Access Pattern Analysis in the EOS Storage System at CERN

Olga Chuchuk[1] and Dirk Duellmann[2]

[1] Taras Shevchenko National University of Kyiv (KNU), Kyiv, Ukraine
CERN, Geneva, Switzerland
olga.chuchuk@cern.ch
[2] CERN, Geneva, Switzerland
dirk.duellmann@cern.ch

Abstract. EOS is a CERN-developed storage system that serves several hundred petabytes of data to the scientific community of the Large Hadron Collider (LHC). In particular, it provides services to the four largest LHC particle detectors: LHCb, CMS, ATLAS, and ALICE. Each of these collaborations uses different workflows to process and analyse its data. EOS has a monitoring system that collects detailed information on the file accesses and can give important insights about the specifics of the physics experiments' workflows. In our study, we analyse the monitoring information accumulated over a six months period and amounting to over 1.3 terabytes and have the goal to help the IT department and the experiments' operations teams to better understand the EOS data flows.
In this contribution, we describe a pipeline, mainly developed in R, for processing large volumes of access logs and perform a comparative analysis of the storage usage in scientific workflows. In particular, we calculate aggregated statistics over a six months period and provide a high-level overview of the experiments' data flows. Additionally, we study how the frequency of data accesses changes over time and estimate to what extent different experiments may benefit from an additional caching layer.

Keywords: Data monitoring · Access patterns · Storage system · Data popularity.

1 Introduction

The Large Hadron Collider (LHC) is the world's largest and most powerful particle accelerator. It is a massive and long-lasting project and therefore requires state-of-the-art approaches to the tasks of data storing and processing. The four largest particle detectors (ALICE, ATLAS, CMS, and LHCb) are located along the LHC ring. Each of them is a large international collaboration that brings together scientists with different backgrounds.

The European Organization for Nuclear Research (CERN) provides an infrastructure for the high-energy physics scientists working at the LHC and its particle detectors. In particular, EOS is a multi-purpose storage system used for the experiment measurements and user analysis data and has been developed at CERN since 2010. As of today, EOS operates over 320 PB of raw disk space and provides multi-protocol, secure access and multi-user management.

EOS aims to provide a reliable service that satisfies the needs of CERN experiments and users. Nevertheless, the scale of the experiments and the diversity of the physics community make it difficult for the operations teams to monitor the system and to adapt it to the ever-increasing user needs. Since the main target area for the EOS service is physics data analysis, it is characterised by many concurrent users, a significant fraction of random data accesses and a large file-open rate.

In this work, we perform a study of EOS as a large distributed storage system in order to help the IT development and operation teams to better understand the needs of the LHC physics experiments. We implement and describe a pipeline for processing EOS access log files and perform a comparative analysis of the four EOS instances serving the needs of LHCb, CMS, ATLAS, and ALICE experiments. We explore the differences between the instances' data workflows. In particular, we give a high-level overview of data life cycles and describe how data popularity changes over time.

The paper is organized as follows. After outlining the background for our research in Section 2, we formalize the problem and describe the analysis pipeline in Section 3. In Section 4, we present the results and interpretations and Section 5 concludes the paper.

2 Background

CERN operates multiple EOS instances, including one for each LHC large particle detector. Each EOS instance consists of metadata servers (also called management (MGM) nodes) and up to several hundred disk servers (file storage (FST) nodes). The MGM servers store lists of filenames together with metadata such as replica numbers, owner rights, timestamps of the main operations (file creation, last update, last access, etc.) and so on. Each disk node contains 24-72 disks and keeps the content of the files. In total, today EOS has 4.8×10^4 disks and stores 6.1×10^9 files.

In order to reduce the data access latency and to optimize the data placement and replication policies, the first version of an XRootD-based proxy cache system was introduced [1]. The later version (XCache) benefits from asynchronous reads, better resource management, full support of vector reads and several other improvements [2].

The CERN Data Centre has been monitored, for more than a decade, with in-house central solutions gathering and storing at the CERN storage facilities a large number of metrics and logging information.

As part of the EOS monitoring system [3], a high volume of detailed logging data has been collected. The log files reflect all the system events (file creation, movement, replication, deletion etc). Today, EOS monitoring collects information about the total data volume, speed of data taking, etc. In this work, we make use of EOS access log files and additionally explore the patterns of data movement within each experiment and working group, such as typical file workflows and times between data creation and deletion.

This work is performed using the analysis utilities of the CERN Data Centre including some large memory machines available for the research needs. As the development tools, we used the R programming language together with the interactive development environment RStudio.

3 Access Log Analysis

The life cycle of files stored in EOS usually consists of three main steps:

1. Creation (a new file coming directly from an experiment, a new file generated by a user as a part of his/her analysis, or a copy of an already existing file generated for the service reliability);
2. Access (the number of accesses can vary from zero to a very large number);
3. Deletion (most of the files coming directly from the physics experiments are not meant to be deleted and are stored in EOS forever).

The process of deletion is not atomic and usually consists of two main steps:

- a version of the file is deleted from a local disk (FST deletion). In this case, other replicas of this file can still exist in the system. Regardless of the number of replicas left, this type of deletion does not erase information about the file on the management node;
- deletion from the management node (MGM deletion). This happens when the file is deleted from the system altogether. Usually, this deletion is propagated to the FST where the local versions of the files are deleted.

The EOS access log files are formatted as text files and each line represents a separate log record [4]. Each line contains a fixed number of key-value pairs and is encoded in the following format: key1=val1&key2=val2&...&keyN=valN.

EOS produces three types of access logs [4]. The first type corresponds to file creations and accesses (event records). These records contain close to 60 metrics: log record, file, filesystem and user identifiers; logical file path; file size on opening and closing; read and write rates; opening and closing timestamps; trace identifier from where the operation was requested and so on.

Another type is deletion records and it has two subtypes that correspond to FST and MGM deletions. These records contain fewer metrics and store the following information: log record, file and filesystem identifiers; file size before the deletion; timestamps of the last change in metadata, the last modification in the file content, the last access and the deletion itself.

The MGM deletion records are more representative of the user workflows comparing to the FST ones since the former is only produced when a user explicitly deletes a file. FST deletions, on the other hand, could be triggered by internal system processes that are not user-related, like balancing.

3.1 Analysis Pipeline

The EOS access logs analysis starts with a daemon running every night to collect log records from the management nodes, where they are generated, to the machinery for further processing. This data is parsed and saved in a more convenient tabular format. At this stage, the log records are separated based on their type (events or deletions). In this study, we use log records only of the first type.

The data filtering step is vital since the raw volume of the analysed log files reaches more than 1.3 terabytes. In order to separate user activity from the system events, we exclude log records that have 'daemon' username as part of the trace identifier[1] and/or have 0, 1 or 2 as the real-user identification number. After that, we reduce the number of metrics and leave only the ones of interest for our research: file identifier, trace identifier, file size on opening and closing, amount of bytes read and written, opening and closing timestamps.

Since there can be more than one opening of a file during one session, we merge log records that have the same file identifiers and happen within one session. For that, as a global session identifier, we use a substring of trace identifier (username, local process identifier and origin host parameters). After aggregation, we update the rest of the metrics within one session as follows:

- opening file size is the opening file size from the record with the earliest opening time;
- closing file size is the closing file size from the record with the latest closing time;
- opening and closing timestamps - the earliest and the latest timestamps accordingly;
- amount of bytes read and written - the sums of all the read and written bytes accordingly.

The operation type (create, read, update, etc.) is not present in the log records explicitly, but can be derived from the existing metrics. We use the following heuristic based on the four metrics (osize, csize, rb, wb - opening file size, closing file size, read bytes, written bytes) to classify records into the following five categories:

1. Creation (osize == 0 **and** csize > 0 **and** wb > 0 **and** rb == 0)
2. Read (osize > 0 **and** csize == osize **and** wb == 0 **and** rb > 0)
3. Update (osize > 0 **and** csize > 0 **and** wb > 0)
4. Noop (wb == 0 **and** rb == 0)

[1] The structure of the trace identifier field is
<user_name>.<process_id>:fd@<origin_host>[.<domain>]

5. Abnormal (csize == 0 **and** (wb > 0 **or** rb > 0))

According to the specifics of the typical flows of physics data in EOS, most of the data is coming directly from the experiments, stored in EOS forever and is not updated afterward. To prove this assumption, we check the number of 'Update', 'Noop' and 'Abnormal' operations to verify that they do not have a significant influence on our research. Furthermore, some files have a phenomenon of multiple 'Creation' operations. This can happen when a file was 'Created' with csize > 0, then the content was removed with an 'Abnormal' operation with csize == 0. The subsequent update operations will be misclassified as 'Create'.

In Table 1, we present the total fraction of files with 'Update', 'Noop', 'Abnormal' operations and/or multiple creations. Both the number and the volume of such files constitute less than one percent for every experiment. Therefore, we assume that most of the traffic on EOS instances for the LHC detectors is 'Create' and 'Read' operations and that the majority of data files are immutable.

Table 1. Fraction of files with 'Update', 'Noop' and 'Abnormal' operations and/or multiple 'Create' operations

Metric	**LHCb**	**CMS**	**ATLAS**
Other Operations, % of related files	0.06	0.26	0.61
Other Operations, % of Instance Volume	0.89	0.05	0.14

After processing daily data, we aggregate it over the full six months period that we considered in our study (from 01/01/2019 to 30/06/2019). This allows us to obtain robust statistical results and avoid insignificant fluctuations that happen over shorter periods of time.

In the following section, results are shown for ATLAS, CMS, and LHCb. Since the ALICE experiment generates the highest volume of log files and given the available computational resources, we obtained aggregated statistics for this instance only for a three months period. We do not present the ALICE results here since the purpose of this paper is to present the comparative analysis of the experiments' workflows on the same timescale.

4 Analysis Results and Interpretations

Most of the calculated metrics significantly vary in absolute numbers from experiment to experiment. Therefore, we decided to additionally compare the obtained metrics with the total instance volume. The result is presented in Table 2. To obtain the total instance volume, we extracted the respective quantities (total space, used space) from the EOS operation web console. This measure is slightly changing over time, but an average approximation over a one year period is precise enough to serve as a reference for our analysis.

Table 2 reveals that LHCb is the smallest experiment and ATLAS is the largest in terms of total operated volume. 'Total Accesses' shows the experiments' turnover (the total amount of read and written bytes). For LHCb and ATLAS, this number reaches more than 300% of the total instance volume, which is significantly higher compared to CMS, where it is less than 200%.

The row 'Writes' shows the total amount of written bytes and the row 'Reads' – the amount of bytes read only during reading operations. Over a period of six months, the LHCb experiment produces a data volume larger than the total instance size, which implies that there is a high number of deletions and that the data is constantly updated. In contrast, ATLAS has a relatively low rate of write operations, but the most intense read traffic; CMS has the lowest read rate amongst experiments.

The last row 'Repeated Reads' shows the fraction of repeated read traffic. For ATLAS and CMS, these fractions are high with respect to the total read workload and account for approximately 80% of it. For LHCb, only 30% of the read workload is repeated and hence we estimate a reduced potential profit from caching.

Table 2. Read/Write workload

Metric	**LHCb**	**CMS**	**ATLAS**
Instance Volume (EOS Control Tower), PB	16.9	40.5	56.3
Total Accessed, PB	55.4	76.0	175
Total Accessed, % of Instance Volume	328	188	310
Writes, PB	22.5	31.5	29.4
Writes, % of Instance Volume	134	77.7	52.3
Reads, PB	32.0	44.4	145
Reads, % of Instance Volume	190	110	257
Repeated Reads, PB	9.8	34.6	122
Repeated Reads, % of Read Workload	30.7	77.8	83.7

Additionally, the derivation of file-specific quantities has enabled us to compare the total volume of the created files versus the read files (see Table 3). 'Created Volume' is defined as the total volume of files with 'Create' operations and 'Read Volume' is the total volume of files with 'Read' operations. 'Repeated Read Volume' indicates the total volume of files with more than one 'Read' operation during the monitored period.

The LHCb experiment has a well-organized workflow. It produces and reads a big amount of data, most of it goes through 'Create' → 'Read' → 'Delete' cycles and is not re-read very often. Also, it has a high rate of deletions, which indicates that it has a limited in storage space.

On the other hand, CMS and ATLAS produce fewer data with respect to their instance size. The read rates show that they mostly use only (50-60%) of their space. Nevertheless, the chances of data re-uses are higher than those for LHCb.

Moreover, we discovered that the amount of read bytes is often significantly smaller than the total file size. Therefore, we included the statistic that shows which fraction of a file is read on average. For ATLAS and LHCb, this number is approximately 80-90%; for CMS, only 55% of a file is read on average. The lower numbers indicate that the caching systems will work less efficiently if they support only a full-file prefetching mode. For example, an average read fraction of 50% means that potentially only half of the cache space will serve its purpose.

Table 3. Created/Read volume

Metric	**LHCb**	**CMS**	**ATLAS**
Instance Volume (EOS Control Tower), PB	16.9	40.5	56.3
Created Volume, PB	22.9	31.5	29.4
Created Volume, % of Instance Volume	136	77.7	52.2
Read Volume, PB	25.6	24.8	29.9
Read Volume, % of Instance Volume	151	61.1	53.2
Repeated Read Volume, % of Read Volume	20.7	54.7	55.1
Average Fraction of File Read, %	88.8	55.1	81.6

To compare the file sizes for the experiments, we plotted the file count density distribution over the file size (see Figure 1). The density plots are normalized and the total area under the curve equals one.

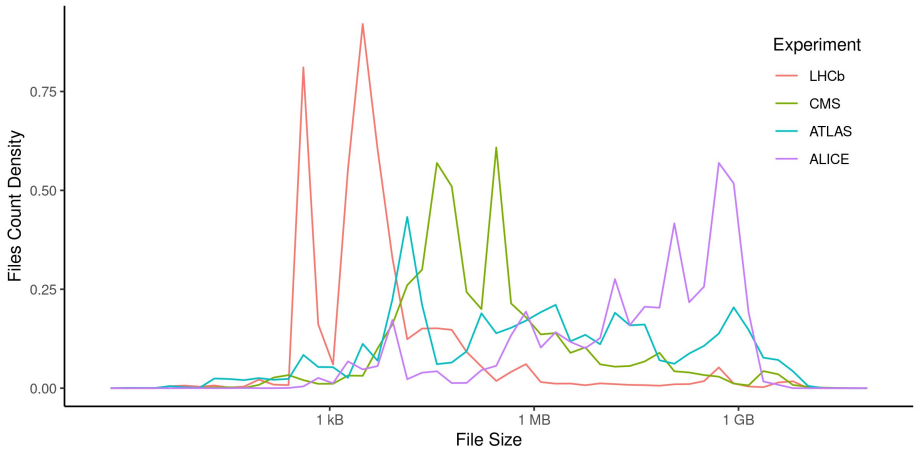

Fig. 1. Normalized probability distribution of file sizes.

The peaks on the plot show that experiments have different preferences for file sizes. Most of the files at the LHCb instance are of size 1 kB approximately. For CMS, the most popular file size is around 100 MB. The ATLAS plot has two

peaks: around 10 MB and 1 GB. The ALICE instance has the biggest relative number of large files; it has a distinct peak around 1 GB.

Furthermore, we compared the usage rates of the newly created files with the ones that were stored in EOS before the monitored period (Table 4). We defined as 'New' the files that were created during the period of consideration, 'Old' are the ones that were created before 01/01/2019.

The LHCb experiment reads almost all of the data that it produces (∼95%); for ATLAS and CMS, these numbers are lower. On the opposite, most of the data that was stored since the beginning of the monitored period was never used afterward.

Table 4. New/Old reads

Metric	**LHCb**	**CMS**	**ATLAS**
Created and Read, PB	21.6	18.2	23.6
Created and Not Read, PB	1.2	13.2	5.7
Created and Read, % of Created Volume	94.4	58.0	80.4
Created and Not Read, % of Created Volume	5.62	42.0	19.6
Old and Read, PB	3.9	6.5	6.3
Old and Not Read, PB	12.9	35.1	45.5
Old and Read, % of Old Volume	23.5	16.0	11.2
Old and Not Read, % of Old Volume	76.6	84.0	88.8

We explored in more detail how the file popularity changes over time (see Figure 2). On this plot, the dots show the dependency between the accessed volume density over the time elapsed since the files' creation. As expected, we observe a decreasing likelihood of data accesses with the increasing time since the data creation.

We fitted this historical data to an exponential decay function:

$$y(t) = y_f + (y_0 - y_f)e^{-\alpha t}$$

where y is the accessed volume density, y_0 is the total created volume, y_f is a constant, t is the time elapsed since the creation, α is the decay rate.

The lines on the plot in Figure 2 represent the fitted exponential decay functions. The legend shows the decay rate and the residual sum-of-squares (RSS) error estimate. Additionally, we find a point in time t^* after which half of the volume will not be accessed again:

$$y(t^*) = \frac{1}{2}y_0$$

CMS data stays popular the longest: the rate α is 0.02 day^{-1} and t^* is almost 60 days. This means that the probability of a file revisit after 60 or more days since its creation is approximately 50%. ATLAS has a higher decay rate: α is 0.04 day^{-1} and t^* is only 20 days.

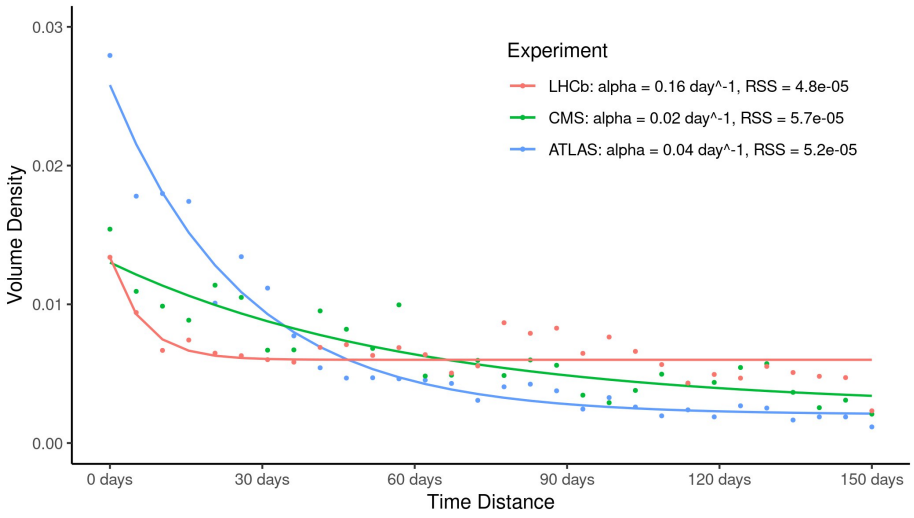

Fig. 2. Time distribution of file accesses.

When the LHCb data is fitted to an exponential decay function, the rate α is the biggest, amounting to 0.16 day^{-1}. The accessed volume never goes as low as $\frac{1}{2}y_0$, therefore the t^* point does not exist. Even though most of the files quickly become unpopular after the creation, there is some fraction of the data that always remains accessed.

5 Summary and Next Steps

We have performed a first-time analysis of the EOS access logs and developed a set of analysis tools to parse, clean and aggregate them. Using the refined data obtained in this way, we derived aggregated statistics over a six months period. In particular, we compared the total turnover and the read/write rates of the EOS instances dedicated to ATLAS, CMS and LHCb experiments. We also demonstrated how the probability of data re-usage decreases depending on the time elapsed since its creation.

In the future, we plan to expand and improve our analysis by using new tools for data processing and by obtaining updated statistics for the year 2020. Specifically, we plan to migrate our pipeline from R to Spark. This will remove some of the memory constraints of our R implementation and enable us to extend the monitored period as well as include ALICE into the comparative analysis.

With the expansion of the main LHC experiment and the construction of the new High-Luminosity LHC (HL-LHC), we expect a significant increase in data to be processed. This could lead to scaling problems in storage systems, networks, and data distribution. There has been some work towards better data locality, which seems to be a promising approach to mitigate this problem [5].

For example, cache systems can greatly benefit from the knowledge about the data file popularity. Studies in this direction have been in place for some of the experiments using the log data available to them [6]. Therefore, given the detailed access data at our disposal, we plan to expand this research beyond the scope of one experiment and look at this problem in the context of the EOS Storage System.

References

1. Bauerdick L, Bloom K., Bockelman B., Bradley D., Dasu S., Dost J., Sfiligoi I., Tadel A., Tadel M., Wuerthwein F. et al.: XRootd, disk-based, caching proxy for optimization of data access, data placement and data replication (2014), Vol. 513
2. Tadel M., Tadel A.: XRootD Proxy File Cache V2 (2016), XRootD Workshop @ ICEPP.
3. Aimar A. , Aguado Corman A., Andrade P., Delgado Fernandez J., Garrido Bear B., Karavakis E., Kulikowski D.M., Magnoni L.: MONIT: Monitoring the CERN Data Centres and the WLCG Infrastructure. EPJ Web of Conferences. 214. 08031. 10.1051/epjconf/201921408031 (2019)
4. Report Log Files. EOS CITRINE Documentation. http://eos-docs.web.cern.ch/eos-docs/using/reports.html.
5. Flix J., Delgado Peris A., Hernández J.M., Pérez Dengra C., Pérez-Calero A., Planas E., Rodriguez Calonge F. J., Sikora A.: CMS data access and usage studies at PIC Tier-1 and CIEMAT Tier-2. CHEP (2019)
6. Meoni M., Perego R., Tonellotto N.: Dataset Popularity Prediction for Caching of CMS Big Data. Journal of Grid Computing. 16. 10.1007/s10723-018-9436-4 (2019)