A cloud-based computing infrastructure for the HERD cosmic-ray experiment

N Mori[1], D Ciangottini[2], M Duranti[2], V Formato[3], D Spiga[2]

[1] INFN sezione di Firenze, via G. Sansone 1, I-50019 Sesto Fiorentino, Italy

[2] INFN sezione di Perugia, Via A. Pascoli, I-06123 Perugia, Italy

[3] INFN Sezione di Roma Tor Vergata, via della Ricerca Scientifica 1, I-00133 Roma, Italy

E-mail: mori@fi.infn.it

Abstract. The HERD experiment will perform direct cosmic-ray detection at the highest ever reached energies, thanks to an innovative design that maximizes the acceptance, and its placement on the future Chinese Space Station which will allow for an extended observation period.

Significant computing and storage resources are foreseen to be needed in order to cope with the necessities of a large community driving a big experimental device with an energy reach above PeV for hadrons and multi-TeV for electrons and positrons. For example, at PeV energies Monte Carlo simulations require a massive amount of computing power, and very large simulated data sets are needed for detector performance studies like electron-proton rejection.

The HERD computing infrastructure is currently being investigated and prototyped in order to provide a flexible, robust and easy to use cloud-based computing and storage platform. It is based on technical solutions originally developed by the "Dynamic On Demand Analysis Service" (DODAS) framework in the context of projects such as INDIGO-DataCloud, EOSC-hub and XDC. It allows to seamlessly access both commercial and institutional cloud resources, in order to efficiently make use of opportunistic resources to cope with high-demand periods (like full dataset reprocessings and specialized Monte Carlo productions), as well transparently integrate with on-premise computing resources managed by an HTCondor batch system. The cloud platform also allows for an easy and efficient deployment of services for the collaboration like calendar, document server, code repository etc. making use of available, free open source solutions. Finally, an Indigo-IAM instance provides a Single-Sign-On service for access control for the whole infrastructure.

1. Introduction

The availabilty of computing resources is one of the most important limiting factors in determining the physics reach of modern High-Energy Physics (HEP) experiments. Each physics result needs a given amount of data processing, and the total available computing power effectively sets a limit on the explorable physics. In a global scenario where the demand for computing power grows faster than the availability there is a significant effort towards efficiency as a way to mitigate the issue. The optimization (especially for multithreading) of computing algorithms and the usage of heterogeneous architectures like GPUs and FPGAs are two of the most successful strategies towards efficiency, and are actively pursued by all the major collaborations.

Still, there are some application domains for which these techniques are only partially exploitable, and that are thus quite bound to a "conventional" computing approach.

For example, parametrized fast simulations might not precisely reproduce the rare events constituting the irreducible proton background of electrons sample selected by means of topological shower analisys in highly-segmented calorimeters. Full simulations are usually needed to accurately reproduce this class of events, and current Monte Carlo toolkits like Fluka and Geant4 run on CPU for a large extent and thus cannot e.g. fully take advantage of heterogeneous systems yet. In a calorimetric high-energy experiment for direct cosmic-ray detection this translates into the need of fully-simulating particle showers at energies exceeding the PeV scale using just "conventional" CPU resources. While software optimizations can still play a major role in squeezing some extra performance, it seems clear that adopting a computing model that can easily profit of opportunistic resources providing additional computing power is a mandatory requirement for some experiment categories (and obviously also a desirable feature for others). This can also help in coping with high-demand periods that frequently happen e.g. in relatively small collaborations due to fluctuations of the numebr of active persons or for experiments whose data flow is not steady.

In the following, the desing and initial prototyping of the computing model for the High-Energy cosmic-Radiation Detection (HERD) experiment is presented. HERD [1] is a next-generation calorimeter for direct detection of cosmic-rays from space, aimed at measuring the hadronic component of the spectrum up to energies of the order of PeV per nucleon, and the electron+positron component at tens of TeV. The proposed model is based on the cloud approach which can provide the flexibility and scalability described in the previous paragraph. The adopted technical solutions are based on those developed by the DODAS project [2] and are implemented on the INFN Cloud [3] infrastructure.

The cloud platform also provides support for solutions to peculiar needs of the HERD collaboration; for example, the lack of an umbrella organization (like CERN for the HEP community) providing services accessible to all the collaboration members can potentially lead to fragmentation, with each participating institute providing its own services and the members needing to have multiple accounts on different institutional infrastructures. On the other hand, free services from commercial providers like Google are not generally available e.g. for the Chinese members. Furthermore, commercial solutions might lead to vendor lock-in situations that could generate management issues (especially for paid services), while a portable, free, open source solution that can be deployed everywhere is more desirable. An implementation of a set of services like calendar, file sharing etc. managed at collaboration level on cloud resources is briefly described at the end of this paper, showing the potential of the cloud approach also for this aspect.

2. General architecture

The prototype infrastructure is sketched in fig. 1, and is based on two pillars: a centralized authentication and authorization (AuthN/Z) service, and a cloud storage. These two services constitute the backbone of the system, dealing with critical issues like access control and data persistence. They are used by the data processing resources and by all the collaboration services, providing an environment where a Single-Sign-On (SSO) mechanism at collaboration level grants access to all the experiment resources including the storage.

3. Authentication and authorization

AuthN/Z is provided by a dedicated INDIGO-IAM [4] instance for HERD (IAM-HERD). User login is managed through a SAML federation with the Identity Providers (IdP) of the most representative institutes participating to the experiment; support for other institutes is provided through the eduGAIN [5] federation. Authentication on the different clients (i.e. experiment services and resources) is done via OpenID Connect (OIDC) [6] tokens. A side benefit is that

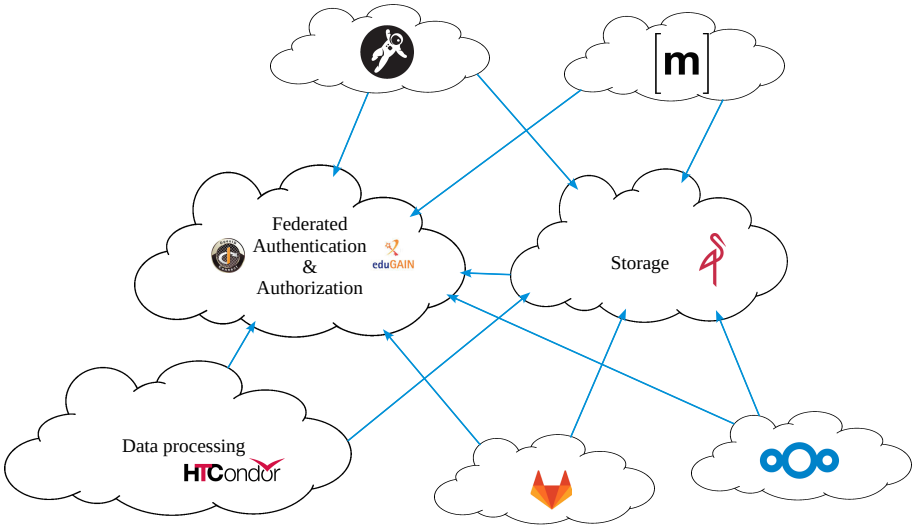

Figure 1. Overview of the general architecture. AuthN/Z and storage provide the foundation services over which the whole system (e.g. data processing and collaboration services) is deployed as cloud-based services (possibly on different cloud resources).

HERD members can access all the collaboration resources with the credentials from their home institute.

4. Storage

A testbed for cloud storage has been deployed in terms of a MinIO [7] cluster (MinIO-HERD). MinIO is an "enterprise-grade open source object storage" compatible with Amazon Simple Storage Solution (S3) [8]. S3 is a well-established cloud technology, and the S3 protocol is widely supported by cloud-native applications. Native S3 storage backends are available and used for all the applications described in the following sections.

The MinIO-HERD testbed cluster is made of 4 instances of MinIO running on cloud Virtual Machines (VM) providing the disk space to the cluster. In turn, disk space is made available to VMs as Ceph [9] filesystems from a central storage pool located at the INFN CNAF data center in Italy. The total amount of available raw space is 100 TB, and is provided by the DICE [10] project. Storage has been configured as a 3+1 erasure coded pool with host failure domain, for a total available space of about 67 TB. User authentication for cloud native tools like rclone is based on OIDC tokens retrieved with the oidc-agent [11] helper, while application access (e.g. for backend storage or backup snapshots) is managed via access/secret key pairs. This solution ease the user access via IAM account while making the application access robust in case of IAM temporary unavailability.

5. Data processing

The architecture of the data processing infrastructure and its interactions with other elements is depicted in fig. 2. The core entity is a High Throughput Computing (HTC) cluster managed by HTCondor [12], automatically deployed on-demand by the INFN Cloud orchestrator as a Kubernetes [13] cluster. The deployment includes computing nodes running containerized execution environments, the HTCondor schedd and additional services like the Kubernets management console.

Users access the cluster by means of a HTCondor client environment running on user's machine (e.g. laptop) as a Docker [14] container. User AuthN with HTCondor is done via

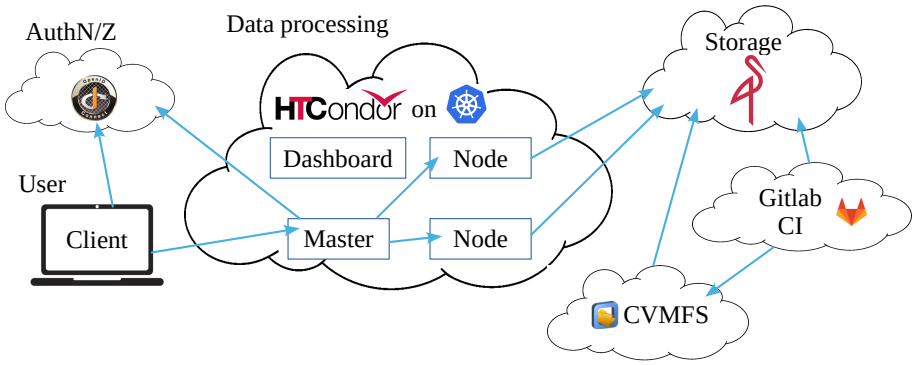

Figure 2. Overview of the data processing architecture

OIDC.

The distribution of the collaboration software is done with the CernVM-FS (CVMFS) [15] distributed file system. The current setup consists of a VM running containerized CVMFS Stratum0 + publisher services with S3 storage backend on MinIO-HERD. CVMFS clients directly interfaces with MinIO-HERD via http, so actually the Stratum0 and the publisher are used only to commit modifications to the software repository. CVMFS clients are mounted on the computing nodes VMs and then bind-mounted in the execution environments, taking advantage of CVMFS caching at host level. The software deployment on CVMFS has been fully automated by means of Gitlab [16] Continuous Integration / Continuous Deployment (CI/CD). The HERD Gitlab instance (see sect. 6) is used for hosting the source code and for managing the development workflow of the collaboration software; CI/CD pipelines have been set up to automatically build and install to CVMFS the master version and the tagged releases, for three different supported Linux distributions.

A set of helper scripts distributed by the HERD collaboration are used to submit the data processing jobs to the schedd from the client environments. The computing jobs make use of the node-local scratch storage area for disk I/O during the job execution. Input/output files are transferred from/to the S3 storage to/from the scratch area at the beginning/end of the job by means of pre-signed URLs retrieved from MinIO-HERD by the helper scripts at job submission time using the boto3-sts [17] library and the user's credentials (token). User's code libraries (currently of size of few tens of MB) and job configuration files are uploaded from the client environment to the schedd at job sumbission time using HTCondor transfer. The whole job submission and execution process can be summarized as follows:

- The user prepares code libraries and job configuration files on the client environment
- The helper scripts retrieves pre-signed URLs for input/output files and prepare a job script containing pre-/post-job file transfer from/to S3
- The helper scripts submit the job script to the schedd together with user's libraries, configuration files and pre-signed URLs
- The runtime environment:
  - transfers the input files from S3 to the scratch storage using the pre-signed URLs
  - launches the data processing executable (distributed via CVMFS with the collaboration software) which processes the input file from the scratch storage and writes the output to scratch storage as well
  - transfers the output files from the scratch storage to S3 using the pre-signed URLs

The data processing infrastructure has been tested with real HERD simulation, reconstruction and data analysis workloads, and meets all the requirements in terms of functionalities.

6. Collaboration services

The cloud approach offers the possibility to self-implement a set of web services at collaboration level, thanks to flexibly-provided resources and the availability of (almost) ready-to-use, free, open-source solutions packaged container images. This aspect has been pursued to develop a generic, open and reusable model that meets the HERD requirements and constraints described in the introduction.

Freely-available docker images have been customized (e.g. to integrate HERD settings and automatic backup/restore to/from S3) for a set of services:

- Events calendar (based on NextCloud)
- Documents server (NextCloud)
- Software development platform (Gitlab)
- Web site (Grav)
- Meetings management (Indico)
- Chat (Matrix/Synapse)

All the services are integrated in the HERD infrastructure, featuring SSO via IAM-HERD, S3 storage on MinIO-HERD, and automatic backup and restore. Currently all the services are in production stage, except for Indico and Matrix which are still being evaluated.

7. Status and outlook

The HERD cloud computing infrastructure is currently terminating the first prototyping phase. The objectives of this stage were to identify the key technologies and deploy a first demonstration testbed to test the compliance with the full workflow, and have been fully met. Future work is foreseen to address the remaining technical issues; for example, pre-signed URLs by design expire after one week, so they are not suited for long-queuing/long-running jobs, being retrieved at job submission time, and a different authentication method/workflow for S3 job I/O must be identified. There are also plans to evaluate the possibility to distribute user's code also via CVMFS. On a broader scope, the federation of resources from different clouds/datacenters is a major topic regarding the future direction of this work. The choice of HTCondor as the batch managent system is a first step in this sense, given the native ways it offers to perform compute resource federation. Data storage and management is another prominent topic for federated environments: the choice of S3 as the storage backend brings compatibility with many HEP de-facto standard solutions that can be leveraged to achieve compatibility with the datalake storage paradigm, and the usage of Rucio [18] as the data manager system is currently foreseen.

The in-house implementaion of collaboration services on the cloud demonstrated the feasibility of such approach for small/medium collaborations, with an affordable and relatively low burden that makes this option a feasible one for collaborations experiencing constraints similar to HERD. Currently the services run as single instances; High Availability deployments will be considered in future depending on the reliability of the single-instance setup on INFN Cloud VMs, while service load is not foreseen to be an issue given the size of the collaboration. The portfolio will eventually be expanded should the necessity for more services come out in future.

To summarize, the cloud approach so far has proven itself capable of solving many issues related to the diverse computing model and workflow needs of the HERD collaboration. No showstopper has currently been identified, and the collaboration is steadily pursuing this approach for all the future developments of its computing infrastructure.

8. Acknowledgments

The authors thank the INFN Cloud staff for continuous support. The authors acknowledge the support of the DICE project under grant n. 101017207, and of the Italian Space Agency (ASI) under ASI-INFN Agreement No. 2021-43-HH.0.

References

[1] Gargano F et al. 2021 *Proc. of Sc.* (ICRC2021) 026
[2] DODAS Project: https://dodas-ts.github.io/dodas-doc/
[3] INFN Cloud: https://www.cloud.infn.it/
[4] Ceccanti A et al. 2016 *J.Phys.Conf.Ser.* **898** 102016
[5] eduGAIN interfederation: https://edugain.org/
[6] OnenID Connect: https://openid.net/connect/
[7] MinIO object storage: https://min.io/
[8] Amazon S3: https://aws.amazon.com/s3/
[9] Ceph distributed storage: https://ceph.io/en/
[10] DICE project: https://www.dice-eosc.eu/
[11] oidc-agent credentials helper: https://github.com/indigo-dc/oidc-agent
[12] Fajardo E M et al. 2015 *J. Phys. Conf. Ser.* **664** 062014
[13] Kubernetes software: https://kubernetes.io
[14] Docker platform: https://www.docker.com/
[15] Blomer J et al. 2011 *J. Phys. Conf. Ser.* **331** 042003
[16] Gitlab software: https://about.gitlab.com
[17] boto3-sts library: https://github.com/dodas-ts/boto3sts
[18] Rucio data management: https://rucio.cern.ch/