

# Architecture and prototype of a WLCG data lake for HL-LHC

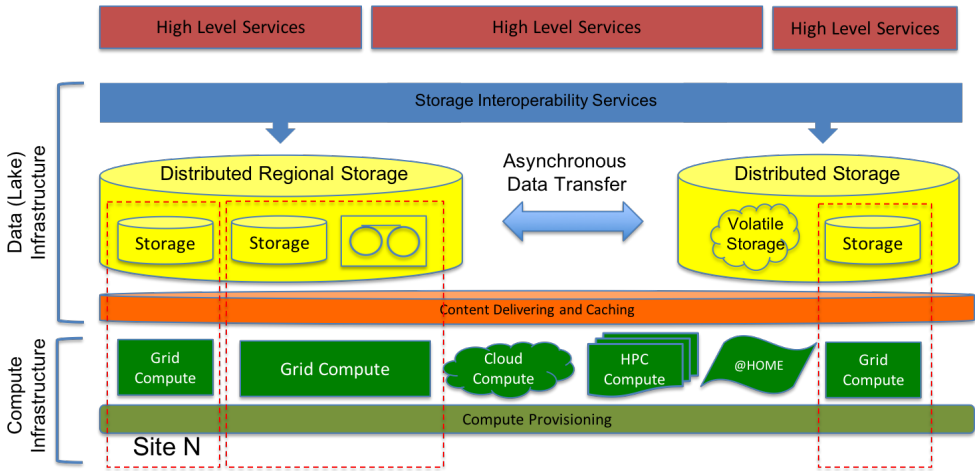
Ian Bird<sup>1</sup>, Simone Campana<sup>1</sup>, Maria Girone<sup>1</sup>, Xavier Espinal<sup>1</sup>, Gavin McCance<sup>1</sup> and Jaroslava Schovancová<sup>1</sup>

<sup>1</sup>CERN, 1 Esplanade Des Particules, Geneva, Switzerland

**Abstract.** The computing strategy document for HL-LHC identifies storage as one of the main WLCG challenges in one decade from now. In the naive assumption of applying today's computing model, the ATLAS and CMS experiments will need one order of magnitude more storage resources than what could be realistically provided by the funding agencies at the same cost of today. The evolution of the computing facilities and the way storage will be organized and consolidated will play a key role in how this possible shortage of resources will be addressed. In this contribution we will describe the architecture of a WLCG data lake, intended as a storage service geographically distributed across large data centers connected by fast network with low latency. Will present the experience with our first prototype, showing how the concept, implemented at different scales, can serve different needs, from regional and national consolidation of storage to an international data provisioning service. We will highlight how the system leverages its distributed nature, the economy of scale and different classes of storage to optimise the hardware and operational cost, through a set of policy driven decisions concerning data placement and data retention. We will discuss how the system leverages or interoperates with existing federated storage solutions. We will finally describe the possible data processing models in this environment and present our first benchmarks.

## 1 Introduction

The HL-LHC program at CERN will start in the second half of the 2020's and will present unprecedented computing challenges. The WLCG project and infrastructure, providing processing and storage resources for the LHC experiments, can realistically rely on a 20% resource growth per year. A naïve extrapolation of the experiments computing needs foresees a shortage of capacity by many factors and storage seems to be the main challenge. At the same time, HEP will share the same computing infrastructure with other sciences of comparable needs. The HEP Software Foundation community produced a white paper [1] indicating possible directions in several areas to address such challenges and in particular to reduce cost of computing. In the domain of data management cost can be reduced both by optimizing hardware use and operations. As an example, as of today the WLCG storage does not expose any notion of Quality Of Service (QoS) and the experiments end up replicating more data than they really need, given the retention policies and data access pattern they want to support. From the operational perspective, the WLCG storage is currently fragmented in more than one hundred endpoints of very different size and with very different level of support, while fewer and larger endpoints would offer the same, or probably a better service



**Fig. 1.** Diagram representing the possible future evolution of the WLCG data and compute infrastructure

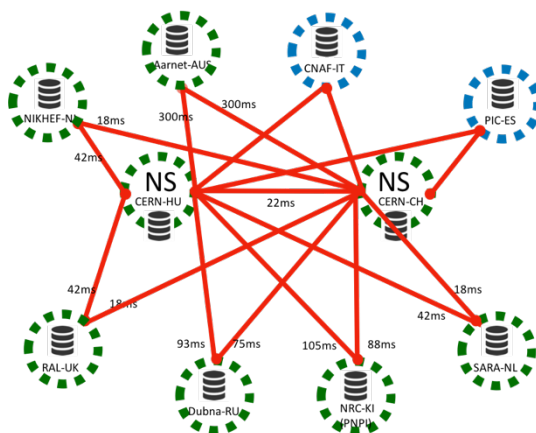
for a lower operational cost. The evolution of WLCG facilities moves in this direction: consolidation of storage resources, connected by reliable Terabit-scale networks and heterogeneous processing capacity not necessarily co-located with storage. The building blocks of such architecture are highlighted in Fig. 1: the storage endpoints of different QoS'es. The exchange of data is guaranteed through efficient Wide Area Network transfer protocols and orchestrated by storage interoperability and high level services. The data is served to the processing centres through a system of content delivering and caching, hiding the effects of latency introduced when data and compute are not co-located. We call such model a Data Lake.

In this paper we present our first studies in deploying a distributed storage infrastructure to prototype the concepts and ideas of a data lake. We will also present our first data access performance measurements in the data lake and compare the results with the current model. The studies are based on real workflows from the LHC experiments.

## 2 Deployment of a Data Lake prototype

To test and demonstrate some of the ideas described in the HSF Community White Paper, we deployed a storage instance with a central name space and data nodes geographically distributed across several WLCG T1s in Europe: the “*eulake*”. To maximize its geographical distribution we also deployed a pool node in Australia. We choose EOS [2] as storage technology because it demonstrated the capability to work as distributed storage for the WLCG T0, offers several of the functionalities we would like to leverage and test and we have in-house expertise from the development and operational perspective. We used the most recent version of EOS namespace, not yet deployed in production instances. The namespace of our prototype is deployed at CERN-Meyrin (read-only) and CERN Wigner (read and write). Fig. 2 illustrates the topology of our prototype. The deployment model of the data nodes varies from centre to centre: some sites deploy EOS pool nodes through containers, others install via RPMs. We tested the basic functionalities of the EOS-based prototype and in particular the possibility to implement QoS, data retention policies (multiple copies of the same file, single copy or erasure coding) and time-based automatic transitioning between one retention policy and the other. The internal metrics of the EOS prototype can be

monitored through the MONIT [3] based infrastructure that offers the same capabilities for the production instances.



**Fig. 2.** The diagram illustrates the topology of our prototype, with a central namespace (NS) deployed across two sites and distributed pool nodes. The green circles indicate the sites already participating to the deployment, while the blue circles are candidate sites. The distances between sites are measured in milliseconds of network round trip time.

Data access to the EOS prototype is enabled through the XrootD protocol, as every pool node can serve data through it. For 3<sup>rd</sup> party data transfers we enabled a gridFTP frontend on top of the instance and all the traffic is channelled through there, as in the production EOS instance at CERN. The prototype was integrated into the ATLAS data management system, Rucio [4], as an integration endpoint, therefore not impacting production. Rucio was used to trigger data replication inside the prototype and such data was used as input for the tests described later in this paper.

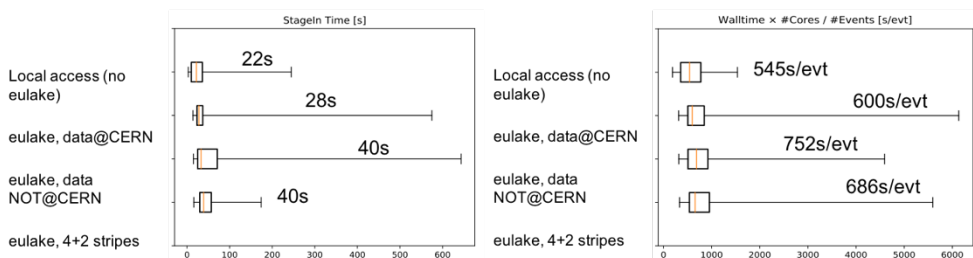
### 3 Data access performance measurements

We carried on a preliminary set of data access and performance measurements in the prototype. For this purpose, we use the HammerCloud [5] framework. The Worker Nodes accessing the data were provisioned through the CERN HTCondor batch system and were located in CERN-Meyrin. In the test, the data was always copied from storage to Worker Node and opened locally by the application. The data were stored and organized following four different test scenarios:

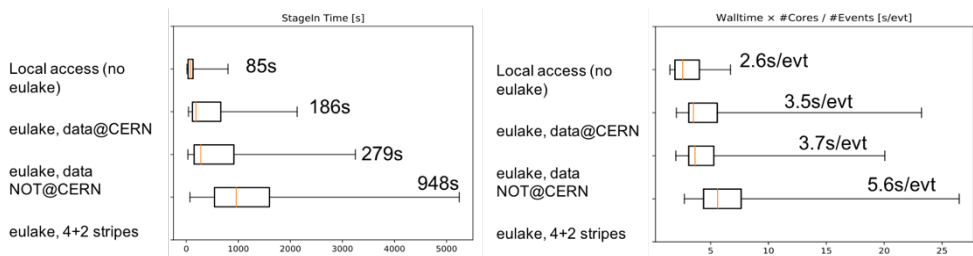
- 1) Data stored in the EOS production instance at CERN-Meyrin (not in the prototype). When accessed from CERN-Meyrin worker nodes, this represents the baseline measurement against which other type of data organization can be compared.

- 2) Data stored at CERN-Meyrin, in the EOS prototype. From the perspective of the data access pattern it is equivalent to 1), but relies on a storage instance of much lower performance.
- 3) Data stored at another site rather than CERN-Meyrin, in the EOS prototype. When accessed from CERN-Meyrin worker nodes this scenario tests the impact of network latency on data access.
- 4) Data stored in the EOS prototype, using erasure coding. Each file is chunked into 4 stripes, with the addition of two additional ones for redundancy. Each stripe is stored at a different site of the prototype and the files are re-assembled on the fly from multiple sources at data access time.

The test scenarios were run for two different workflows. The first workflow is CPU bound: downloads 40 MB of input and processes two events in approximately 5 minutes per event. The second workflow is I/O bound: downloads 6 GB of input, processes 1000 events in approximately 2 seconds per event. The results of the four test scenarios for the first and second workflow are summarized in Fig. 3 and Fig. 4 respectively.



**Fig. 3.** The results of the data access tests in the case of a CPU bound application. The panel on the left side shows the measurement of the stage-in (download) time in the different scenarios, while the panel on the right side shows the WallClock time per event measurement. For each measurement, the red line indicates the 50<sup>th</sup> percentile, the box corresponds to the 25<sup>th</sup> (left edge) and 75<sup>th</sup> (right edge) percentile and the left and right sides of the error bars show respectively the minimum and the maximum.



**Fig. 4.** The results of the data access tests in the case of a I/O bound application. The panel on the left side shows the measurement of the stage-in (download) time in the different scenarios, while the panel on the right side shows the WallClock time per event measurement.

For both workflows, the baseline local access results with the production instance (no eulake) show obviously the best results. In the CPU bound workflow, the results for the other scenarios are comparable with the baseline, while with much larger error bars, and therefore with more fluctuating results. In particular, the average WallClock time per event varies in average within 25% from the baseline and shows very coherent behaviour among the three scenarios where eulake is involved. The difference in stage-in time is more significant (up to a factor 2 from the baseline) but the impact is minimal as the workflow is CPU bound.

The I/O bound workflow results are considerably different. In the default scenario the processing time per event is approximately 2.6 seconds with very small fluctuations. In the case of local access to data in eulake (rather than the EOS production instance) this time increases to 3.5 seconds. This is due to the very limited hardware setup of eulake with respect of the CERN EOS production instance, both in terms of vertical scaling (performance of the head nodes) and horizontal scaling (number of disk servers in the data nodes). The performance in the scenario of remote access (data not at CERN) through eulake is surprisingly similar (3.7 seconds per event) while with large uncertainties. The impact of wide area network latency is therefore not significant in this test case, while of course in this limited setup we were not capable to test the impact at the level of bandwidth.

The result in the case of input data organized through erasure coding at different sites is considerably worst (5.6 seconds per event) with the largest fluctuations among all measurements. Because of the limited hardware available in the prototype and very different performance and reliability depending on the computer centre participating in eulake, this was foreseeable. A larger scale test with more homogeneous hardware and quality of service would be needed to draw conclusive evidence.

### 3 Conclusions

We set up a distributed storage instance based on the EOS technology. Such instance was small in terms of storage volume, with roughly 10 TB of storage provided per data centre, in one or two disk servers. The instance however was widely geographically distributed, with network distances varying between a few milliseconds and up to 300 milliseconds. The deployment of data nodes followed several approaches depending on the data centre: from RPMs to Docker containers to volume export in the case of a CEPH underlying storage system. We integrated such distributed storage instances with the ATLAS distributed computing services (Rucio and the workload management system PanDA [6]). The next step would be integrating with the CMS framework and test more specific workflows.

In summary, we have a prototype in place and understand what we want to measure. The impact of latency in distributed storage seems to be quantifiable at the few tenths of percent for I/O bound workflows and seems not to impact CPU bound workflows. The limited test statistic and the limited hardware setup in the prototype do not allow however to draw firm conclusions, as the uncertainties in the distributions of the measurements are large. The prototype we set up serves to test many ideas in preparation for HL-LHC.

### References

1. A. Alves *et al.*, A Roadmap for HEP Software and Computing R&D for the 2020s, arXiv:1712.06982 [physics.comp-ph] (2017)

2. A. Peters, L. Janyst, Exabyte Scale Storage at CERN, Journal of Physics: Conference Series. Vol. 331. No. 5. IOP Publishing (2011)
3. A. Aimar *et al.*, Unified Monitoring Architecture for IT and Grid Services, Journal of Physics: Conference Series. <http://stacks.iop.org/1742-6596/898/i=9/a=092033> (2017)
4. M. Barisits, T. Beermann, V. Garonne, T. Javurek, M. Lassnig, C. Serfon, The ATLAS Data Management System Rucio: Supporting LHC Run-2 and beyond, ACAT, Seattle (2017)
5. J. Schovancová, A. Di Girolamo, A. Fkiaras, V. Mancinelli, Evolution of HammerCloud to commission CERN Compute resources, The 23rd International Conference on Computing in High Energy and Nuclear Physics, Sofia (2018)
6. F H Barreiro Megino *et al.*, Panda for atlas distributed computing in the next decade. Journal of Physics: Conference Series, 898(5):052002 (2017)