadronic tau and a light lepton was taken as benchmark to test this approach:

$$q\bar{q} \to W^{\pm}W^{\pm}q\bar{q} \to \tau_h^{\pm}\nu_\tau\ell^{\pm}\nu_\ell q\bar{q}$$

The physics analysis was converted from a legacy iterative approach to an RDataFrame-based approach

We demonstrated a gain of one order of magnitude in terms of time with respect to a batch-like approach, see the dedicated paper [1]

Benchmark run on **96 CPUs at Legnaro production tier2 site**



Preprocessing on ~1 TB 2017 MC samples (filtering and corrections computation) **stored at Legnaro** done with user CPU usage at 80/90 % and network read throughput at 800/900 MB/s

PocketCoffea's 2018 Z->μμ

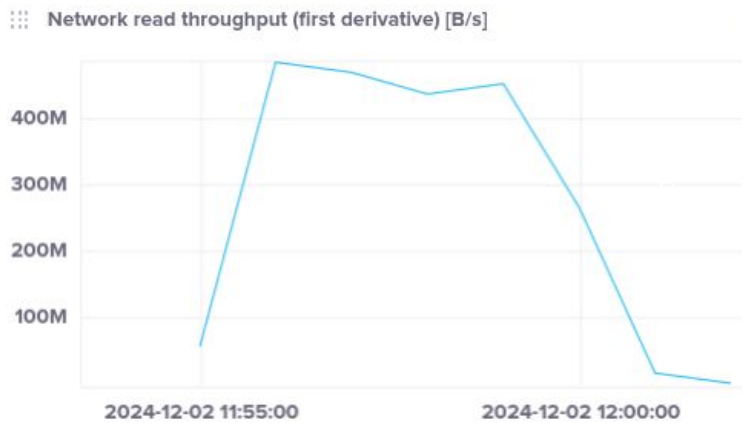
Joint effort with Davide Valsecchi (thanks!)

First tests with a Coffea-based analysis:

- 2018 Z->μμ using PocketCoffea configuration layer:
  - object preselections, skim, event selection, computation of basic scale factors with their variations and histogramming
- Dataset:
  - Data + DY: 1.2 TB
    - 1.2 bln events
    - **stored at Legnaro**
    - chunksize 4mln

Category	Events	Throughput (events/s)
Total	1180932962	3181431.42
Skimmed	822423975	2215608.81
Preselected	69283566	186649.81

Benchmark run **on the same 96 CPUs at Legnaro production tier2 site as RDF's benchmark**



With chunksize 4mln, user CPU usage at 80/90% and network read throughput at 400/500 MB/s:

- we are efficiently using the CPU, throughput is smaller wrt RDF's benchmark due to the different workload

PocketCoffea's 2018 TTBar flow

3 different benchmarks (run on **the same 96 CPUs at Legnaro** production tier2 site as RDF's benchmark) of increasing complexity:

Dataset: 2018 UL NanoAOD TTBar ~1TB, 476mln events, **stored at Legnaro**

Chunksize: 1mln

Small: objects and event preselection, scale factors with just 1 variation, no subsamples, 2 categories, 1 variation, 5 histograms

Medium: pobjects and event preselections, scale factors and their 7 variations, 3 subsamples, 4 categories, 5 histograms

Large: objects and event preselections, scale factors and their 7 variations, 3 subsamples, 14 categories, 5 histograms

CPU usage at 90%, throughput between 250-500 MB/s: the larger the workload, smaller reading throughput

Category	Events	Throughput (events/s)
Total	476408000	3080715.65
Skimmed	181836162	1175852.44
Preselected	63595247	411241.78

Category	Events	Throughput (events/s)
Total	476408000	2515867.48
Skimmed	181836162	960260.29
Preselected	63595247	335840.74

Category	Events	Throughput (events/s)
Total	476408000	1760817.70
Skimmed	181836162	672071.70
Preselected	63595247	235049.87

Conclusions and next steps

INFN

High throughput analysis is becoming a must due to the forthcoming HEP experiments resource needs:

- New paradigm based on declarativeness and efficiency: quasi-interactivity
- Python (ROOT's RDataFrame and Coffea) and Dask as the main players (but not the only ones)

The **high rate analysis platforms** (ICSC and CMS INFN) have proven to be able to support **diverse use cases**:

- CMS Coffea-based benchmark analysis has successfully been tested, adding up to the already tested ROOT's RDataFrame benchmark workflows:
  - **Both approaches reach throughput of O(100-1000 MB/s) with high CPU usage**:
    - This is an order of magnitude higher than what can be achieved via legacy (pure à la batch) approaches (10-100 MB/s)

Although this already represents a big step forward towards a really interactive analysis, even better performances can be achieved via high performance hardware **(hpc bubbles)**

Next steps:

- Add a real Coffea-based user analysis to corroborate results
- Test on high performance hardware

Backup

Key technology enabler: Distributed Dask

Dask.distributed is a centrally managed, distributed, dynamic task scheduler:

- The dask scheduler process coordinates **actions** of several Dask worker processes and the concurrent **requests** of clients

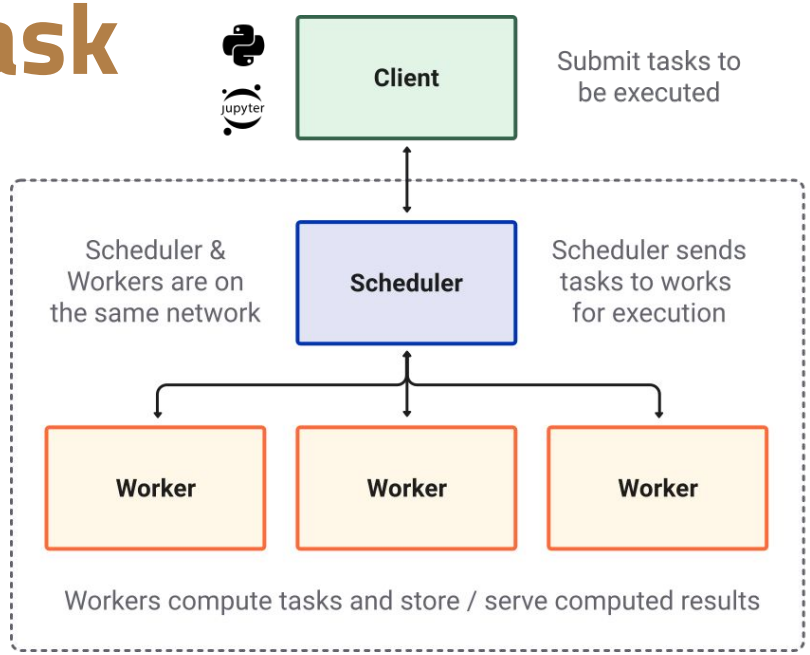
Users connects a local Python session to the scheduler and submitting work

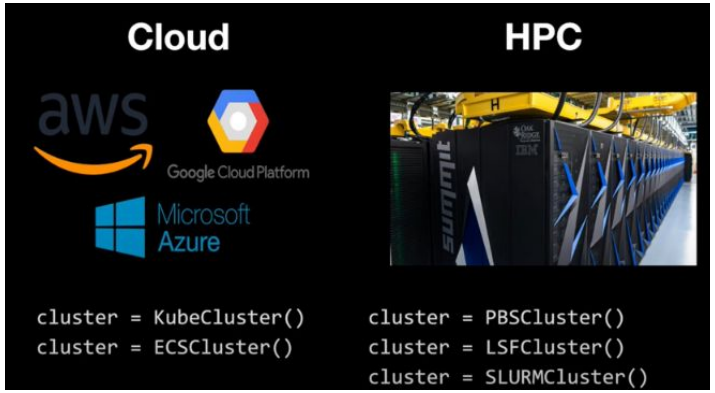
Best to use a **cluster manager** utility class which deploys a scheduler and the necessary workers

- **Dask-jobqueue** is a set of cluster managers for job queueing systems
- Supports PBS, Slurm, LSF, HTCondor, …

This obviously fits very well with the python analysis ecosystem, but also ROOT's RDataFrame can use Dask as backend:

- **Dask enables analysis on very different resource providers.. provided you get access to your data!**





…

High rate platform details

INFN

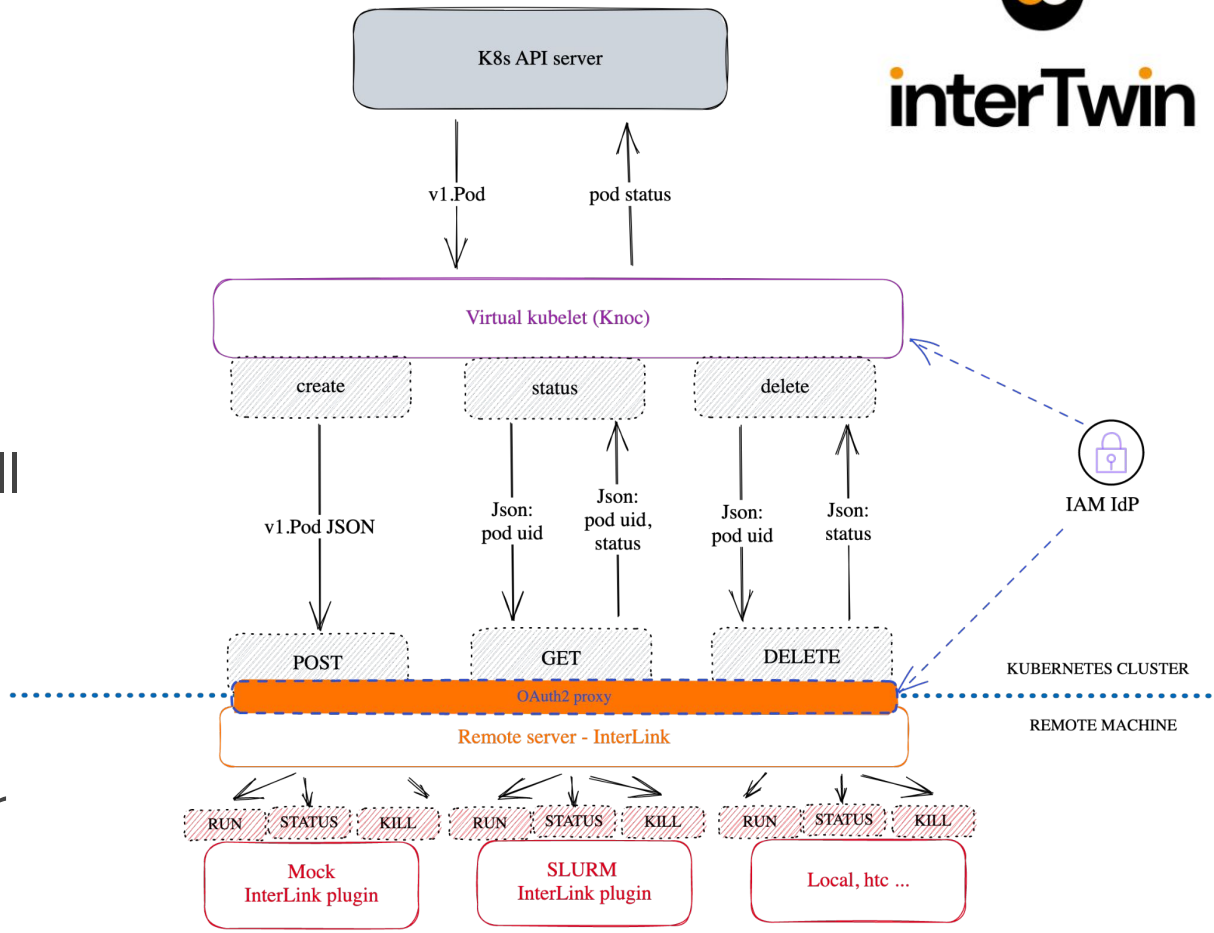
- **Access and security**: After connecting to an entrypoint URL, the user reaches a Jupyterhub [2] instance that, after authentication and authorization via INDIGO-IAM [3], allocates the required resources for the user's working area
- **User interface**: The user interface is based on Jupyterlab, customised 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)
- **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 [7] with multi-threading)
- **Deployment**: The deployment of the Kubernetes [4] resources needed for the spawning of this platform, is handled via HELM [5] charts available in the GitHub organization [6].This allows a seamless, flexible, scalable and fault-tolerant deployment on the available resources, with a limited impact on the admin's work time

A possible solution: InterLink

InterLink aims to provide an abstraction for the execution of a Kubernetes pod on any remote resource capable of managing a container execution lifecycle.

The project consists of two main components:

- **A Kubernetes Virtual Node**: based on the VirtualKubelet technology. Translating request for a kubernetes pod execution into a remote call to the interLink API server.
- **The interLink API server**: a modular and pluggable REST server where you can create your own container manager plugin (called sidecar), or use the existing ones: remote docker execution on a remote host, singularity Container on a remote SLURM or **HTCondor batch system**, etc...



https://github.com/interTwin-eu/interLink

InterLink: development context and ICSC related activities

INFN

The technical solution (interLink) has been initially prototyped by INFN in the context of the interTwin EU Funded project and is now enhanced within the ICSC development/research programme.

In particular

- **It is part of the Spoke0** infrastructural toolkits. As such it is under consolidation, testing and improvement
- **It is part of the Spoke2 - WP5 work plan**
  - in this respect there is a ongoing integration effort to extend the High rate analysis platform over HTC/HPC computing resources
- **Also part of the Spoke3 integration plan**
  - idea is to benefit of the interLink capabilities to offer highly dynamic access to specialized HW (i.e. over Leonardo)
  - integrating offloading with data retrieval from the data-lake prototype

Many fruitful sinergies should lead toward a generic technical solution, versatile and extendible based on specific needs.