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

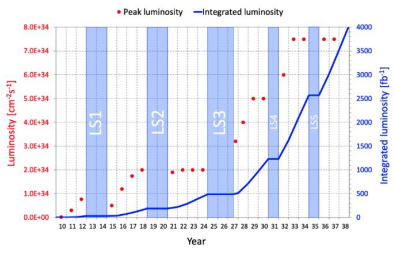
Continuous integration of analysis workflows on a distributed analysis facility

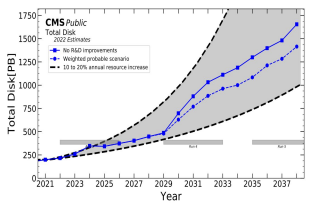
Matteo Bartolini on behalf of the CMS Collaboration

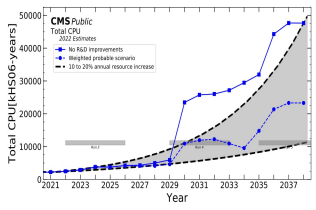
INFN
UNIVERSITÀ DEGLI STUDI FIRENZE
CMS

Motivation

The upcoming high-luminosity phase at the **CERN Large Hadron Collider (LHC)** and at future accelerator facilities will require an increasing amount of computing resources [1].







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

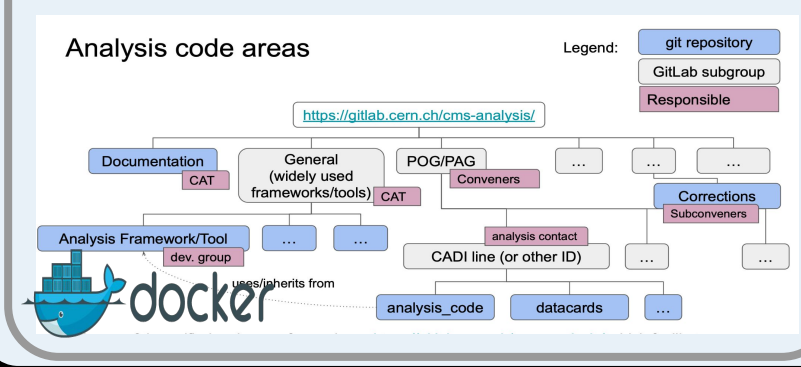
To better analyse this increasing amount of Big Data:

- Optimize the usage of CPU and storage;
- Promote the usage of better data formats;
- Enforce good practices for better reproducibility -> creation of the CAT group in CMS [6]

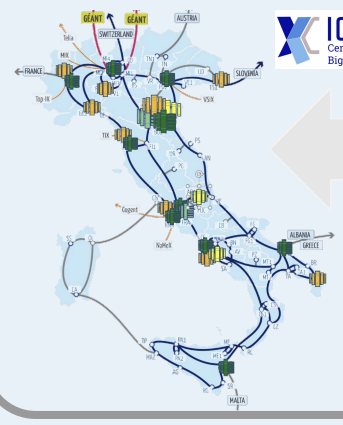
- New software based on declarative programming and interactive workflows;
- Distributed computing on geographically separated resources.

Improving reproducibility

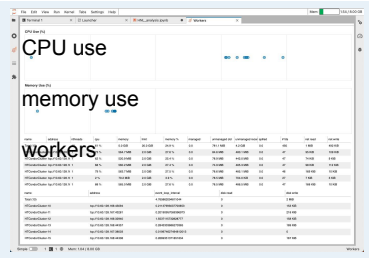
- Creation of the Common Analysis Tool group in CMS
- Common area where everyone in CMS puts their analysis
- Lives on GitLab
- Keeps the general framework separated from the analysis code
- Frameworks are containerized using Docker

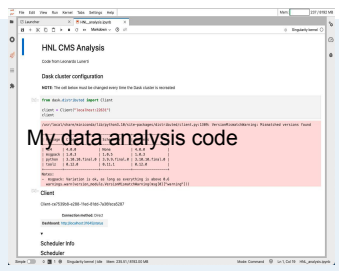


Distributed resources



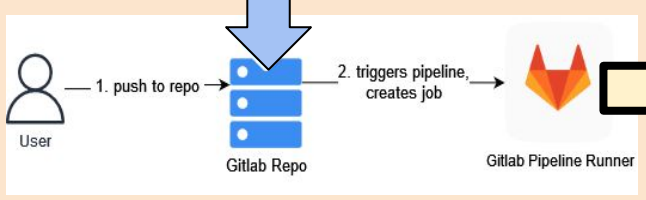
ICSC Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing





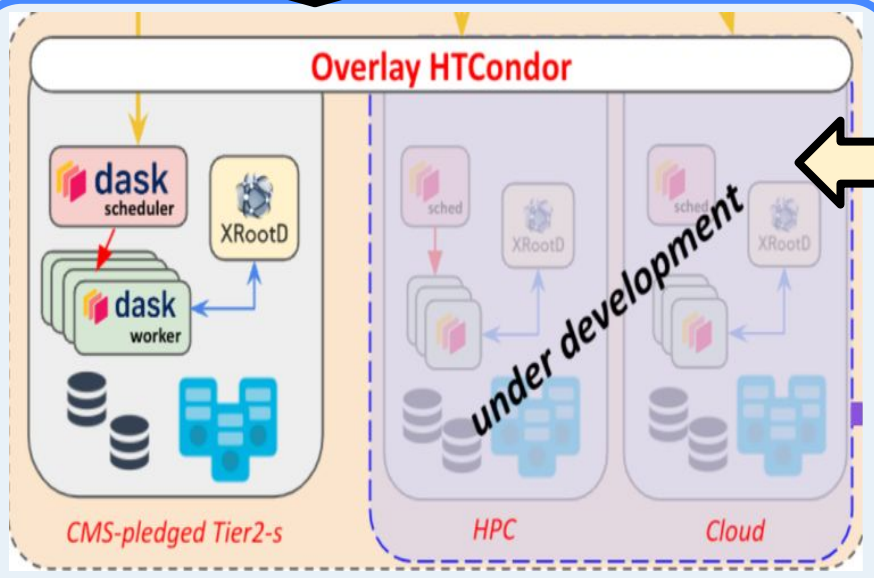
- The main idea is to develop analysis workflows where the users can have access to distributed computing on geographically separated resources in a transparent way
- The Italian ICSC is building a network of distributed resources to be made available for the entire scientific community

CI triggered analysis execution



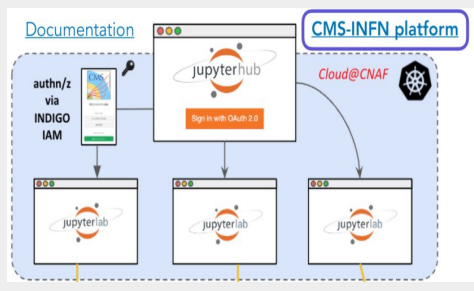
- The CAT framework relies heavily on CI/CD functionalities to automatically run your analysis at every new commit
- However, the CI is not suited for CPU heavy tasks, such as skimming of large datasets
- It is possible to leverage the backend infrastructure of the CMS Tier 2 cluster
- Offload the work by submitting the jobs in batches to the remote CMS HTCondor cluster
- Can easily extend this to other clusters in the future

- CAT service account with robot certificate is used to dynamically generate proxies
- Communication with EOS handled via XrootD
- Jobs to HTCondor are submitted as cmscat user
- The spawned remote workers load the containerized framework from the CAT repository making the whole process architecture-transparent
- Data stored on EOS are easily accessible by the worker via XRootD



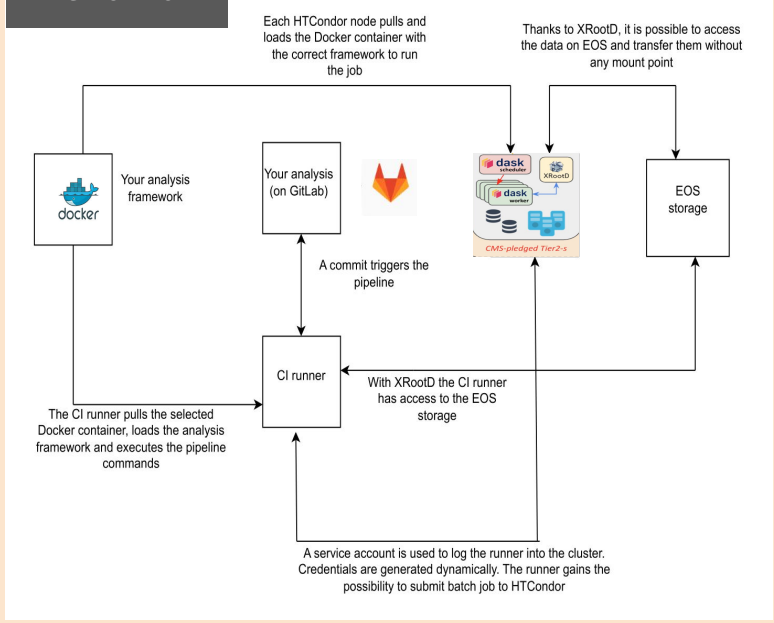
- On the CMS side the INFN has pledged Tier2 resources for testing purposes while the nationwide ICSC infrastructure is under development
- This has allowed for the creation of an experimental high-rate platform
- It also opens the door for the integration of these hardware resources into analysis workflows which are run on the GitLab CI
- CPU heavy pipelines can be created

The high rate platform

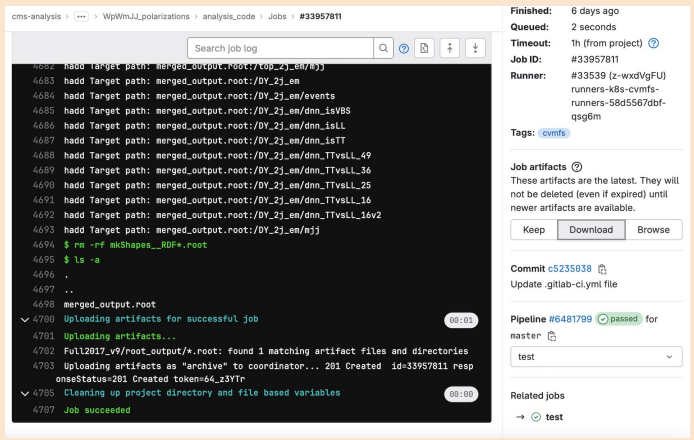


- Event selection and histogramming interactively with ROOT RDataFrame and JupyterLab [3],[4]
- Dask used as a backend to distributed the workload among the many cores in a way that is nearly transparent to the user
- First tests and use cases evaluated on the INFN CMS Analysis platform [2]
- For details on use cases see talk by Federica Simone

The workflow



- Once the jobs have been submitted to HTCondor, a loop keeps the pipeline running by checking the status of the jobs



- Once all the jobs are done running, the script will exit the loop and the outputs are transferred back to the CI runner
- The output files can be copied somewhere else, i.e EOS, or downloaded directly as artifacts from the CI

References

1. https://twiki.cern.ch/twiki/bin/view/CMSPublic/CMSOfflineComputingResults
2. T. Tedeschi et al. Prototyping a ROOT-based distributed analysis workflow for HL-LHC: The CMS use case, Computer Physics Communications, Volume 295, 2024, 108965.
3. https://root.cern/doc/master/classROOT_1_1RDataFrame.html
4. https://jupyterlab.readthedocs.io/en/latest
5. https://docs.dask.org/en/stable/
6. The CAT framework
7. HTCondor