Quasi interactive high throughput analysis of high energy physics data

M. Bartolini(3)(*), A. Cagnotta(8), T. Diotalevi(1), A. D'Onofrio(8),
F. Giuseppe Gravili(4), F. Maria Simone(6), P. Mastrandrea(7),
M. Numan Anwar(6), G. Sabella(8), B. Spisso(8), A. Tarasio(5), and T. Tedeschi(2)

(1) *Università di Bologna e INFN, Sezione di Bologna - Italy*
(2) *INFN, Sezione di Perugia - Italy*
(3) *Università di Firenze e INFN, Sezione di Firenze - Italy*
(4) *Università del Salento e INFN, Sezione di Lecce - Italy*
(5) *Università della Calabria e INFN Laboratori Nazionali di Frascati - Italy*
(6) *Politecnico di Bari e INFN, Sezione di Bari - Italy*
(7) *INFN, Sezione di Pisa - Italy*
(8) *Università di Napoli e INFN, Sezione di Napoli - Italy*

Summary. — In this article a new quasi interactive platform for high throughput data analysis will be presented. The platform will leverage the cloud resources provided by ISCS (High-Performance Computing, Big Data e Quantum Computing Research Centre). An overview of the technologies used and preliminary results of some real use case will be shown.

1. – Introduction

Data analysis in the field of High Energy Physics presents typical big data requirements, such as the vast amount of data to be processed efficiently and quickly. The Large Hadron Collider in its high luminosity phase (HL-LHC) will produce about 100 PB/year of data, ushering in the era of high precision physics. On one hand the high luminosity phase will require ever growing computational resources, on the other hand it is desirable to find new paradigms and models that take advantage of these new resources, once available. Thanks to funding from the Italian National Recovery and Resilience Plan (PNRR), the High-Performance Computing, Big Data e Quantum Computing Research (ICSC) Centre was established [1]. Its activity will be dedicated to the creation and maintenance of new heterogeneous distributed resources. This new infrastructure will represent a unique strategic asset not only for Italy, but for the international scientific

(*) Speaker

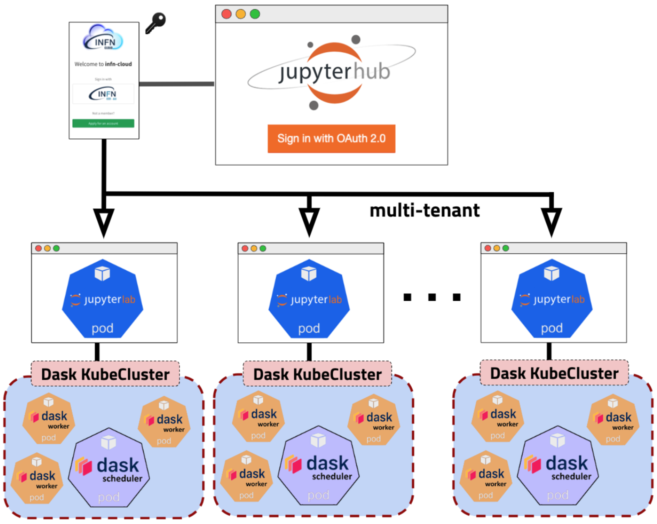

Fig. 1. – Schematization of the high-rate platform. When the user logs in an instance of JupyterLab is created and underlying hardware resources assigned on a fair-share basis.

community as well. Within the HEP([1]) community this new infrastructure will allow to profit even more from the use of modern data analysis tools, such as ROOT RDataframe [2], which implement multi-process and other low-level optimisations in a way that is transparent to the user and easily scalable to the available cores. The foreseen increase in the available bandwidth for I/O([2]) operations will also enable analysts to integrate in their analysis workflow tools designed for fast, low latency data access.

2. – The high-rate platform

To leverage the new resources described in the section above an effort to develop a user-friendly high-rate platform is ongoing. A simple sketch is shown in Fig 1. The analysts will be able to spawn their own JupyterLab session on a Kubernetes cluster by logging into a JupyterHub. Kubernetes [3] ensures an efficient and transparent orchestration of the containers by starting the Jupyterlab pods and managing the Dusk cluster via Kubecluster [4]. Moreover, the JupyterLab interface is highly flexible and customizable: the user can write his own code and process large datasets in parallel on the Dask cluster[5]. The Dask scheduling system hides complexity away from front end users by deploying the execution on distributed hardware. By using Docker containers [6], it will be possible to easily set up working environments tailored to the specific analysis use case. It will also be possible to have real time access to the monitoring of the Dask cluster, thanks to plug in such as Dask Labextension [7]. In the next section a few use case, which are currently in R&D([3]) state, will be discussed.

([1]) High Energy Physics
([2]) Input/Output
([3]) Research & Development

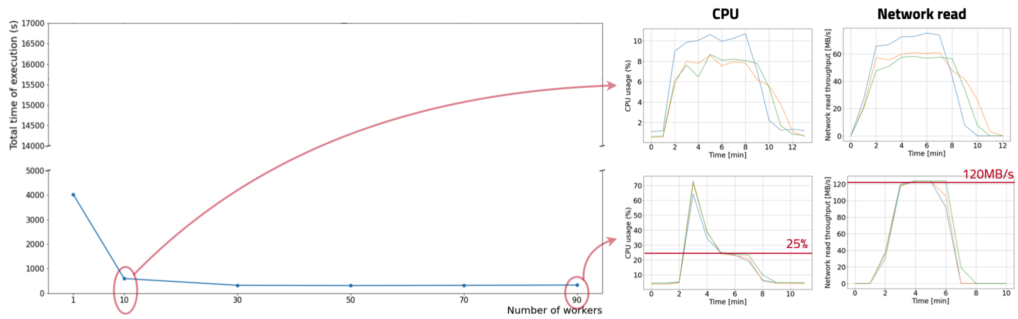

Fig. 2. – Execution time as a function of the number of workers when running the entire analysis workflow on the CMS b-parking dataset. The flattening of the curve at high number of workers is due to the bottleneck of the limited bandwidth.

3. – Use cases with HEP data

There are different use cases where gains in processing time are being evaluated. As a practical example a test within a CMS [8] Experiment physics analysis is reported. This particular analysis is implemented in RDataFrame and using its distributed features compatible with the Dask backend. This test was performed on a 69 GB data set from the CMS b-parking stream[9]. The workflow includes a series of physical selections over the input files and finally a creation of various histograms as output. This computation is translated into a computational graph model by RDataFrame, which integrates seamlessly with distributed architectures. This feature enables efficient distributed data processing and scalable performance with the number of workers available. Looking at Fig. 2 we can indeed see that the execution time decreases as a function of the number of node workers. However, it starts saturating above 30 due to a limited network bandwidth, which becomes the bottleneck, as the test was carried out using a testbed site at the CMS Tier2 datacenter in Legnaro [10]. This underlines the importance of a high network bandwidth at the server side if one is to scale performances. A second use case is the simulation of $Z \to e^+e^-$ process at FCC([4]). These are considered benchmarks for detector studies. In this case the events were simulated with EDM4hep [11] and detector effects were simulated using a gaussian smearing. Just like the previous example the pre-processing and data skimming was done using RDataFrame. Fig. 3 shows the execution time as a function of the amount of data generated and the number of workers used. In this case the test was carried out using a local high rate platform from INFN. Even this case the execution time seems to be scaling well with the number of workers. The national computing infrastructure may also be used as the backend support of the GitLab CI [12] (Continuos Integration) for a easy access to distributed resources for CPU-intensive calculations. Currently, most of the time the CI is used to run simple and short pipelines due to the limited hardware resources. Thanks to the new infrastructure, it will be possible to have CI pipelines where the submission of long batch jobs will be offloaded to the resources of the new computing infrastructure. In this way analysts will be able to quickly run different tests on their data directly with the Gitlab CI, perform different analyses in parallel and, at the same time, keep track of all the changes made.

([4]) Future Circular Collider

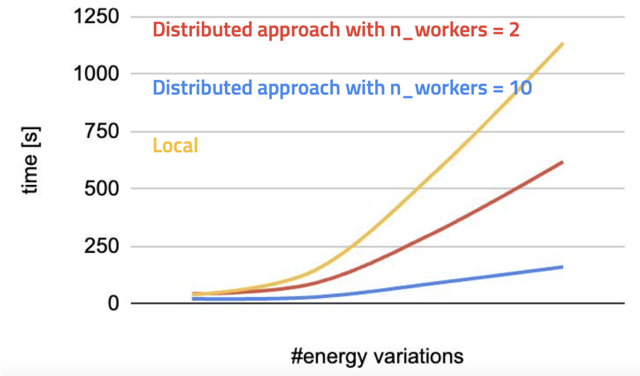

Fig. 3. – Execution time as a function of the amount of data generated and the number of workers used in the simulation of $Z \to e^+e^-$ process at FCC.

4. – Conclusions

The HL-LHC and Future Colliders represent an unprecedented challenge that should prompt HEP experiments to rethink their computing models. The interactive analysis approach shown in this document looks promising and it will greatly benefit from the upcoming distributed resources. Different use cases to evaluate the improvements that this new approach will potentially bring are being developed.

* * *

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

REFERENCES

[1] ICSC, `https://www.supercomputing-icsc.it/en/icsc-home/`
[2] RDataFrame, `https://root.cern/doc/master/classROOT_1_1RDataFrame.html`
[3] Kubernetes, `https://kubernetes.io`
[4] Kubecluster, `https://kubernetes.dask.org/en/latest/operator_kubecluster.html`
[5] Dask, `https://docs.dask.org/en/stable/dataframe.html`
[6] Docker, `https://www.docker.com/get-started/`
[7] DaskLab, `https://github.com/dask/dask-labextension`
[8] S. Chatrchyan *et al.* (CMS Collaboration), *The CMS Experiment at the CERN LHC*, JINST, vol. 3, pp. S08004, 2008. doi:10.1088/1748-0221/3/08/S08004.
[9] Swagata Mukherjee, *Data Scouting and Data Parking with the CMS High level Trigger*, PoS **EPS-HEP2019** (2020) 139. doi:10.22323/1.364.0139
[10] S. Badoer *et al.*, *Journal of Physics: Conference Series*, vol. 513, p. 032090, 2014.
[11] EDM4hep, `https://github.com/key4hep/EDM4hep`
[12] Gitlab, `https://docs.gitlab.com/ee/ci/`