Federated data storage evolution in HENP: data lakes and beyond

Andrey Zarochentsev[1], Xavier Espinal[2], Andrey Kiryanov[3], Jaroslava Schovancová[2]

[1]St. Petersburg State University. 13B Universitetskaya emb., Saint Petersburg, Russia
[2]CERN. Geneva 23, Switzerland
[3]Petersburg Nuclear Physics Institute of NRC "KI". 1 Orlova roshcha, Gatchina, Russia

Abstract. Storage has been identified as the main challenge for the future distributed computing infrastructures: Particle Physics (HL-LHC, DUNE, Belle-II), Astrophysics and Cosmology (SKA, LSST). In particular, the High Luminosity LHC (HL-LHC) will begin operations in the year of 2026 with expected data volumes to increase by at least an order of magnitude as compared with the present systems. Extrapolating from existing trends in disk and tape pricing, and assuming flat infrastructure budgets, the implications for data handling for end-user analysis are significant. HENP experiments need to manage data across a variety of mediums based on the types of data and its uses: from tapes (cold storage) to disks and solid state drives (hot storage) to caches (including world wide access data in clouds and "data lakes"). The DataLake R&D project aims at exploring an evolution of distributed storage while bearing in mind very high demands of the HL-LHC era. Its primary objective is to optimize hardware usage and operational costs of a storage system deployed across distributed centers connected by fat networks and operated as a single service. Such storage would host a large fraction of the data and optimize the cost, eliminating inefficiencies due to fragmentation. In this talk we will highlight current status of the project, its achievements, interconnection with other research activities in this field like WLCG-DOMA and ATLAS-Google DataOcean, and future plans.

1. Introduction

The High Luminosity LHC (HL-LHC) will be a multi-Exabyte challenge where the envisaged Storage and Compute needs are a factor 10/100 above the expected technology evolution and flat funding [1] (fig.1).

WLCG community needs to evolve current computing and data organization, management and access models in order to introduce changes in the way the computing infrastructure is currently used, mainly focused on optimizations to improve efficiency and performance, not forgetting simplification of operations. These are the ingredients that will allow to drive down costs and be able to satisfy the HL-LHC requirements.

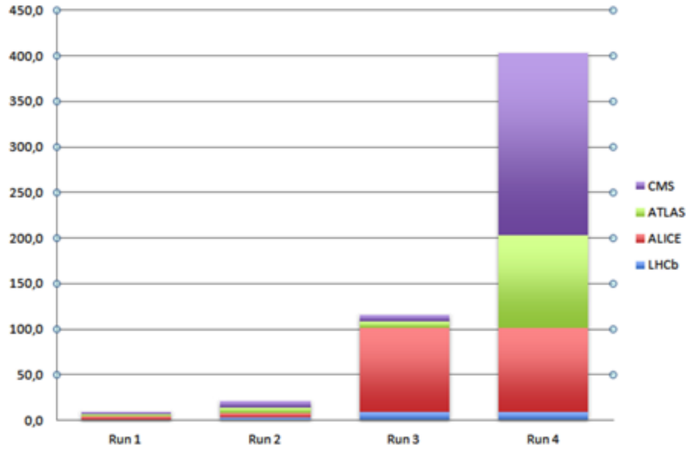

Fig. 1. Rough estimate of a raw data volume of major LHC experiments (in PB per year) [1].

Technologies that will address the HL-LHC computing challenges may be applicable for other communities, such as SKA, DUNE, CTA, LSST, BELLE-II, JUNO, etc. to manage large-scale data volumes. One of such technologies that we will discuss in this paper is Data Lake A Data Lake in WLCG context is described as a set of sites associated by network or geographical proximity providing a common storage layer. The Data Lake holds the big part of the experiments data and provide the data access layer to compute hence reducing the overall storage needs. Proximity could be defined by geography, connectivity, funding or a shared user community. This requires that their combined storage capacity and network bandwidth can meet the demands of the designated task and that usage of the different sites is transparent to the users, which, in turn, implies some form of trust relationship between the sites and a way to locate data, ranging from a simple file catalogue to a full-fledged namespace.

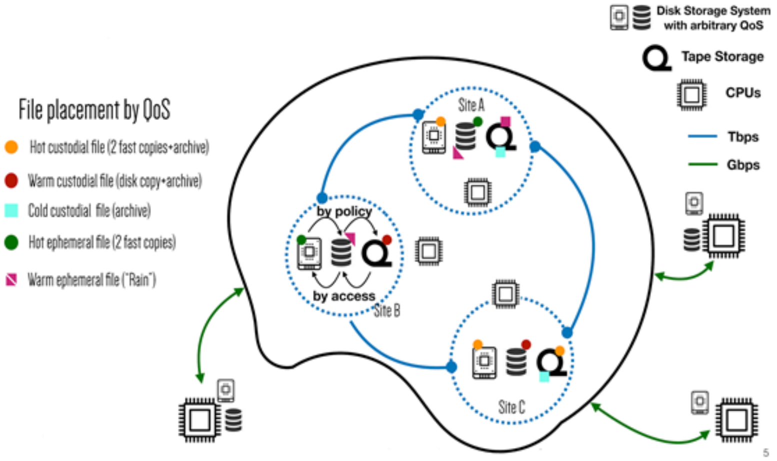
Disk Storage System with arbitrary QoS
Tape Storage
CPUs
Tbps
Gbps
File placement by QoS
Hot custodial file (2 fast copies+archive)
Warm custodial file (disk copy+archive)
Cold custodial file (archive)
Hot ephemeral file (2 fast copies)
Warm ephemeral file ("Rain")
Site A
by policy
by access
Site B
Site C
5

Fig. 2. A Data Lake comprising three sites with different compute capabilities [2].

While access for users is transparent, the population and management of the storages within the Data Lake is a planned and managed activity. This includes the transitions between QoS levels (fig. 2). These operations are done on the granularity of the Data Lake. Data is moved to or from the Data Lake as a whole, not to or from a specific site, for instance internal data movement and files layout are managed and handled by the Data Lake internally. Resource management within the Data Lake is the responsibility of the Data Lake.

Based on what has been described we think some of the fundamental requirements for a WLCG Data Lake infrastructure are:

- Common namespace and interoperability
- Coexistence of different QoS
- Geo-awareness
- File transitioning based on namespace rules
- File layout flexibility
- Distributed redundancy
- Fast access to data, latency compensation via caching
- Built-in fault tolerance

It's worth mentioning that Data Lake is one of the several storage-related R&D projects conducted in parallel. Other R&D projects aimed to address proper handling of storages with different QoS include:

- Data Carousel (ATLAS)
- Data Ocean (ATLAS + Google)
- Data Streaming

All of them are in progress as a part of WLCG DOMA [3] or/and IRIS-HEP [4] global R&Ds for the HL-LHC. We strongly believe that it is important to develop a coherent solution to address the HL-LHC data challenges and to coordinate above and future projects.

2. EULake prototype

In order to evaluate existing storage technologies and their applicability in the Data Lake model, as well as prototype and test ideas, a Data Lake prototype spanning several WLCG sites with the namespace management nodes located at CERN has been built. Several years ago, in 2015, a similar prototype based on EOS [5] and dCache [6] storage systems was built on Russian sites during the Russian Federated Data Storage project [7]. Existing expertise in building and testing federated multi-site storages allowed us to fruitfully join the EULake with some decent resources (see section 3) and conduct important functional and performance tests.

As of 2019, the Data Lake prototype (named EULake) spans seven European/Russian sites: CERN, JINR, NIKHEF, PIC, PNPI (part of NRC "KI"), RAL, SARA and three Australian sites (Melbourne, Perth and Brisbane). Some of them only provide storage resources, others, including CERN and PNPI, also provide accompanying dedicated compute endpoints that allow to conduct real-life HammerCloud [8] tests on EULake infrastructure. All sites have also deployed perfSONAR [9] servers to automate network monitoring.

Initially, EOS storage system developed at CERN was the only software component used to build a working EULake prototype. One of the reasons was a rich feature set of EOS, which maps nicely into the basic requirements defined above:

- Built-in namespace
- Storage groups and catalog attributes
- Geotags and Geo-scheduling
- Layout types (replica, RAIN)
- Support of xrootd [10] protocol and related proxy tools (xCache)
- Support for slave metadata managers (MGMs)
- EOS already has the machinery to support proposed QoS types (see table 1)

We are able to measure performance of EULake with HammerCloud[8], leveraging standard full-chain workflows and data access patterns. Initial focus was on ATLAS workflow with four data access (read) scenarios:

Base. Local access (no EULake)
A. EULake, data@CERN, compute@CERN
B. EULake, data NOT@CERN, compute@CERN
C. EULake, 4+2 stripes, compute@CERN

Data is copied from the storage to the worker node at the job start.

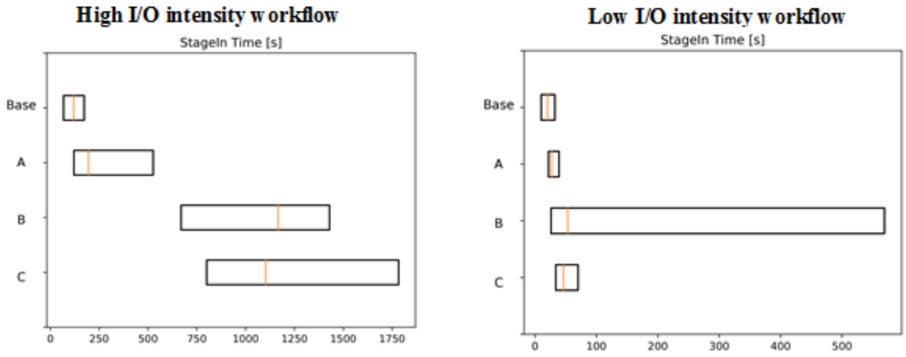

Fig. 3. Stage-in times in seconds for High-I/O and Low-I/O intensity workflows in four aforementioned scenarios: Base, A, B and C.
Lower is better. Black box shows 25%-75% percentile range, orange line marks the median.

Table 1. Mapping between DataLake QoS and EOS usage scenarios.

DataLake	EOS
Hot custodial file (2 fast copes+archive) Warn custodial file (disk copy+archive) Cold custodial file (archive) Hot ephemeral file (2 fast copy) Warm ephemeral file (Rain)	group.X replica 2 + CTA group.Y replica 3 + CTA group.Z plain + CTA group.W replica 2 group.U RAIN

For different test scenarios different storage endpoints were defined in HammerCloud with prescribed paths to directories with attributes that corresponded to expected QoS characteristics – binding to specific pools, number of replicas, type of data redundancy (replication or RAIN). You can see sum results of tests in fig. 3.

3. Russian resources and work in EUlake

In order to take part in EULake a participating site has to provide some resources. Currently in Russia two major scientific centers participate in Data Lake R&D: NRC “Kurchatov Institute” and JINR, both using a virtualized environment.

Unlike at CERN, at PNPI EOS is not installed on bare hardware, but deployed on top of Ceph [11] storage. The reason for this is added flexibility. Ceph allows for easy re-allocation of storage space between consumers with configurable redundancy for different types of data.

During the initial allocation of EULake resources at PNPI an interoperation between EOS and Ceph had to be verified for any possible incompatibilities and performance bottlenecks. We have conducted EOS performance tests in three possible configurations: Ceph block device with replication, Ceph block device with Erasure Coding and Ceph filesystem (fig. 4). In our tests we were using the latest available at the moment version of Ceph Mimic 13.2.1.

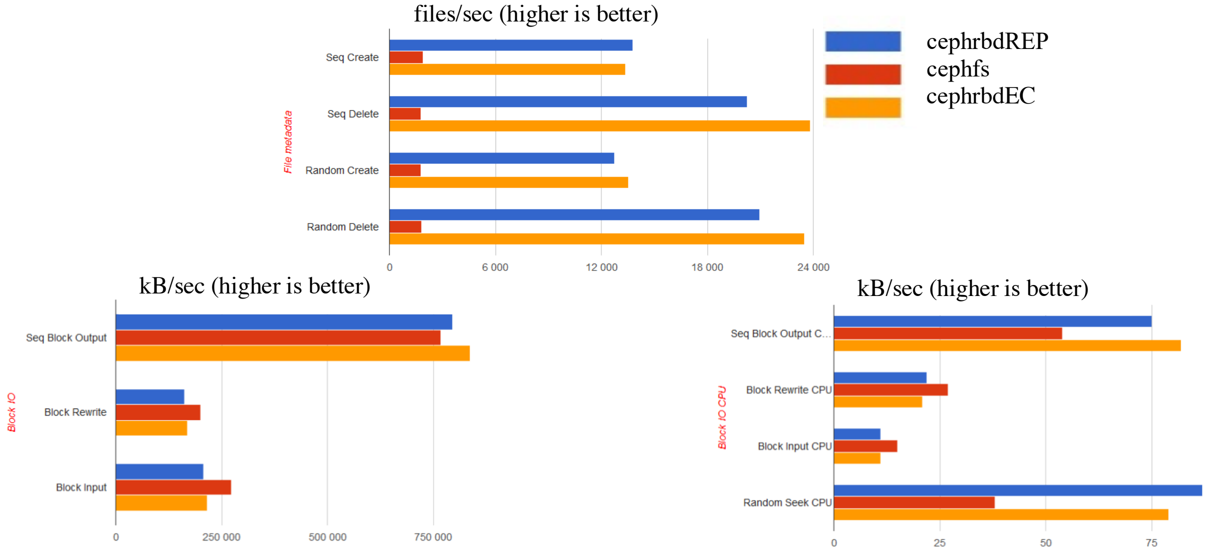

Fig. 4. EOS on top of Ceph performance measurements.

As it can be seen from the results, block I/O performance of Ceph replicated (cephrbdREP) and Erasure Coded (cephrbdEC) block devices as well as Ceph filesystem (cephfs) with EOS was on par, including the CPU utilization, while performance of metadata operations was significantly slower with Ceph filesystem. This was expected as Ceph filesystem maintains coherent metadata across all clients which adds latency overhead. As a conclusion we have decided not to deploy EOS on top of Ceph filesystem, but keep the deployment on top of Replicated and Erasure Coded block devices, which can be seen as different QoS types in Data Lake terms.

After local testing of PNPI disk pools, the authors proceeded to test the Russian part of EULake. For this purpose PNPI FSTs with Ceph backend were connected to EULake and marked with geo-tags according to the hierarchical scheme: RU::PNPI. Also, before the start of testing, geo-tags of JINR FSTs were changed according to the same scheme: RU::JINR.

This hierarchical scheme allows a sequential binding to the client location starting from the rightmost side of the geo-tag. If you search for the closest transfer endpoint, in the beginning the match of the complete geo-tag is evaluated, then parts of the geo-tag separated by double colons are dropped from the right side until a match is found: RU::PNPI::DISK1 RU::PNPI RU.

After geo-tagging, storage endpoints for different parts of EULake namespace were created in HammerCloud, corresponding to different layouts (Plain, Replica (2 stripes), RAIN (4 + 2 stripes)) and different placement policies (Gathered, Hybrid, Simple (based on client geotag)). Tests were started with HammerCloud utilizing two types of ATLAS workflows typically found in the real life:

- Simul - low I/O intensity workflow running CPU-intensive ATLAS simulation with Geant 4.
- ProductionDerivation - high I/O intensity workflow running ATLAS reconstruction jobs.

After the first successful tests from CERN (fig. 5) it was decided to enhance the tests by using client geolocation. For this purpose, a Compute Element (CE) with characteristics that satisfy ATLAS data processing requirements was deployed at the PNPI resource center and registered with the HammerCloud. Unfortunately, HammerCloud configuration for the new CE at PNPI took longer than expected and the statistics for the tests were gathering slowly.

In order to get some basic results faster it was decided to simultaneously run local read/write tests for the EULake with exactly the same scenarios that were configured for the HammerCloud, which represent combinations of 3 layouts and 3 placement policies:

- Layouts: Plain, Replica (2 stripes), RAIN (4+2 stripes)
- Placement policies: RU:Gathered (on all Russian sites), RU:Hybrid (on Russian sites with a copy on any pool in EULake), Simple (on the closest data pool).

The xrdstress utility, a part of the EOS test package, and a simple copy by the xrdcp command were used as a testing tool. The xrdstress utility gives averaged results - the mean value and the dispersion of the read/write speed. But in the case of our tests intermediate peaks in transfer performance were also interesting, so we had to use both tools side by side to measure transfer performance for EULake endpoint mentioned above. A similar testing methodology has already been used by the authors when working for the Russian Federated Storage project [7,13,14,15], but for the EULake case test scripts were somewhat refined for existing realities and updated software.

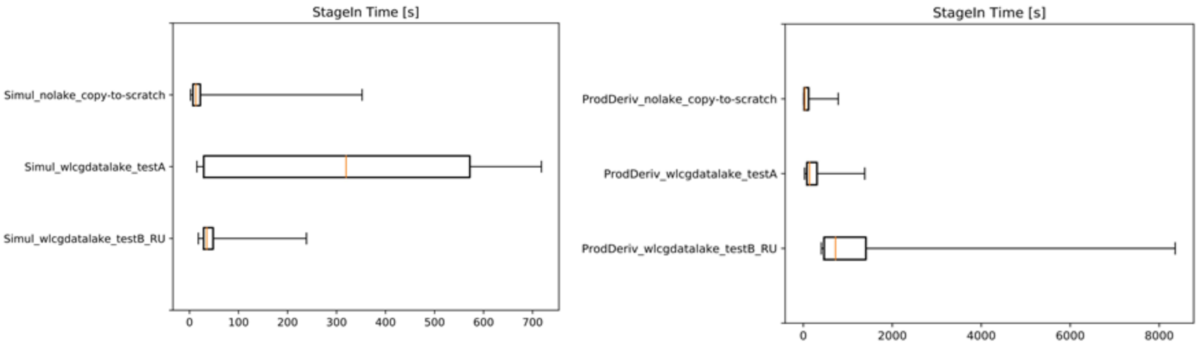

Fig. 5. Stage-in times in seconds for Simul (Low-I/O) and ProductionDerivation (High-I/O) workflows in three scenarios: no EULake, EULake with data@CERN and compute@CERN (test A), EULake with data@RU and compute@CERN (test B). Lower is better. Black box shows 25%-75% percentile range, black line shows 1%-99% percentile range, orange line marks the median.

The tests were conducted from a client computer – virtual machine with 2 cores, 4GB RAM, 10GB Ethernet located inside the local network of the PNPI resource center with RU::PNPI geo-tag. Reading was always done from the nearest data pool which had the necessary replica, and in the case of a simple placement policy writing was also directed to the nearest data pool.

Expected results: Availability of geo-local replicas should improve file read (stage-in) speed; An ability to tie directories to local storages (FSTs) should improve write speed for files in such directories (stage-out).

The results of (fig. 6, 7) almost coincided with the expectations, with the exception of a few subtleties that were well understood. Placement policy defines where to write a file, but does not guarantee that a file will say at this location, because a file can be moved by EOS balancers in the background. This leads to situations when a file was written to the closest data pool, but at the time of reading it's no longer available there.

The read/write speed for the RAIN layout drops significantly compared to the replica layout. This was not so noticeable during the tests at CERN (fig. 3) where all parts of RAIN were located at pools inside CERN network. The reason for this is also clear - the xrootd client does not have a native RAIN support, therefore at first all four stripes of the file are gathered on a single pool defined by the geotag, and only after that the file is transferred to the client. The same happens during the write - the file is copied to a single pool from where its stripes are transferred to other pools by EOS and only after that the client is sent a message about successful transfer completion. In our tests network connection between RAIN pools was often much worse that it was between the client and the closest pool.

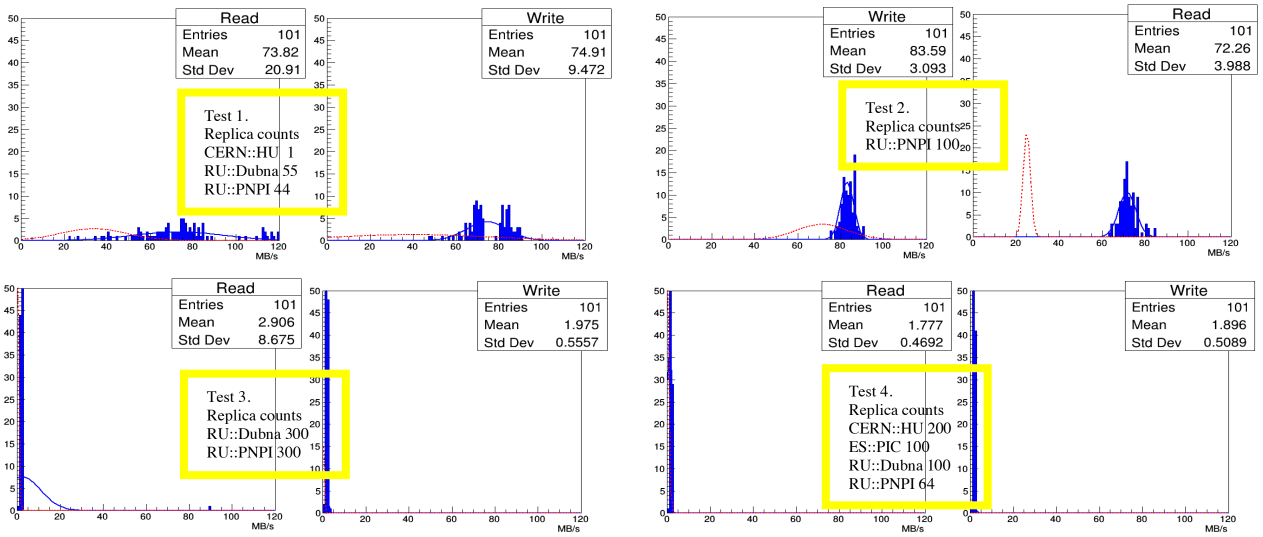

Fig. 6. Test results with a single replica (tests 1,3) for plain and RAIN (tests 3,4) layouts while writing to the closest data pool (tests 2,4) and all Russian sites(tests 1,3). Numbers in the yellow frames show the replica distribution at the end of the tests. The dashed red line shows xrdstress result, the blue histogram shows the xrdcp results. Both tests have the similar parameters: 100 repetitions, 100 MB file size.

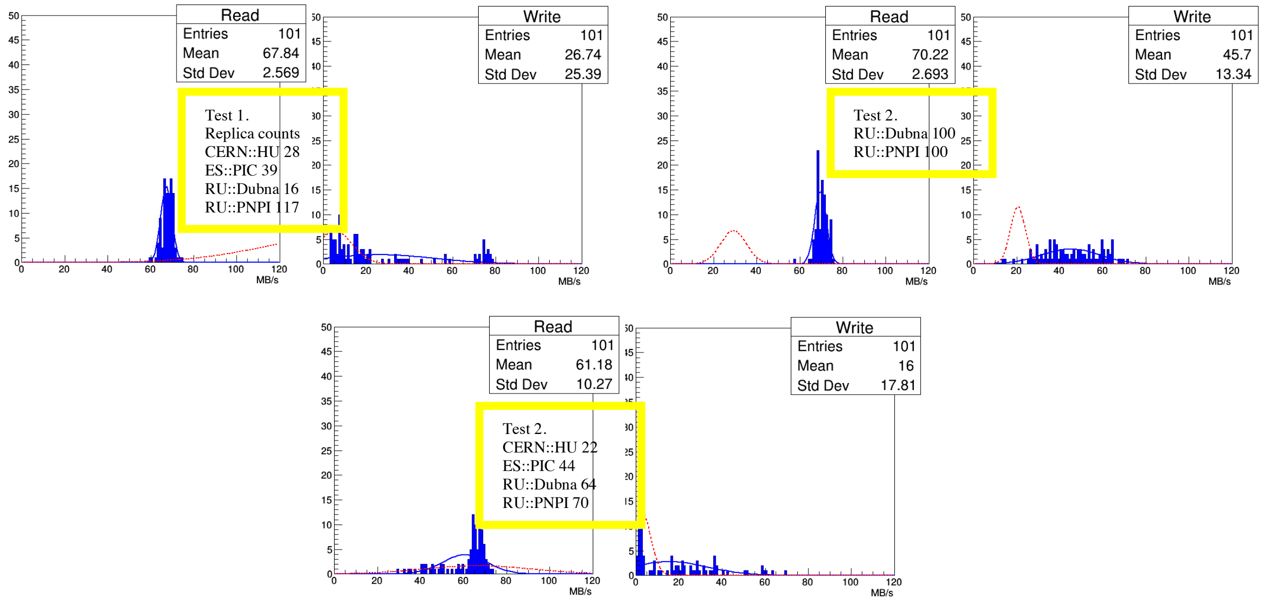

Fig. 7. Test results with creation of two replicas with different placement policies: on the closest data pool (test 1), on all Russian sites (test 2) and on a Russian site with a copy on a non-Russian site (test 3). Numbers in the yellow frames show the replica distribution at the end of the tests. The dashed red line shows xrdstress result, the blue histogram shows the xrdcp results. Both tests have the similar parameters: 100 repetitions, 100 MB file size.

4. Conclusions

In this paper we have shown how Ceph can be used as one of the underlying storage technologies for the DataLake. We have exploited and compared various options for provisioning of the Ceph storage.

As the primary results of this work we can show how the Data Lake prototype can be built based purely on EOS. On the other hand we discovered some inconsistencies with the proposed schemes.

First and foremost are placement policies. In the DataLake, when storing data in accordance with a certain QoS, it is assumed that the data will stay at the corresponding QoS pools. EOS placement policies define the pool where the data is first transferred, but they do not guarantee that the data will stay there. The data can migrate to a different pool possibly corresponding to a different QoS depending on the balancing settings. This problem can be dealt with by turning off the global balancing, but leaving the group balancing on and defining groups as separate QoS. But this solution is not universal and in principle artificial - it forces us to abandon some functionality of the system in favor of the requirements. And it still does not protect the data from migrating between the regions. This will be addressed at software level and need more testing at scale as both the concepts and the code are under development.

The second is somewhat expected inefficiency of the EOS RAIN storage layout. Without client-side support for multi-striped I/O, this layout is only efficient when all parts of RAIN are in the same high-throughput network. As we can see, RAIN loses performance significantly if it spans storage servers outside of CERN network. This problem can be mitigated by deploying a caching server on the client's network. There are ongoing R&D activities in WLCG to understand caching mechanisms and the ability to efficiently hide latency in different scenarios.

Of course, EOS is not the only software that can be used for such infrastructure. During the Russian Federated Data Storage project it was shown that dCache (version 2 at the time) can also be used in such a distributed installation. Moreover, dCache has significantly improved feature-wise in the last years with the release of version 3.

In order to allow sites and communities to have a freedom of choice of the storage system, and evaluate a slightly more heterogeneous Data Lake, EULake is currently transitioning from a EOS-only system into EOS + Rucio [15].

Acknowledgments

The authors express appreciation to the computing centers of NRC "Kurchatov Institute", JINR, SPbSU and other institutes for provided resources. And also M5 CONF2019_1 in SPbSU.

References

[1] D. Adamova, M. Litmaath, New strategies of the LHC experiments to meet the computing requirements of the HL-LHC era, in proceedings of 55th International Winter Meeting on Nuclear Physics, PoS (BORMIO2017) 053, 2017.

[2] X. Espinal, Data Lake R&D: high level goals, Joint WLCG and HSF workshop, 2018.

[3] https://twiki.cern.ch/twiki/bin/view/LCG/DomaActivities

[4] http://iris-hep.org/

[5] A. J. Peters, L. Janyst, Exabyte Scale Storage at CERN, 2011 J. Phys.: Conf. Ser.331 052015

[6] https://www.dcache.org/

[7] A.Kiryanov, A. Klimentov, D. Krasnopevtsev, E. Ryabinkin, A. Zarochentsev, Federated data storage system prototype for LHC experiments and data intensive science, J. Phys.: Conf. Ser. 898 062016, 2016.

[8] J. Schovancová, A. Di Girolamo, A. Fkiaras, V. Mancinelli, Evolution of HammerCloud to commission CERN Compute resources, to appear in proceedings of the 23rd International Conference on Computing in High Energy and Nuclear Physics, Sofia, 2018

[9] https://www.perfsonar.net/

[10] http://xrootd.org/

[11] https://ceph.com/

[12] A.Kiryanov, A.Klimentov, A. Zarochentsev, A, NRC "KI" participation in the datalake project. Volume 2267, 2018, Pages 457-461. (CEUR Workshop Proceedings), 2018

[13] A. Kiryanov, A Klimentov, A. Zarochentsev. Russian scientific data lake. Open Systems Journal, issue 4, 2018

[14] A. Kiryanov, A Klimentov, A. Zarochentsev, M. Grigorieva, BigData and computing challenges in high energy and nuclear physics, Journal of Instrumentation, 29 Jun 2017

[15] 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