Analysis and Workload Characterization of the CERN EOS Storage System

Devashish R. Purandare
dpuranda@ucsc.edu
UC Santa Cruz
Santa Cruz, California, USA

Daniel Bittman
dbittman@ucsc.edu
UC Santa Cruz
Santa Cruz, California, USA

Ethan L. Miller
elm@ucsc.edu
UC Santa Cruz
Santa Cruz, California, USA

Abstract

Modern, large-scale scientific computing runs on complex exascale storage systems that support even more complex data workloads. Understanding the data access and movement patterns is vital for informing the design of future iterations of existing systems and next-generation systems. Yet we are lacking in publicly available traces and tools to help us understand even one system in depth, let alone correlate long-term cross-system trends.

In this work, we investigate the workload characteristics of the CERN EOS filesystem, analyzing over 2.49 billion events containing over 300 PB in reads and 150 PB in writes across 11 months. We contrast our finding with analyses from other scientific storage systems, allowing us to observe larger trends that appear over the years and revisit and question conventional wisdom such as "write once, read maybe" and the influence of user actions on system-wide data movement. By studying trace capture mechanisms across these systems, we motivate a standardized trace collection and analysis toolset, so that future researchers can more easily study existing systems to aid in system design.

CCS Concepts • **Information systems → Storage architectures**; *Magnetic tapes*; *Tape libraries*; *Magnetic disks*; • **Applied computing** → Physics.

Keywords Workload Characterization, Trace Analysis, File Systems, Scientific Archives, Archival Storage, Large Scale Storage, Supercomputing

ACM Reference Format:
Devashish R. Purandare, Daniel Bittman, and Ethan L. Miller. 2022. Analysis and Workload Characterization of the CERN EOS Storage System. In *Workshop on Challenges and Opportunities of Efficient and Performant Storage Systems (CHEOPS '22), April 5, 2022, RENNES, France.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3503646.3524293

CHEOPS '22, April 5, 2022, RENNES, France

ACM ISBN 978-1-4503-9209-9/22/04.
https://doi.org/10.1145/3503646.3524293

1 Introduction

Modern scientific and archival storage systems capture, store, and process large amounts of data. They often have custom designs build to navigate a wide variety of media characteristics such as cost, throughput, latency, and density. However, the decisions systems designers make around these characteristics is best informed by the future use of the system in a real data processing environment. While this naturally results in a system that grows and evolves over time, the ability to both improve existing systems and design next-generation systems is predicated on our ability to capture and analyze the traces of access patterns and data movement in the scientific storage and data processing systems of today.

The complexity of these systems, however, makes tracing and analysis quite difficult, especially when paired with the dependency of performance and cost on exact workflow and data processing models. These systems are often multi-tiered [2, 14], providing various performance and density characteristics for different tiers, thus navigating the cost/performance/density trade-off space by amplifying data movement through the network, and at the cost of complexity and coherence. Optimizations like de-duplication, prefetching, and caching are vital to avoid slowdowns, improve media lifetime, and reduce data movement, but without live tracing and deeper understanding of the emergent behavior of the system driven by real-world data access, bottlenecks are inevitable.

Unfortunately, we are sorely lacking in detailed analyses of these large-scale systems. In the last decade, only two such systems [2, 9] have been analyzed, since without internal access to research laboratories, few workloads are available for study. Further, it is difficult to draw generalized conclusions from the analyses of different systems, as each are custom built and use different data processing workflows. To address this, we propose creating a public repository of scientific storage traces and analysis tools that will enable researchers to extract, organize, and analyze different traces from various systems, and more importanly, correlate common patterns and trends between them. We kick off this work by adding another system to the analysis pile, the CERN EOS storage system. In particular, this paper looks at:

- **CERN EOS File System Workload:** Our observations about the workload span 11 months of operation the CERN EOS system. We observe that data

movement still dominates read-write activity and the continued existence of the "Write Once, Read Maybe" paradigm.
- **Trends in Large-scale Scientific Archives:** We observe the evolution of scientific archives, analyzing the change in workloads, as well as storage media.
- **Motivating Open Trace Archives:** We discuss the limitations of varied trace formats and the lack of availability of traces, and propose public trace repositories in a standardized format to enable better analysis and system design.

2 Related Work

In the last decade, there have been relatively few studies on trace analysis of scientific storage archives. Adams et al. [2–4] presented an analysis on NCAR MSS traces from 2008–2010, looking at the file size distribution, directory structure, and user characteristics. More recently, in 2015, Grawinkel et al. [9] presented an analysis on the ECMWF storage system, characterizing the workload of the storage archive, and analyzed caching techniques for such workloads. It is useful to compare how the CERN EOS workload that we study differs from the findings in these systems. We observe the change in magnitude of data over the years and evolution in system design. Further, we look at the trace capture formats across these systems and discuss the advantages and limitations of each technique and suggest best practices based on our observations. Our analysis, looking over 2.49 billion events, 188 million files, and observed reads and writes over 300 PB and 150 PB respectively is the largest dataset analyzed for a scientific archive.

We discuss the paradigm of "Write Once, Read Maybe" in Section 5, where most files are rarely updated after the initial write and rarely read. Grawinkel et al. [10] describe the Lonestar system that does aggressive power optimizations assuming such a pattern of reads and writes. Colarelli et al. [8] demonstrated the Massive Array of Idle Disks (MAID) framework to reduce power consumption in a write once, read maybe model. Pergamum [17] was an optimization over MAID suggested by Storer et al., which added NVRAM at each of the idle disks to provide high performance. These inspired the design of Internet Archive [11] by Jaffe et al. Our analysis looks at the analysis pool, a subset of the larger system, and yet we observe a similar workload, and these optimizations would help the CERN EOS system.

Jensen and Reed [12] looked at file archive activity at the National Center of Supercomputing Applications in 1993. In the same year Miller and Katz [15] performed an analysis on file migration in the NCAR environment. These studies establish file access patterns and observe that the writes remain relatively stable while reads fluctuate, as user actions cause most reads while writes are automated. Despite the differences between these systems in terms of time, scale, storage media, and application use, we observe similar read patterns. With our system, being in the analysis tier, we see that writes follow a similar fluctuating pattern as well, while writes to the system remain steady.

There have been several studies on analyzing storage changes over long periods. Agrawal et al. [5] looked at changing file system metadata. Breslau et al. [7] looked at web caching logs and presented a perspective on how Zipf's law applies to storage. This hypothesis is observed in our findings, and we observe the Zipfian distribution of reads and writes. A small set of users performs most reads and writes, and similarly, a small set of files see the majority of reads and writes. We observe the Zipfian distribution throughout the workload, from file sizes to the amount of file accesses.

3 Background

The European Organization for Nuclear Research (CERN) was established in 1954 as a joint research initiative between member states in particle physics. CERN captures scientific data from a range of particle accelerators, the most popular being the Large Hadron Collider. As of 2017, CERN had 230 PB of permanent storage on magnetic tape, 70 PB of which were captured in the same year [1]. To process and store this data, CERN uses a custom file system, known as the CERN EOS file system [16].

Four experiments share the EOS file system at CERN, namely Atlas, Alice, CMS, and LHCB [13]. From our collaboration with CERN, we acquired traces from the Compact Muon Solenoid (CMS) experiment. The traces document all system activity over 326 days, ranging from 13th October 2016 – 3rd September 2017. The total size of the trace files is 133 GB compressed. These traces capture the state of a file after an operation occurs on it. While we can often infer the type of operation, it is not explicitly recorded.

3.1 The CERN storage system

CERN uses a three tiered storage system (see Figure 1), providing data storage with various characteristics.

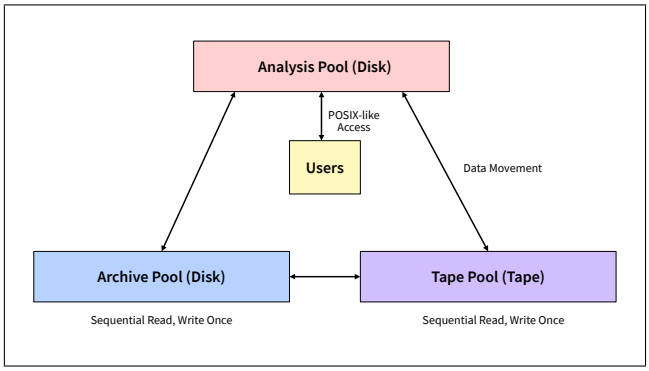

Figure 1. The multi-tiered CERN EOS storage system [16]

Analysis Pool: The Analysis pool is the storage that users directly access and can be up to a petabyte of spinning hard disks. Analysis pool is where data is fetched from the other pools to be processed. Updates to data are only possible in the analysis pool. This access uses POSIX-like file system semantics. *Our traces are from the Analysis pool.*

Archive Pool and Tape Pool: The Archive pool is intermediate disk storage, which only allows sequential read and write-once semantics. This pool is not open to direct user access and provides medium latency access to files, which can be a few milliseconds to a second. The Tape pool is high-density magnetic tape storage that puts batches of files into containers and supports sequential read, write-once access. Fetching data from the tape pool has a latency of 10^1–10^3 seconds—the tape pool stores cold data which is not in active use. Users do not have direct access to the tape pool. While the disk pool can go up to a few petabytes, limited by cost, the tape pools can go all the way up an exabyte. Data actively being processed is pulled into the analysis pool, and users get direct random read/write access to this data.

4 Workload Characterization

In this section, we will look at file size characteristics, read-write activity, and per-application statistics of system uses. We compare these with the trace analysis from NCAR and ECMWF.

	CERN – Our Work	**NCAR**	**ECMWF**
Timespan	2016–17	2008–2010	2012–2014
	11 months	3 years	$2\frac{1}{3}$ years
Events	2.49 Billion	188 Million	127.4 Million
Unique Files	188.7 Million	69 Million	137.5 Million
Storage Size	95 PB	11.7 PB	14.8 PB
Users	1977	1600	1190
Read Volume	300 PB	NA	7.24 PB
Write Volume	150 PB	NA	11.83 PB
Cache	NO[1]	YES	YES

Table 1. Comparison with ECMWF [9] and NCAR [2].

4.1 File Characteristics

The unique file identifier (`fid`) recorded in the traces allows us to track files. As the traces capture change in metadata of each `fid`, we can infer the nature of the operation (create, read, update, delete) of these files. However, files that did not see any operation are not recorded in these traces.

We can observe from Fig. 2, 90% of files are below 1 gigabyte in size but use only about 15% of the storage. When we consider the volume of storage, the Cumulative Distribution Function (CDF) curve gets steeper, with files between 1 gigabyte and 32 gigabytes making up about 63% of the volume,

[1]Data is processed in the analysis pool, there is no dedicated caching for this pool; however, this pool acts as a cache for other pools.

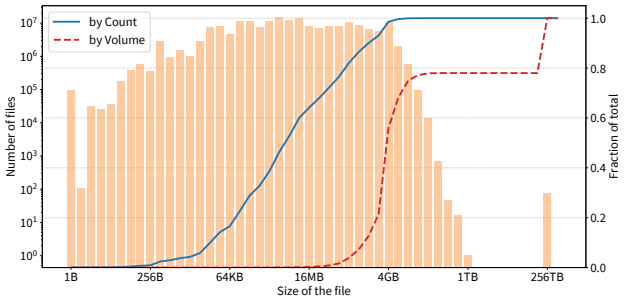

Figure 2. The Cumulative Distribution Function (CDF) for file size, by count and by volume, and the file size distribution histogram in CERN EOS system.

with twelve 256 terabyte files making up the rest (22%). These files are snapshots of virtual machines and would benefit from having a dedicated caching strategy.

Compared to similar analysis on NCAR [2] we observe that though the overall trend is similar, larger files occupy more volume in CERN as opposed to NCAR. Some of this can be attributed to the fact that these sets are years apart, and CERN files about collision data tend to be a couple of orders of magnitude larger than atmospheric data captured by NCAR. In the ECMWF analysis [9], files between 1MB and 48MB make up the overwhelming majority of the volume.

Observation: *In CERN EOS workload, larger files form a more significant share of total volume, with files below 64 MB making up less than 5% of data volume, even though they are more than 75% of all observed files.*

4.2 User Read / Write Activity

In our traces, we observe the existence of 188 million files, of which we observe the creation of 139 million files. While CERN does not specify the type of action, we specify creations as a `fid` starting at size (`osize`) 0 and the operation performing a write (written bytes `wb`) that is positive.

- We observe the creation (first write) of 139 million files, out of which 0.22% or 314,000 are written to twice.
- Less than 0.09% files see further updates.
- The most actively written file saw 81,000 updates.
- We see 85 million read actions, and 42% of files are read at least two times.
- 12.5% of those files see three or more reads

Observation: *Updates to files are rare, with 0.22% files being updated once, and only 0.09% files updated more than once. The workload of this system is Write Once, Read Maybe.*

4.2.1 Reads

We observe that the total amount of data read over the 326 days is 300 petabytes. Almost all users perform reads. At least half the users read more than four terabytes of data and

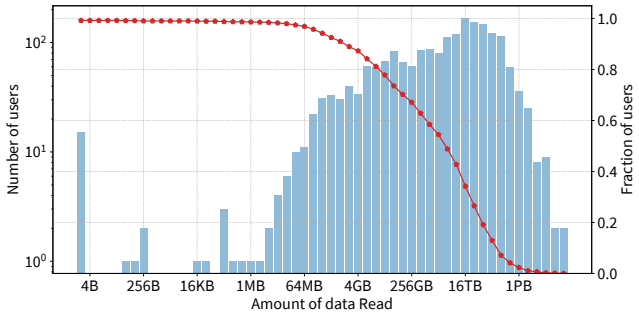

Figure 3. The Complementary Cumulative Distribution Function (CCDF) and histogram for the amount of data read by users (CERN).

as seen in Fig. 3, more than 80% of all users read at least 16 gigabytes of data.

Users	Top 1	Top 5	Top 5%
Data Read	42%	62%	89%
Data Written	27%	95%	99%

Table 2. Amount of data Read/Written by Users.

As seen in Table 2, the top reader read 43 petabytes of data, accounting for 42% of data read by volume. The top 5% of users by read volume make up about 89% of all read observed in the system. With writes, this unbalanced nature of a small set of users performing a large chunk of actions is even sharper. These top accounts are system users performing tasks such as replication, fetch, or dispatch of data.

Observation: *Most users perform relatively large reads on files within EOS with more than half the users reading at least 8 TB of data. The top reader, while being responsible for 42% of all data read, is a system user responsible for balancing, data movement, and replication.*

4.2.2 Writes

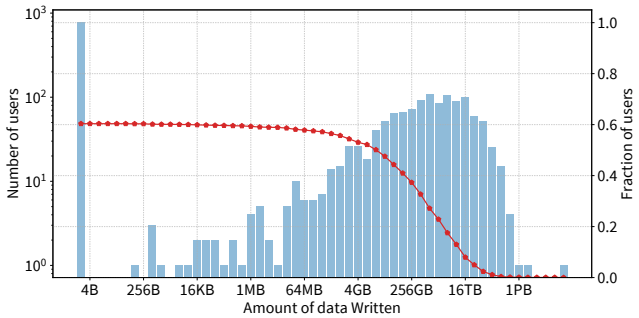

Figure 4. The Complementary Cumulative Distribution Function (CCDF) and histogram for the amount of data written by users (CERN).

We see significant differences between User Read Statistics: Fig. 3, and User Write Statistics: Fig. 4. As many as 782 users, which make up almost 40% of all users, perform no writes to the system. Of the users who write data to the system, at least half write more than 16 gigabytes. The writes associated per user are much smaller than the reads. This is not surprising for a scientific data archive, where most captured data is refined and aggregated to produce results.

Observation: *40% (782) of users do not perform any writes, while we see relatively large writes by a small set of users. 95% (142 PB) of observed writes by volume are performed by five users, which includes system users responsible for replication, as well as applications like FTP.*

Our observations are in contrast to the ECMWF analysis, which sees a 2:1 write to read ratio. This is caused by the nature of the analysis pool, where data of interest is fetched and processed. While writes will outnumber reads at the system level, in the analysis pool this is reveresed as only the data to be processed is fetched. We cannot observe data directly stored on the tape archives, and if the users request a particular set of data, it is likely to see reads. Further, operations such as replication and third-party-copy read data but write it to a remote location, causing larger reads than writes in the system under consideration.

A small subset of users perform most writes, as seen in Table 2. The top writer wrote 42 petabytes of data and was responsible for 27% of all data written. The top 5 writers wrote 95% of all the write volume, and the top 5% made up almost all writes in the system. Most of the users that write large amounts of data are system users that handle data movement and copying. We make this inference using the User ID field associated with each action. UID 0 is the root, and system users have UIDs below 100. It is crucial to break this usage down into a user and system-user comparison.

Observation: *As EOS analysis pool contains user-requested data, we observe that reads outnumber writes at least 2:1 by volume in our traces, in contrast to related work. A relatively small number of users perform most reads and writes, and updates to files are rare (<0.09% files receive updates).*

4.3 User Workloads

	Actions	Written	Read	Affected Files
System	204.75 Million	26.03 PB	26.05 PB	38.60 Million
User	2214.87 Million	131.40 PB	285.20 PB	119.92 Million

Table 3. Reads and writes between user and system actions.

When we break down the tasks by the user accounts (`uid`) performing the task, we see that system accounts perform about 9% of all actions. They read about 26 petabytes of data and write roughly the same amount of data. Since system

tasks are mainly load balancing, replication, and data movement, these numbers are very similar. We can further break down system tasks as seen in Table 4.

4.3.1 System Workloads

Application	Actions	Written	Read	Affected Files
Archive	4	0 B	70.56 GB	2
Converter	500	0 B	0 B	241
Balancing	154.38 Million	20.24 PB	20.24 PB	38.60 Million
Draining	50.23 Million	5.74 PB	5.74 PB	22.72 Million
Replication	129,581	60.87 TB	72.83 TB	27,132

Table 4. System tasks.

System user accounts perform maintenance tasks such as balancing and replication. The system writes, on average, are significantly larger in granularity than user writes as they involve copying sets of data. While system tasks cause a lot of data movement, most read and write requests come from individual user accounts.

4.3.2 User Applications

Application	Actions	Written	Read	Affected Files
GridFTP	92.13 M	33.5 PB	45.75 PB	46.28 M
EOScopy	12.61 M	376.38 TB	270.3 TB	3.82 M
Filecopy	174	0 B	0 B	89
FUSE	200.57 M	0.49 PB	2.88 PB	19.79 M
Other	1894.60 M	85.24 PB	229 PB	119.92 M
Point5	5.43 M	7.4 PB	0	2.78 M
Restore	13,122	15.65 GB	0	6,304
TPC	9.50 M	4.32 PB	7.24 PB	6.68 M

Table 5. User applications.

Several user applications on EOS are used to fetch data `EOSCopy`, `Filecopy` are used to move data within the filesystem, `GridFTP` is the file transfer protocol mount used for getting and putting data using FTP and performs a significant amount of writes (more than all system tasks) and reads. `FUSE` is a popular userspace file system interface. Here, `Point5` refers to all the tasks related to the Point5 endpoint of the Large Hadron Collider. `TPC` is the third-party copy tool used by researchers outside of CERN to access CERN data. Most user reads and writes are not tagged by a particular application type.

Observation: *While data movement between the various pools is the cause for a majority of reads and writes in the system, most of the data movement happens on user requests rather than system actions. This is evident from the read and write statistics of various internal and external copy utilities.*

5 Discussion

Trace analysis is a common technique used in several studies of large-scale systems. In this section, we compare conventional wisdom to observations from the CERN data and include similar analyses such as NCAR [2], and ECMWF [9].

Analyze System Components in Isolation: Analysis from earlier systems observed that writes were much larger than reads, however, in the CERN EOS system, when we focus on the analysis pool where data is fetched on-demand, we see twice the reads by volume as compared to writes. Any design considerations for the analysis pool should make it read-optimized, as more than half the users read more than 4 TB of data while writing up to 16 GB of data.

Filesystem Metadata can Inform File Activity: Filesystems can provide information about the type of file, whether it is immutable, expected size, along with expected read-write activity. Since a small set of users perform most actions, the users associated with a file can inform us about the potential reads and writes to the file. These metadata fields in the filesystem can help us make placement decisions, picking the right storage media, as we upgrade these systems with flash and non-volatile memories. Bel et. al [6] looked at utilizing trace features to train deep neural networks for optimal data placement across heterogeneous media. This analysis used the features represented in the data, and could be expanded to inferred features, like the path in the filesystem, or users associated with activity on a file.

Write Once, Read Maybe Depends on your View: "Write Once, Read Maybe" is a common assumption in archival systems. Most archival data is written once, rarely updated, and may not be read at all. As seen in Fig. 5, we observe semantics similar to write-once. Only 0.22% of files see updates, going down to 0.09% of all files for more than one update. On the other hand, the read graph shows that while most files are read once, many files are read a second and a third time in a Zipfian heavy-tailed distribution.

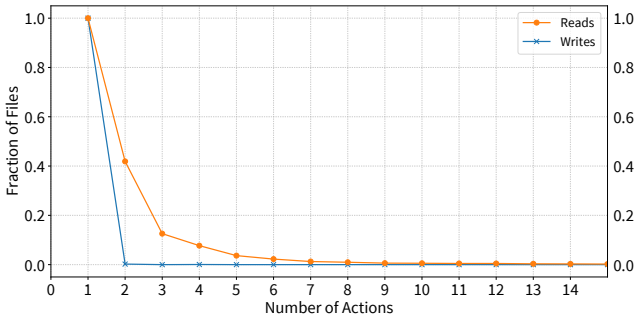

Figure 5. CERN : Likelihood of reads and writes to a file

Migrations make up most of the activity, but can be user-triggered: In NCAR MSS traces, it was observed that system actions made up 66% of all actions observed in the traces. Most of these were automated migrations. In CERN traces, while most actions are related to the migration of data, almost all of these actions are user-initiated. CERN archive also differs from traditional archives, as it is an active processing system, and the traces we analyzed are from

the disk cache where data is fetched and processed. As we saw in Table 3 only 9% actions in the CERN system are automated, while 91% are user-driven. Hence, system optimizations should not make any assumptions about data movement being the sole domain of the system.

Reflection on Trace formats: CERN traces record the state of each file that was accessed, noting the timestamps, users, written and read bytes, and a variety of information about the file. We need to infer the rest of the data from these records. The CERN format helps observe how files change over time but does not capture actions. ECMWF traces include a system snapshot and a description of the event; they further include metadata snapshots from tape archives, which describe the files, users, and other information. NCAR traces, on the other hand, recorded actions. So the trace records actions like CREATE and MIGRATE, and the associated metadata, including the user performing the action and its effects. Getting uniform and consistent analysis across these formats can be difficult, or in some cases impossible, as they may capture different aspects of an action.

Tracing in the Future: Part of the trouble with such cross-system correlation, is managing the different trace formats. The problem goes beyond simple encoding and representation, but permeates through the choices the designers make on which fields to record and which operations are meaningful, all the way to minor semantic differences between fields in different traces. Thus, any effort into future cross-system correlation of trends using an understanding of workflow will require some heavy lifting on part of researchers before they can begin an analysis. Future systems built should include a plan for tracing from the start, and should use lessons learned by previous trace analysis to inform the design of trace capture, ideally in a well-defined format to make integration with other tracing systems possible.

While just studying individual systems provides valuable insights into *that* system and its operation, this field is held back by a lack of cross-system correlation of performance and cost characteristics with (a) what a scientific computing platform is built for — primarily data collection, or active data processing — and (b) the technology and system architecture wisdom of the time. Live tracing gives us insights, but the lack of publicly available traces and tooling considerably limits system designers. We plan to formalize this project into a larger more modular framework for trace analysis. The tools we built to understand the CERN EOS system are, of course, purpose built for that system, but we plan to continue studying trace formats so that we can better generalize our work into a uniform tracing format that we can argue makes for easier system analysis and cross-system comparisons. Such a system will enable researches to more easily study and build future systems by being able to learn from many systems architectures, their live traces and usage characteristics, and how those traces are correlated with the type of computing done.

We will continue studying the CERN EOS system and comparing to other system traces. While the studies we performed in this paper are a good start, there is still much to understand. For example, since our analysis was focused on comparison to other tracing work, we ignored some features present in the CERN EOS traces, including information about how data was distributed across physical devices. This will allow us to study hardware failures as part of the system analysis and the performance of disks over time and how that affects user-perceived performance.

6 Conclusion

This work presented workload characterization of the CERN EOS filesystem and compared it with other large-scale scientific archives. Even with changes in data volume, storage media, and storage techniques over decades, the CERN EOS workload follows the fundamental trends in large-scale storage, such as a Zipfian distribution for access, Write Once (rare updates to data), and Read Maybe (most data is never read). But as we look at the CERN EOS analysis pool in isolation, our view differs from other large-scale systems. Data movement in the CERN system is user-triggered, and we see twice the volume of reads compared to writes. Since only the data of interest is fetched to this tier, the reads outnumber writes both in number and volume. We observe that it is essential to look at individual components of a larger system in isolation to suggest optimizations.

With limited data points for such analysis, it is difficult for academic systems researchers to understand and analyze large-scale archives without direct access to large research institutions. Cross-system analysis is vital, however, to avoid becoming stuck with wisdom derived from a limited understanding of few systems. Since these systems often differ both architecturally and in the intent of their usage, observing similarities and differences across many diverse systems will lead to a better, generalized understanding of how to build them for the future. Looking forward, we must not only put forth real effort into making tooling and data sets publicly available, but also commit to updating our understanding as these systems and their use cases evolve, by continuing the perform analysis and comparisons between systems. With this "call to arms" we hope that we can develop future tooling for uniform trace collection and analysis, making it easier to study long term trends across many diverse large-scale scientific computing systems.

Acknowledgments

This project was supported in part by the National Science Foundation under Grant No. IIP-1266400 and Grant No. IIP-1841545, and the industry members of Center for Research in Storage Systems (CRSS) at UC Santa Cruz. We are grateful

to the European Organization for Nuclear Research (CERN) sharing their file system traces and the help we received from Dr. Dirk Duellmann. We are grateful to Matt Bryson, and Darrell D. E. Long for their help in acquisition and the storage of these traces. We thank Shel Finkelstein and Peter Alvaro for their feedback and advice which helped make this work stronger. We would like to thank the anonymous referees for their valuable comments and helpful suggestions.

References

[1] CERN Annual report 2017. Tech. rep., CERN, Geneva, 2018.

[2] Adams, I., Madden, B., Frank, J., Storer, M. W., and Miller, E. L. Usage behavior of a large-scale scientific archive. In *Proceedings of the 2012 International Conference for High Performance Computing, Networking, Storage and Analysis (SC12)* (Nov. 2012).

[3] Adams, I. F. *Understanding Long-Term Storage Access Patterns*. PhD thesis, University of California, Santa Cruz, 2013.

[4] Adams, I. F., Storer, M. W., and Miller, E. L. Analysis of workload behavior in scientific and historical long-term data repositories. *ACM Transactions on Storage 8*, 2 (2012).

[5] Agrawal, N., Bolosky, W. J., Douceur, J. R., and Lorch, J. R. A five-year study of file-system metadata. In *Proceedings of the 5th USENIX Conference on File and Storage Technologies (FAST '07)* (Feb. 2007), pp. 31–45.

[6] Bel, O., Chang, K., Tallent, N., Duellman, D., Miller, E. L., Nawab, F., and Long, D. D. E. Geomancy: Automated performance enhancement through data layout optimization. In *Proceeding of the Conference on Mass Storage Systems and Technologies (MSST '20)* (Oct. 2020).

[7] Breslau, L., Cao, P., Fan, L., Phillips, G., and Shenker, S. On the Implications of Zipf's Law for Web Caching. In *3rd International WWW Caching Workshop* (June 1998).

[8] Colarelli, D., and Grunwald, D. Massive arrays of idle disks for storage archives. In *Proceedings of the 2002 ACM/IEEE Conference on Supercomputing (SC '02)* (Nov. 2002).

[9] Grawinkel, M., Nagel, L., Mäsker, M., Padua, F., Brinkmann, A., and Sorth, L. Analysis of the ECMWF storage landscape. In *Proceedings of the 13th USENIX Conference on File and Storage Technologies (FAST '15)* (Feb. 2015), pp. 15–26.

[10] Grawinkel, M., Pargmann, M., Dömer, H., and Brinkmann, A. Lonestar: an energy-aware disk based long-term archival storage system. In *Proceedings of the 17th International Conference on Parallel and Distributed Systems (ICPADS '11)* (2011), pp. 380–387.

[11] Jaffe, E., and Kirkpatrick, S. Architecture of the Internet Archive. In *Proceedings of The Israeli Experimental Systems Conference (SYSTOR '09)* (May 2009).

[12] Jensen, D. W., and Reed, D. A. File archive activity in a supercomputer environment. Tech. Rep. UIUCDCS-R-91-1672, University of Illinois at Urbana-Champaign, Apr. 1991.

[13] Lamanna, M. The LHC computing grid project at CERN. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 534*, 1-2 (2004), 1–6.

[14] Li, Y., Bel, O., Chang, K., Miller, E. L., and Long, D. D. E. CAPES: Unsupervised storage performance tuning using neural network-based deep reinforcement learning. In *Proceedings of the 2015 International Conference for High Performance Computing, Networking, Storage and Analysis (SC17)* (Nov. 2017).

[15] Miller, E., and Katz, R. An analysis of file migration in a Unix supercomputing environment. In *Proceedings of the Winter 1993 USENIX Technical Conference* (Jan. 1993), pp. 421–433.

[16] Peters, A. J., and Janyst, L. Exabyte scale storage at CERN. *Journal of Physics: Conference Series 331*, 5 (dec 2011), 052015.

[17] Storer, M. W., Greenan, K. M., Miller, E. L., and Voruganti, K. Pergamum: Replacing tape with energy efficient, reliable, disk-based archival storage. In *Proceedings of the 6th USENIX Conference on File and Storage Technologies (FAST '08)* (Feb. 2008).