First experiences with a portable analysis infrastructure for LHC at INFN

Diego Ciangottini[1], Tommaso Boccali[2], Andrea Ceccanti[3], Daniele Spiga[1], Davide Salomoni[3], Tommaso Tedeschi[1], and Mirco Tracolli[1]

[1] INFN Sezione di Perugia

[2] INFN Sezione di Pisa

[3] INFN-CNAF

Outline

- Our vision
- The approach
- The architectural pillars
- Data access for CMS experiment
- First experience with user workflows
- Results and challenges
- Conclusions and plans

Our vision

Simplify the setup for a new generation of multi-purpose facility for:

- Making a **typical LHC analysis workflow quicker w.r.t. a GRID based workflow**
  - supporting the majority of the analysis use cases based on **flat rootple/numpy-ish array**
  - compatible with day to day analysis development: **Interactive / quasi interactive**
- Transparent / “easy” **access to specialized HW**
  - looking forward to more ML-based analysis / workflow
    - E.g. starting with a typical signal-vs-background discriminator
- **Reproducible and scalable** environment capable to offload toward external resources (e.g. HPC, cloud)
  - Possibly abstracting away from the lower level infrastructure implementations
  - **Integration within the portfolio of INFN-Cloud** infrastructure
  - Offloading **intensive workflow to HPC (i.e. at CINECA)**

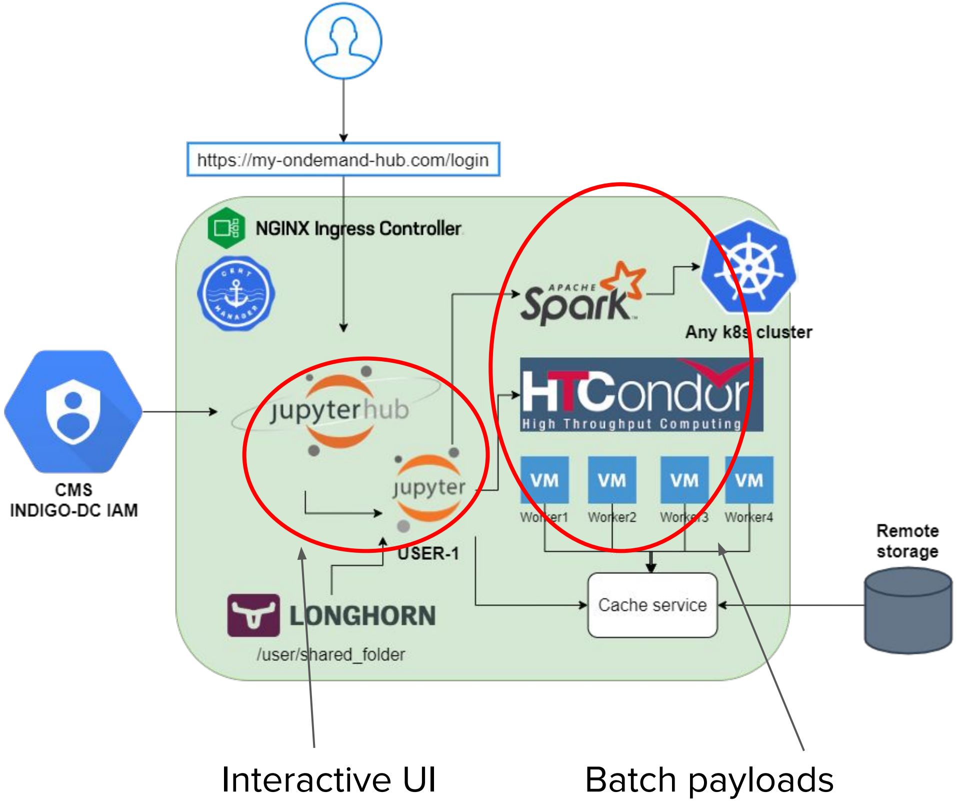
The approach

- Given the current variety of tools to manage and deploy container based infrastructure the aim has moved to **simplify the setup of such a facility on top of Kubernetes** (whether being it provided by commercial clouds or by on-demand and self-hosted solutions)
    - A **single machine equivalent deployment on Docker** is also available for situation in which a multi-node setup is not required
- Highly based on **service composition model**
    - Customize and re-use templates
        - Also for different experiment needs
    - Containers
    - Avoid technology lock-in
    - Declarative/template based approach

The architectural pillars

- **JupyterHub as the single entrypoint**
  - **Helm Charts + Helmfile** adopted as templating
    - Full integrability within the services portfolio of the INFN-Cloud
  - **Docker-compose** for single machine env
- **Token-based authentication** via Indigo-IAM
  - The access to compute and cache resources is managed via OIDC claims
- **Interactive and auto-scalable batch analysis** as an all-in-one solution

N.B. No CMS-specific parts here!
But rather a customizable base setup!

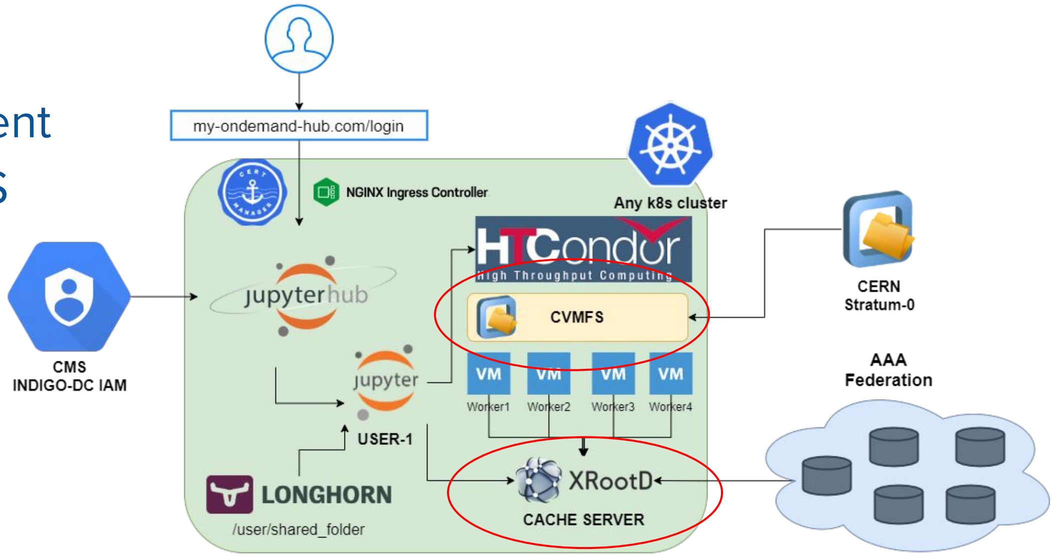


A key point: the data access

Based on experiment needs, **the setup can be customized** via Helm values thanks to the modularity of the component integration. E.g. data access for the CMS experiment deployment:

- The **experiment software** is shared through a repository hosted on **CVMFS**
- An **XCache server** configured **to interact with the CMS remote storage federation**

Work done in synergy with the ESCAPE EU project



- HTCondor on **K8s automatic scaling**
- Experimenting **Auth/N translation layer via XCache**

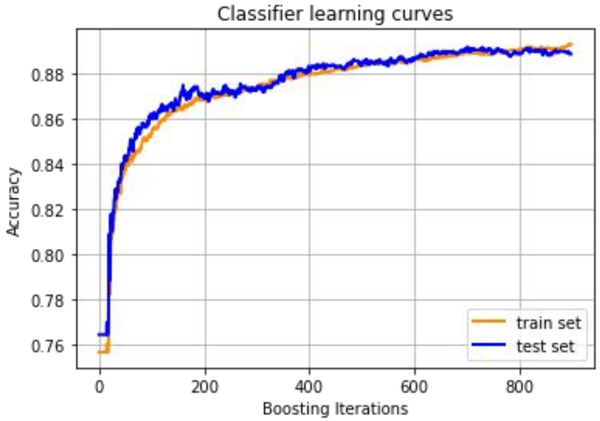
User-driven development

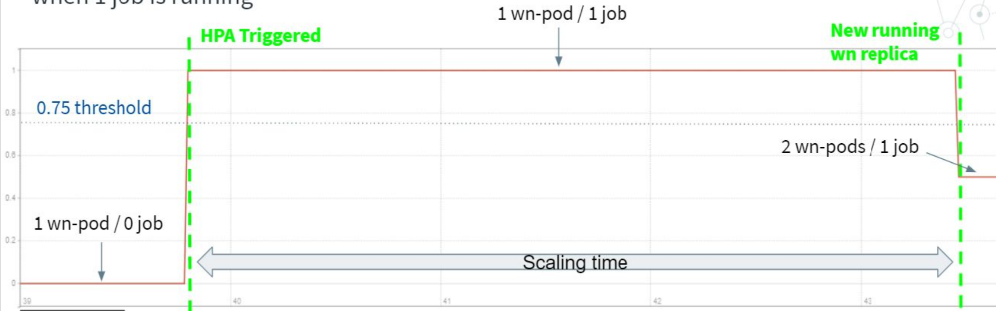
- Development **driven by the users' feedback**
  - **Started the commissioning of the prototype** with a real ongoing analysis at CMS
  - Integration with the workflow of "ssWW VBS with hadronic tau, mu/electron and two jets in final state"
    - PyROOT and NanoAOD-tools libraries for cut-based analysis steps
    - most used data-science and ML libraries for studies performed via Jupyter notebook
    - 7 TB NanoAOD dataset analyzed via integrated HTCondor batch
    - 3 TB skimmed data (flat rootples) inspected via interactive python-based analysis
- The first set of tests have **proven to provide users with access to an all-in-one solution:**
  - From the submission to HTCondor to the interactive python-based programming
    - **reducing the time to re-run a single step of the analysis**
- A **reduced overhead** from the user perspective comes also from the **adoption of the AuthN/Z model based on OIDC** w.r.t. the one currently based on X509

Current results and lessons

We learnt how to **satisfy a set of minimal requirements**:

- **Analysis validation**
  - Including a first ML discriminator
  - No problem reported about caching layer
- **Automatic scalability and resource optimization**





`avg(condor_slot_activity_busy)` as a function of time, scaling from 1 to 2 pods if when 1 job is running

Next challenges:

- Integration of a **node-level caching**
- **Dynamic offloading** of payloads to specialized (e.g. GPU) or opportunistic resources
  - based on system load

Conclusion and plans

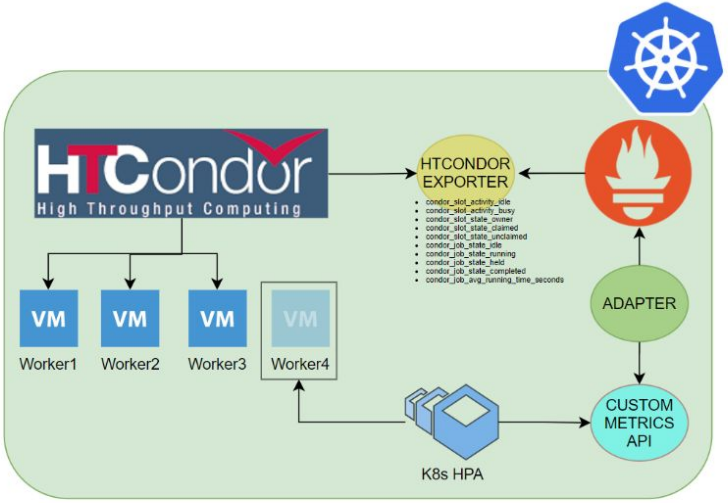
A **first experience** about providing an **analysis infrastructure for the physicist at INFN** has been made, now it's crucial to move toward an **evolution in terms of scale and integrations**:

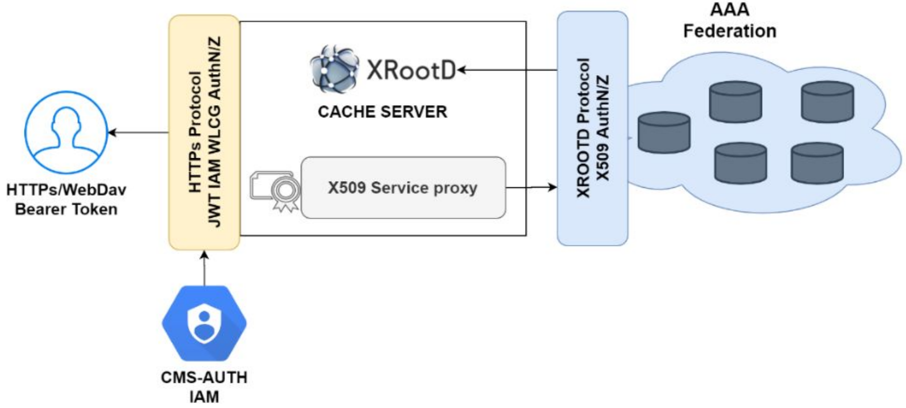
- Push further the integration, **starting a comprehensive test campaign at national level**
  - i.e. via INFN-Cloud resources, via HPC at CINECA resources
  - Opening to other testers/experiments by the end of the year
    - Helpful in tuning further the requirements
    - Comparison planned with same deployment on single machine setup
- **Transparent exploitation of heterogeneous hardware** and hybrid providers
  - e.g. scaling out toward Clouds and HPC
- **Measurement on the impact the cache layer** on high I/O workflows

BACKUP

Key features

- HTCondor on **K8s automatic scaling**
  - Autoscaling based on custom metrics of HTCondor Worker Nodes
    - Any metric coming from HTCondor queue can be configured as a trigger
- Experimenting **Auth/N translation layer via XCache**
  - Local cache auth/N via OIDC on client side, while x509 service proxy is used to fetch data from AAA federation
  - This makes the whole facility able to be almost X509-free


HTCondor
High Throughput Computing
HTCONDOR EXPORTER
condor_slot_activity_idle
condor_slot_activity_busy
condor_slot_state_owner
condor_slot_state_claimed
condor_slot_state_unclaimed
condor_job_state_idle
condor_job_state_running
condor_job_state_held
condor_job_state_completed
condor_job_avg_running_time_seconds
VM
Worker1
VM
Worker2
VM
Worker3
VM
Worker4
ADAPTER
CUSTOM METRICS API
K8s HPA


HTTPs/WebDav
Bearer Token
HTTPs Protocol
JWT IAM WLCG AuthN/Z
XRootD
CACHE SERVER
X509 Service proxy
XROOTD Protocol
X509 AuthN/Z
AAA
Federation
CMS-AUTH
IAM