Evaluation of the Impact of Various Local Data Caching Configurations on Tier2/Tier3 WLCG Sites

Aleksandr Alekseev[6,7,8], Stephane Jezequel[9], Andrey Kiryanov[1], Alexei Klimentov[3], Tatiana Korchuganova[6,7,8], Valery Mitsyn[4], Danila Oleynik[4], Serge Smirnov[5], Andrey Zarochentsev[2]

[1] NRC "Kurchatov Institute" – PNPI, Gatchina, Russia
[2] Saint Petersburg State University, Saint Petersburg, Russia
[3] Brookhaven National Laboratory, Upton, NY, USA
[4] Joint Institute for Nuclear Research, Dubna, Russia
[5] National Research Nuclear University MEPhI, Moscow, Russia
[6] Institute for System Programming RAS, Moscow, Russia
[7] University Andres Bello, Santiago, Chili
[8] Plekhanov University of Economy, Moscow, Russia
[9] Laboratoire d'Annecy de Physique des Particules, Annecy, France
kiryanov@cern.ch

Abstract. In this paper, we describe various data caching scenarios test implementation and lessons learned. In particular, we show how local data caches may be configured, deployed, and tested. In our studies, we are using xCache, which is a special type of Xrootd server setup to cache input data for a physics analysis. A relatively large Tier2 storage is used as a primary data source and several geographically distributed smaller WLCG sites configured specifically for this test. All sites are connected to the LHCONE network. The testbed configuration is evaluated using both synthetic tests and real ATLAS computational jobs submitted via the HammerCloud toolkit. The impact and realistic applicability of different local cache configurations is explained, including both the network infrastructure and the configuration of computing nodes.

Keywords: Federated Storage, xCache, WLCG, DOMA.

1 Introduction

HENP experiments are preparing for the HL-LHC era, which will bring an unprecedented volume of scientific data. This data will need to be stored and processed by collaborations, but the expected resources growth is nowhere near extrapolated requirements of existing models both in storage volume and compute power. It is well understood that computing models need to evolve. Such evolution includes multiple aspects:

- Optimized data processing, squeezing the maximum from the available CPU/GPGPU/FPGA resources.

- Optimized data storage, reduction of the number of copies, different data access methods, full utilization of network resources.
- Cost optimizations, no high-end expensive RAID setups, no underutilized CPUs on storage servers, no HDDs with 90% free space on the worker nodes.
- Deployment optimizations, scalability and containerization with on-demand expansion into the cloud (both community and commercial).
- Operational cost optimization, more standardized solutions, lower requirements on unique Grid expertise.

2 Ongoing R&D Projects

WLCG and experiments have launched multiple R&D projects to address HL-LHC challenges:

- Data Lake. The aim is to consolidate geographically distributed data storage systems connected by fast network with low latency. The Data Lake model as an evolution of the current infrastructure bringing reduction of the storage and operational costs.
- Intelligent Data Delivery Service (iDDS). The intelligent data delivery system will deliver events as opposed to delivering bytes. This allows an edge service to prepare data for production consumption, the on-disk data format to evolve independently of applications, and decrease the latency between the application and the storage. The first implementation in April-May 2020 for Data carousel and active ML workflows.
- Hot/Cold storage. Data placement and data migration between “Hot” and “Cold” storages using data popularity information.
- Data format and I/O. Evaluating new formats (e.g. parquet) and I/O performance for HENP data.
- Third Party Copy. Improve bulk data transfers between sites and find a viable replacement to the GridFTP protocol.
- Operations Intelligence. Reduce the HEP experiments computing operations effort by exploiting anomaly detection, time series and classification techniques to help the operators in their daily routines, and to improve the overall system efficiency and resource utilization.
- Data Carousel. Use tape more effectively and actively in distributed computing context.

3 Objectives of this work

This research is conducted in collaboration with the European Data Lake Project, which is part of the WLCG DOMA initiative [1]. We will show a few possible ways of optimizing remote data access from the worker nodes in somewhat small T2/T3 setups or dynamically scaled containerized deployments for physics analysis payloads. This kind of deployment implies the necessity of heavy site-remote read-biased

data I/O, and time slot (t) allocated for analysis job is normally split into three phases (disregard some overhead): input read (t_1), compute (t_2) and output write (t_3). Sometimes analysis payloads can read and write data while performing computation which makes it hard to separate t_1 from t_2 and t_2 from t_3, but in any case, at least some data needs to be preloaded before computation can start. Here, we will focus on optimizing t_1 and thus improving the CPU utilization of a compute resource.

In order to optimize the read time (t_1) in cases where hardware and network performance cannot be easily improved, various caching systems are standardly used. However, any kind of caching is only effective with a sufficient cache hit ratio. The very first thing we need to check is the real repeatability of read requests during standard physics workflow. Let us try to evaluate the typical number of read requests to a single file (K) of the ATLAS experiment data suitable for user analysis. Figure 1 shows the ATLAS derivation data sample popularity (number of usage) by users' analysis tasks [2].

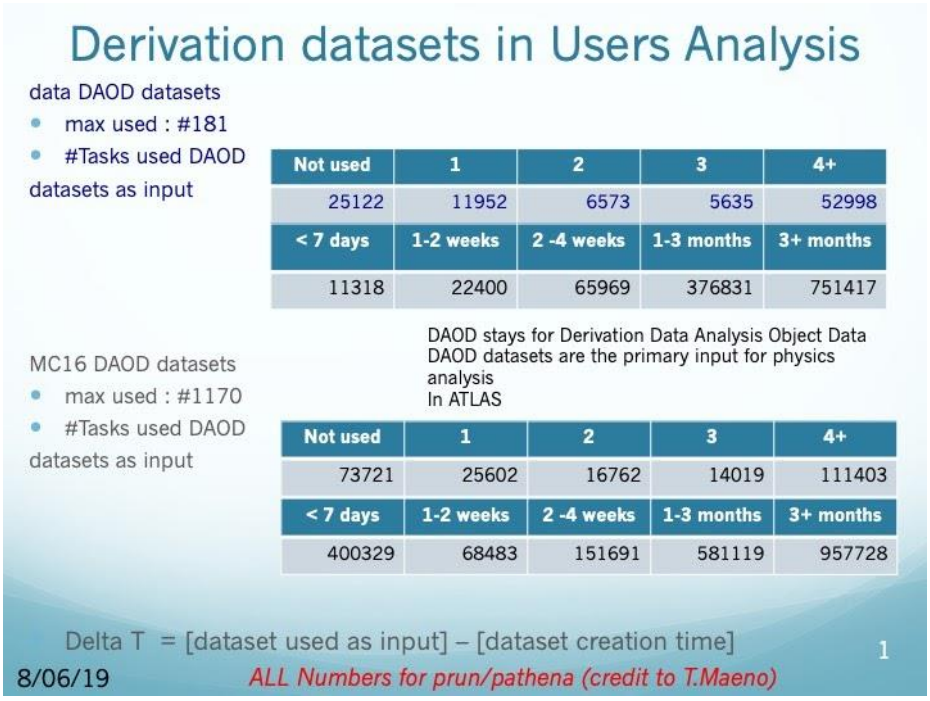

Fig. 1. Derivation datasets usage in ATLAS physics analysis.

There is at least one dataset that was accessed 1170 times. On average, ATLAS DAOD datasets consist of 50 files, which means that each file in this dataset was accessed at least 20 times if the data popularity is evenly split between them. We take K = 20 as the basis of our tests. In the end, we have to build a distributed data processing system where the computing element (CE) is distinct and distant from the primary storage system; computing tasks are submitted by users, and these tasks can

eventually request access to the same input file up to 20 times. The input file is located on a primary storage and its size varies from 1 to 5 GB (the overview of ATLAS data files is given in the HEPiX talk [3]). In this case, the infrastructure and building blocks of CEs can vary significantly between the sites.

4 The test bed

For the scheme we have explained above, it is necessary to describe some specific details such as the data access protocol and a caching system. In our tests, we will use an xrootd protocol which is widely employed by LHC experiments and has an important property of supporting redirects. The latter feature is important when building distributed storage systems, including distributed caching systems. As a caching software we will use an xCache which is, basically, a standard xrootd server configured in a special way.

We have decided to exploit three data caching schemes shown in Fig. 2. There is no universal solution due to the hardware (especially network) differences on different sites. With these schemes, we tested three quite obvious scenarios:

1. A single dedicated cache server for sites having a modest external connectivity (~1 Gbps) and a relatively good internal network for worker nodes (>=10 Gbps).
2. A local isolated cache on each worker node for sites having a good external connectivity (>=10 Gbps), but modest internal network for worker nodes (~1 Gbps).
3. A shared cache between worker nodes for sites having external and internal networks of the same relatively high quality (>=10 Gbps) – this approach requires some sort of service discovery.

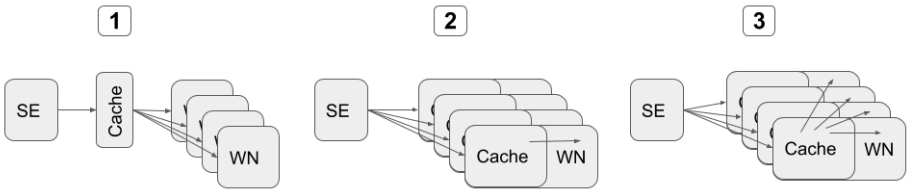

Fig. 2. Three data caching schemes.

At the first stage of the testing, scenarios 1 and 2 were implemented using resources of JINR, PNPI and MEPhI (Fig. 3). JINR was used as a primary storage with 10 Gbps uplink while still having a local CE with 1 Gbps internal network. This CE was used as a reference and no caching system was deployed there. Tests with JINR CE were only carried out at the very beginning; later, such tests lost their value. The PNPI CE located 520 km (~ 11 ms latency) from JINR has 10 Gbps internal network and 10 Gbps uplink to primary storage. The MEPhI CE located 120 km (~1 ms latency) from JINR has 1 Gbps internal network and 10 Gbps uplink to primary storage.

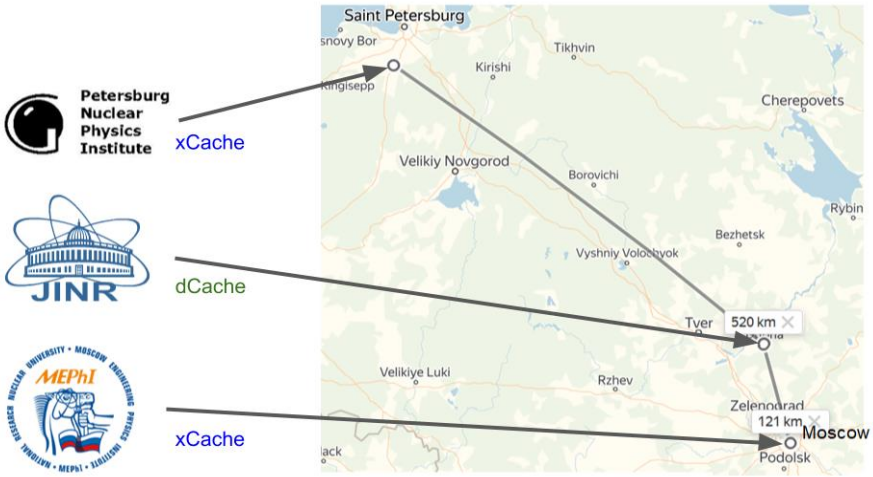

Fig. 3. Testbed on the map.

5 Tests and results

In order to receive some useful performance metrics, we needed tthe appropriate tests. In this case, the authors already had some experience in testing distributed storage systems with both synthetic tests [4] and the HammerCloud toolkit used by the ATLAS experiment [5], both of which were used for testing the EOS-based distributed storage [6].

- As synthetic tests, a simple file copy by the xrdcp tool was used.
- As a HammerCloud payload a real-life Athena analysis task was submitted to the CEs.

The first tests, which were reported at HEPiX Workshop [3], were conducted at PNPI and JINR sites, using only the dedicated xCache. Figures 4, 5, 6 show the results of HammerCloud tests with and without xCache (JINR is a primary storage, so xCache was not used there). The results, as can be seen from the graphs:

- Wallclock at PNPI (t):
  - Direct mean time = 2698 ± 577 s
  - xCache mean time = 1934 ± 139 s
  - Difference ~ 770s, ~30%
- Download input files time at PNPI (t_1):
  - Direct mean time = 811 ± 574 s
  - xCache mean time = 53 ± 137 s
  - Difference ~ 770s, ~95%
- Download input files time at JINR (t_1):
  - Direct mean time = 117 ± 17 s

These results give an idea of the fundamental benefits of using xCache.

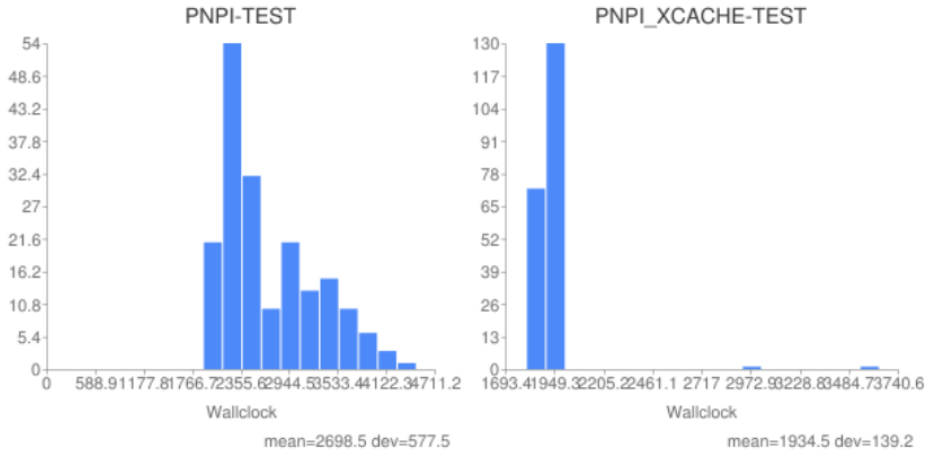

Fig. 4. Wallclock time at PNPI using HammerCloud test N20146370 from Template 1099 (copy2scratch). Direct read on the left, dedicated xCache on the right.

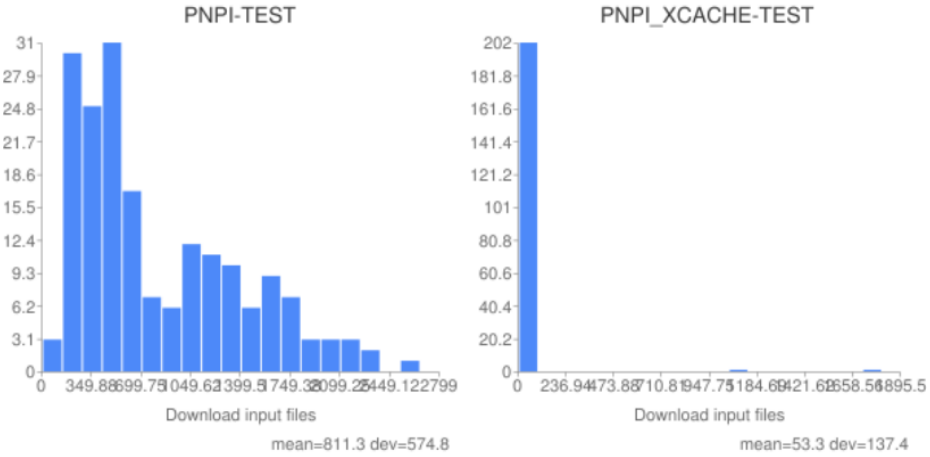

Fig. 5. Download input files time at PNPI using HammerCloud test N20146370 from Template 1099 (copy2scratch). Direct read on the left, dedicated xCache on the right.

We have made several improvements in our testbed configuration and software in the last 5 months:

- New site was added – MEPhI (Moscow).
- New Torque with task affinity patches was installed.
- New ARC CE was deployed at MEPhI.
- New monitoring (ELK) was designed and implemented.
- New node-local tests were added.
- Network backbone was improved at MEPhI.

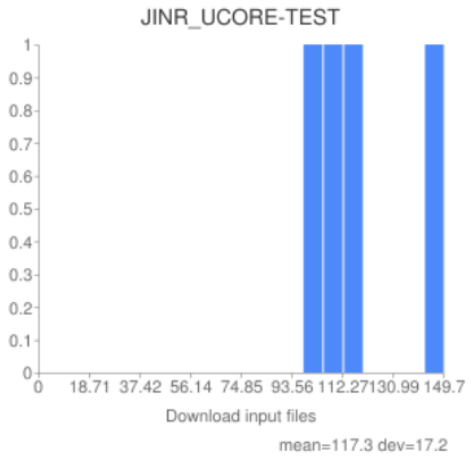

Fig. 6. Download input files time at JINR using HammerCloud test N20146370 from Template 1099 (copy2scratch). Direct read only.

The following synthetic tests were carried out taking into account the average number of requests to a single file (20) and the fact that the task can land on a random worker node, which is important in the case of a local cache (case 2 in Fig. 2). Tests were carried out in batches, since the load and available bandwidth of the external network is variable and it was necessary to compare different caching schemes in the same external network conditions.

Figures 7, 8, and 9 show the results of these tests. PNPI was tested with both dedicated and local caches, while MEPhI only with dedicated cache because of the shortage of local disk resources on the worker nodes. The results clearly show the benefit of using a dedicated cache for both sites, which is a bit unexpected for MEPhI, since the local network there is worse than the external one, and no improvement from using the cache was expected. At the same time, we can see minimum benefits from using the local cache which are within the margin of error.

HammerCloud tests were carried out in two scenarios only: direct read and dedicated cache, as there were technical problems registering a site with a local cache in the ATLAS information system (AGIS). The tests themselves have also changed since 2019, in particular, the template for test jobs was changed from HITS (digitization and reconstruction) to Derivation (AOD and DAOD) which is more I/O-intensive and have a larger input file size per event than with HITS. Figures 10 and 11 show the results of comparative tests using HammerCloud copy2scratch template (the input file is entirely downloaded to the working node before execution) for PNPI and MEPhI, respectively.

It can be seen that in all cases the gain from using the cache is obvious, which is expected, since in these tests there was no limit on the number of reads of a single file. Also, the gain in download input files time accurately matches the overall gain in the total time of the task execution.

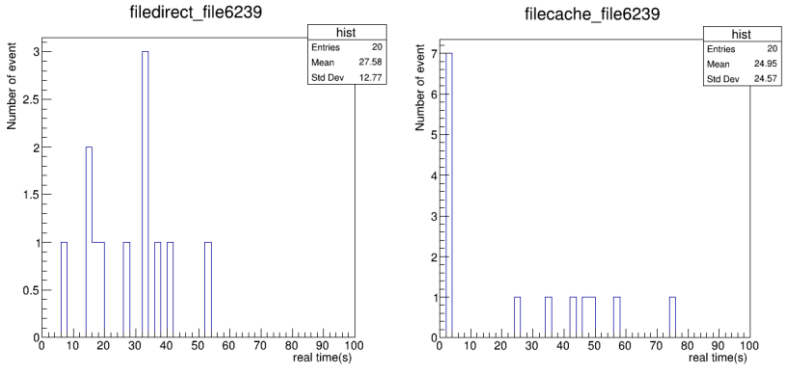

Fig. 7. Synthetic tests at PNPI. Direct access on the left, local cache on the right.

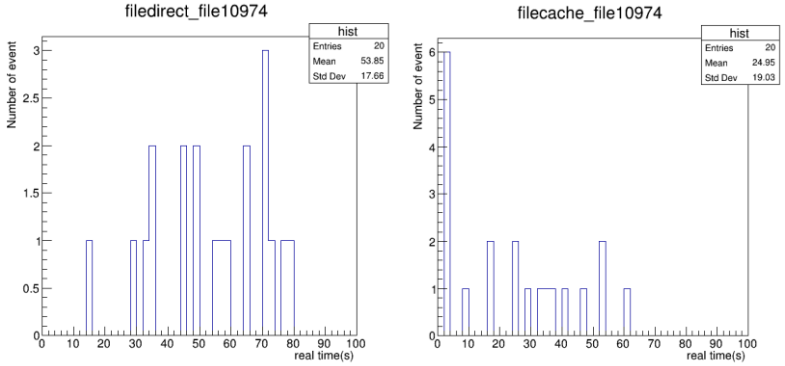

Fig. 8. Synthetic tests at PNPI. Direct access on the left, dedicated cache on the right.

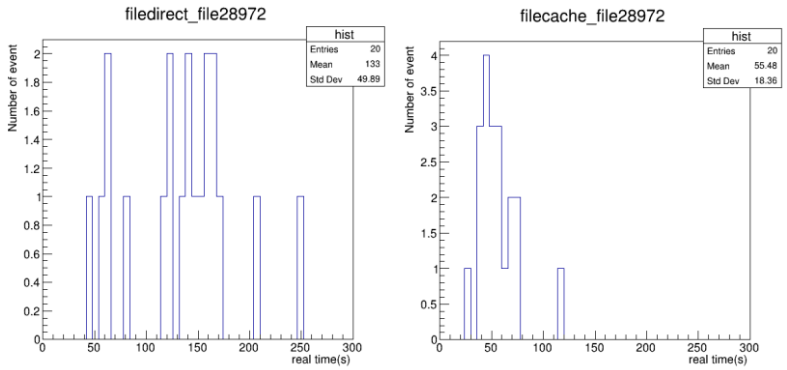

Fig. 9. Synthetic tests at MEPhI. Direct access on the left, dedicated cache on the right.

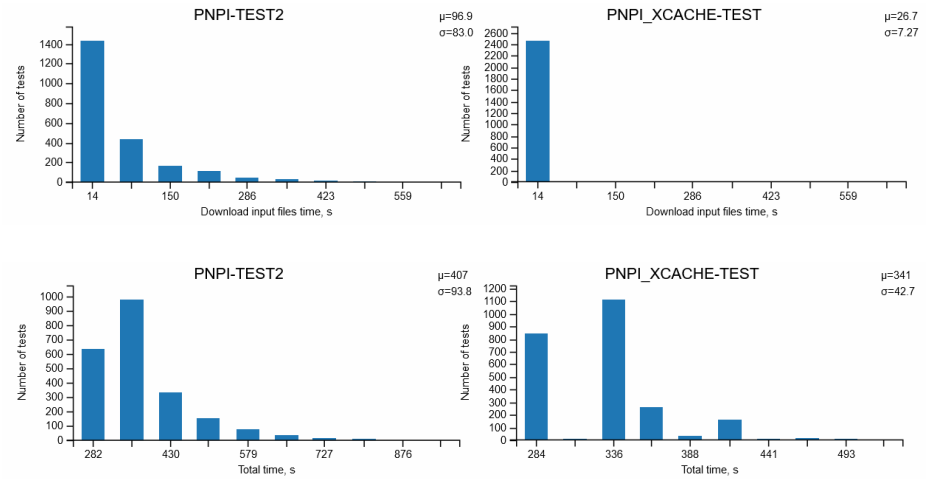

Fig. 10. HammerCloud tests (copy2scratch) at PNPI. Top row - download input files time (t_1) with direct read (left) and dedicated cache (right). Bottom row - total time (t) with direct read (left) and dedicated cache (right).

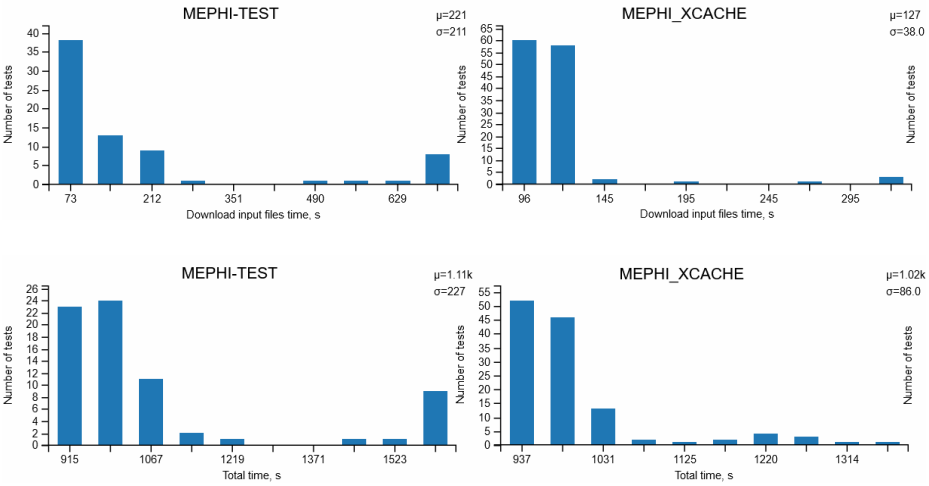

Fig. 11. HammerCloud tests (copy2scratch) at MEPhI. Top row - download input files time (t_1) with direct read (left) and dedicated cache (right). Bottom row - total time (t) with direct read (left) and dedicated cache (right).

6 Conclusions and future work

We have successfully passed “a pilot project phase” (PoC). During PoC, we have configured and tested two types of xCache setup: dedicated cache and local cache. We have shown performance benefits of using xCache on smaller sites using synthet-

ic and real-life ATLAS analysis workloads. Together with the WLCG community, we need to address the Data Lake challenge in a global context. The DOMA ACCESS initiative is the first step in this direction. We will work closely with DOMA and ATLAS to define the next steps, in particular we will be interested to test our setup for other HL-LHC R&Ds, such as Data Carousel, QOS and hot/cold storage, etc.

As a result of this work, we have observed an apparent benefit of a dedicated cache even for a limited number of requests to a single file, while for the local cache the benefit is severely doubtful. A dedicated cache, on the other hand, implies some additional operational and hardware costs that might not be justified by the expected performance benefits. The idea of a distributed cache on local nodes (case 3 on Fig. 2), which the authors see as very productive, still needs to be understood. Our near-term plans will include implementation and further evaluation of this idea.

7 Acknowledgements

It is a collaborative effort in ATLAS and WLCG (Operations, Distributed Computing, Software developers) and sites (JINR, MEPhI, SPbSU and PNPI). Thanks to all.

We thank Simone Campana and Xavier Espinal for discussions and their contribution to the tests methodology.

In Russia, this project is supported by the Russian Science Foundation, Project No. 19-71-30008 (the research is conducted in the Plekhanov Russian University of Economics).

References

1. Data Organization, Management and Access (DOMA). https://iris-hep.org/doma.html, last accessed 2020/06/25.
2. ATLAS HEP-Google R&D project. Technical Interchange Meeting. https://indico.cern.ch/event/921179/contributions/3870250/subcontributions/307490/attachments/2042093/3420405/RnD_HEPGCP.pdf, last accessed 2020/06/25.
3. A. Kiryanov, A. Klimentov, A. Zarochentsev et al, ATLAS Data Carousel Project. HEPiX Autumn Workshop, 14-19 Oct. 2019, Amsterdam, Nederlands.
4. A. Kiryanov, A. Klimentov, D. Krasnopevtsev, E. Ryabinkin, A. Zarochentsev, Federated data storage system prototype for LHC experiments and data intensive science, CEUR Workshop Proceedings, v. 1787, pp. 40-47.
5. J. Schovancova, 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.
6. X. Espinal, A. Kiryanov, A. Klimentov, J. Schovancova, A. Zarochentsev, Federated data storage evolution in HENP: data lakes and beyond, ACAT, 10-15 Mar. 2019, Saas Fee, Switzerland.