Understanding Data Access Patterns for dCache System

Julian Bellavita[1,*], *Caitlin* Sim[1,**], *Kesheng* Wu[2,***], *Alex* Sim[2,****], *Shinjae* Yoo[3,†], *Hiro* Ito[3,‡], *Vincent* Garonne[3,§], and *Eric* Lancon[4,¶]

[1]University of California at Berkeley, USA
[2]Lawrence Berkeley National Laboratory, USA
[3]Brookhaven National Laboratory, USA

Abstract. The storage management system dCache acts as a disk cache for high-energy physics (HEP) data from the US ATLAS community. Since its disk capacity is considerably smaller than the total volume of ATLAS data, a heuristic is needed to determine what data should be kept on disks. An effective heuristic would be to keep the data files that are expected to be heavily accessed in the near future. Through a careful study of access statistics, we find a few most popular datasets are accessed much more frequently than others, even though these popular datasets change over time. If we could predict the near-term popularity of datasets, we could pin the most popular ones in the disk cache to prevent their accidental removal and guarantee their availability. To predict a dataset popularity, we present several methods for forecasting the number of times a dataset will be accessed in the next day. Test results show that these methods could predict the next-day access counts of popular datasets reliably. This observation is confirmed with dCache logs from two separate time ranges.

1 Introduction

The storage management system dCache at Brookhaven National Laboratory (BNL) is the disk cache for a large collection of high-energy physics (HEP) data, primarily from the A Toroidal LHC ApparatuS (ATLAS) experiment [1, 2]. Storage space on dCache is much smaller than the full ATLAS data collection residing on tape archives and distributed data centers. Therefore, a policy is needed to determine what data files to keep in the cache and what files to evict. A good policy would keep files that will be frequently accessed *in the near future*. In this work, we use the past file access information to predict the number of times that files are accessed in the next few days. The analysis tasks from the ATLAS community often access predefined groups called datasets [3]. Therefore, this study predicts how many times a dataset will be accessed in the future rather than the popularity of each individual file.

*e-mail: jbellavita@berkeley.edu
**e-mail: caitlinsim@berkeley.edu
***e-mail: kwu@lbl.gov
****e-mail: asim@lbl.gov
†e-mail: sjyoo@bnl.gov
‡e-mail: hito@rcf.rhic.bnl.gov
§e-mail: vgaronne@cern.ch
¶e-mail: elancon@bnl.gov

To understand the access patterns of these datasets, we study the summary statistics of the access logs from dCache server and explore the intrinsic structure through unsupervised learning techniques such as K-means clustering. We find that the majority of datasets are accessed relatively small number of times, while there are a small number of highly popular datasets accessed many thousands of times per day. Based on this observation on access statistics, pinning the popular datasets on disk could guarantee their availability without negative impact on the overall disk cache operations.

To predict the popularity of datasets, we present a neural network that can forecast the anticipated access count of each dataset next day [4]. The neural network was trained using the information about dCache transactions from Oct. 2020 to Apr. 2021. We process the raw dCache logs into daily access statistics with the next day's access count as the target variable for learning. We find that the neural network is able to accurately forecast dataset popularity for the most popular datasets in a time range. The same model could to applied to another time range (Jan. 2022 to Jan. 2023) to capture the relative popularity of the datasets. This suggests that the popularity of a dataset is predictable and therefore could be used to advise the dCache system operators to pin the popular ones.

This paper is organized as follows. Section 2 provides general background information on the dCache system, and also summarizes related work. Section 3 provides a general description of the data we analyzed in the study. Sections 4 and 5 presents the results of our analysis, and Section 6 provides closing remarks.

2 Background and Related Work

dCache is a hierarchical storage system that uses a combination of disk and tape storage to provide both high-speed access and long-term archiving of data. It uses the High-Performance Storage System (HPSS) [5] tape-based storage system as its backend at BNL. Data is stored persistently via HPSS, and recently accessed data is placed in a disk cache for faster access. In this work, we examine a dCache installation at BNL that is responsible for storing data produced by the ATLAS experiment at CERN as a WLCG Tier-1 storage site.

In [6], a novel approach to optimizing the performance of web map caching systems using a neural network is presented. The authors propose the use of an adaptive neural network method for making intelligent decisions regarding tile replacement within the cache. This method aims to enhance the efficiency of web map services by dynamically predicting which map tiles are more likely to be requested in the near future and prioritizing their retention in the cache. In [7], the authors propose a caching policy based on a feedforward neural network that predicts future content popularity. Finally, in [8], the authors study the use of neural networks in computer hardware, with a focus on developing a cache prefetching policy based on memory access patterns forecasted by a neural network.

3 dCache Logs

All of the data analyzed in this work was obtained from dCache server logs at BNL. This section describes the structure and contents of these logs, and it also provides a general overview of how data was extracted from the dCache logs.

3.1 dCache Log Description

dCache server logs contain information about the transactions handled by the dCache system on a particular day. There are three types of dCache logs: Billing Logs, DSN Logs, and Door

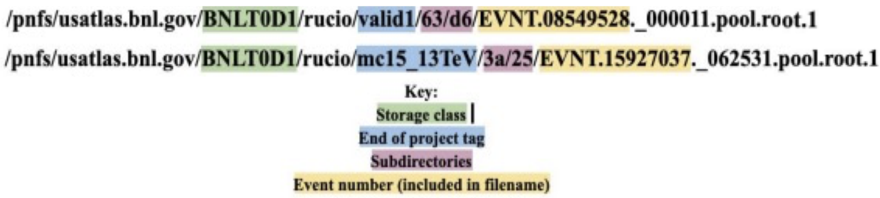


Figure 1: Two examples of dCache filepaths to demonstrate how TIDs are identified. The TID of each file is highlighted in yellow.

Logs. Door logs and Billing logs contain information about dCache transactions, such as the type of the transaction (read, write, etc.), the number of bytes transferred by the transaction, and most importantly, the path of the file that the transaction is directed towards. DSN logs contain mappings between filepaths and datasets.

3.2 TID Grouping

HEP collaborations like ATLAS generate files in groups known as datasets, and each of these groups (datasets) is produced by a *task* (such as a physical experiment and a simulation) with a Task ID, or TID. Each file in dCache has the unique TID of its corresponding dataset in its path. Two example dCache filepaths with their TIDs highlighted in yellow are presented in Figure 1. By analyzing filepaths, it is possible to group files into datasets via their TID. In this analysis, we examine the popularity of individual datasets instead of individual files. The dCache system operators are considering policies specified in TIDs rather than individual files. This is because it is common for all files in a dataset to be accessed together. Therefore, if a dataset with a specific TID is expected to be very popular in the next few days, it might make sense to pin all files of the dataset in disk.

3.3 Data Analysis Pipeline

In order to turn the raw data in the dCache logs into a usable format, we built a pipeline that extracts relevant information from dCache logs and creates a Pandas Dataframe that is suitable for further analysis, henceforth referred to as the "Master Dataframe". Each row in the Master Dataframe contains information about a single TID on a single day. This means that if the same TID is accessed on two different days, there will be two separate rows, each containing information about that TID on a different day. The Master Dataframe contains the following information about each TID: the total size of the TID in bytes, the number of files in the TID, the number of bytes read from the TID on a single day, and the number of times the TID is accessed on a single day.

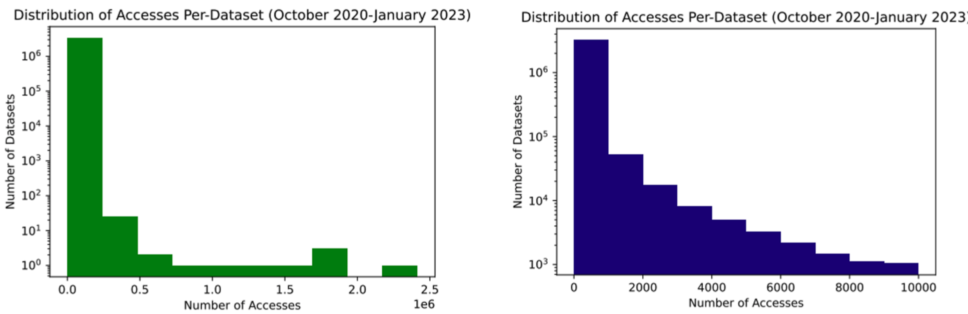
3.4 Timeframes Studied

Our analysis was conducted upon two separate time ranges. The first time range encompasses October 2020 to April 2021, and the second time range encompasses January 2022 to January 2023. We chose to analyze two separate date ranges in order to verify that the patterns we observed in one date range would appear again in another date range.

https://doi.org/10.1051/epjconf/202429501053

Table 1: Total numbers of unique TIDs accessed and total number of accesses across all TIDs for both date ranges.

	10/2020 – 4/2021	1/2022 – 1/2023	**TOTAL**
TIDs Accessed	576,577	677,215	1,166,312
Total Accesses	247,598,874	302,615,187	550,214,061



(a) (b)

Figure 2: (a) Distribution of the number of accesses per dataset for both date ranges concatenated. There is a right skew, and most datasets are accessed infrequently. (b) Zoomed-in of the largest bin

4 Summary Statistics and K-Means Clustering

In order to get an understanding of how dCache datasets tend to be accessed, we performed an exploratory analysis of dCache accesses. By analyzing the Master Dataframe described in Section 3.3, it is easy to get the information needed for this analysis.

4.1 Summary Statistics

Table 1 shows summary statistics for both date ranges. The first row shows the total number of unique TIDs (or datasets) accessed in each date range. The second row shows the total number of times dCache datasets are accessed in each date range.

Figure 2a shows the distribution of total accesses per dataset, and Figure 2b is a zoomed-in histogram of the largest bin from Figure 2a. For these figures, we looked at both date ranges concatenated. In both plots, it is clear that the majority of datasets are accessed relatively few times, and there is a small group of datasets that are accessed a large number of times. This small group of datasets is the group that we want to identify ahead of time and pin in dCache, since they are accessed frequently. We also examined the total number of dCache accesses per month, for each month in both date ranges. This information is summarized in Figure 3.

4.2 K-Means Clustering

Next, we wanted to explore the relationship between present and future numbers of accesses. We performed K-Means clustering on both date ranges, clustering the datasets according to their present and next day access counts. To determine the best value for the K parameter, we used the elbow method on values of K from 1 to 29. These results are summarized in Figure 4. Based on this plot, the optimal value for K is 4.

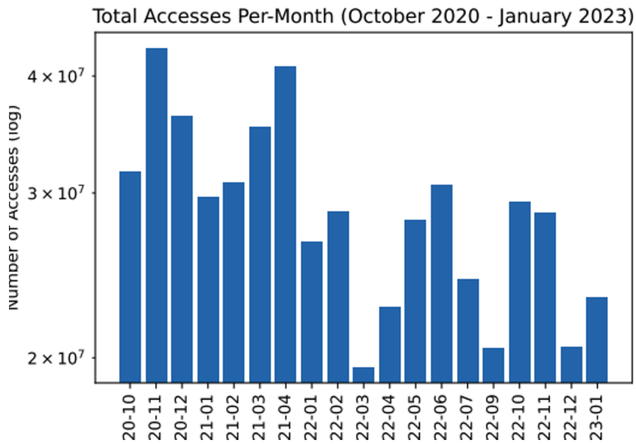


Figure 3: Total number of dCache accesses for each month in both of the date ranges.

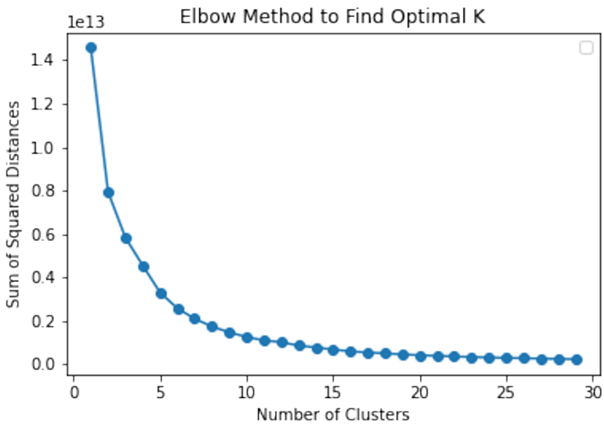


Figure 4: Elbow method to determine optimal K value.

Figure 5 shows the results of K-Means clustering with $K = 4$ for the first date range. The cluster corresponding to datasets with less than 10^4 accesses is extremely large, whereas the clusters corresponding to higher numbers of accesses are small. This indicates that the majority of datasets are accessed relatively few times, and that there is also a small number of highly popular datasets. Pinning the small group of very popular datasets in dCache would achieve our goal of a popularity-based caching policy.

To verify the pattern we observed for the first date range (from Oct. 2020 to Apr. 2021), we performed K-Means clustering with $K = 4$ on the second date range (from Jan. 2022 to Jan. 2023). Figure 6 shows the results of this second round of K-Means clustering. As with the first date range, we can identify a small group of frequently accessed datasets, and a large group of infrequently accessed datasets. There is a gap in the large purple group that is not present in the plot for the first date range, but there is still a clear divide between the two groups. Overall, this indicates that it is common for there to be a small group of popular datasets. Next, we describe our approach to forecast which datasets will fall into this group.

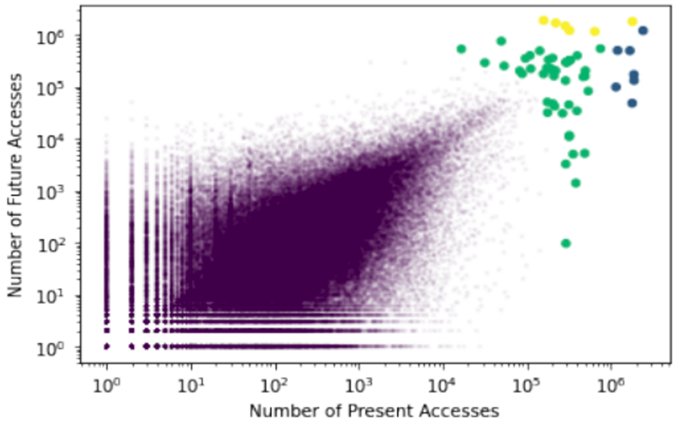


Figure 5: K-means clustering with $K = 4$. Data is from August 2020-March 2021. A small number of datasets are accessed much more frequently than others and their access counts might be predictable.

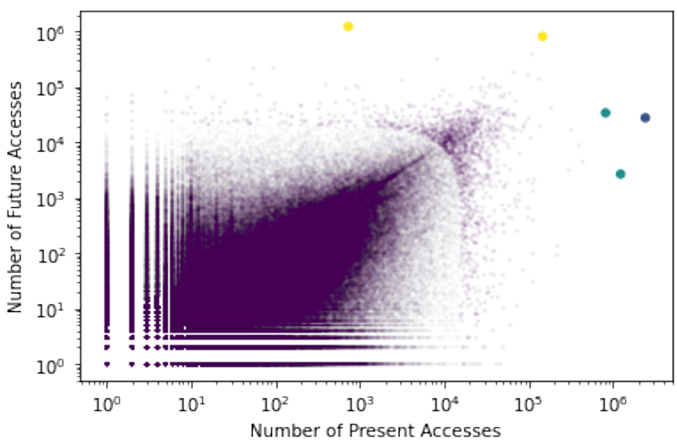


Figure 6: K-means clustering with k=4. Data is from January 2022-January 2023. As with the previous date range, there is a small group of very popular datasets.

5 Prediction Model

In order to predict dataset popularity, we built a neural network that is capable of forecasting how many times a dataset will be accessed. The neural network was trained using data from the first date range from Oct. 2020 to Apr. 2021, and validated on both date ranges (one from Oct. 2020 to Apr. 2021 and another from Jan. 2022 to Jan. 2023). Each record in the data contains information regarding a particular dCache dataset on a particular day. The data contains the following features: the number of files in a dataset, the number of times a dataset is accessed on the date corresponding to the record, the size of a dataset in bytes, the number of bytes read from a dataset, the number of times a dataset is accessed one day after the date corresponding to the record (this is the feature forecast by the deep neural network), and the type of files found in the dataset (e.g. experiment results, server log data, etc.). The deep neural network was built using PyTorch; it uses 2 dense layers, the *tanh* activation function, and the ADAM optimizer [9].

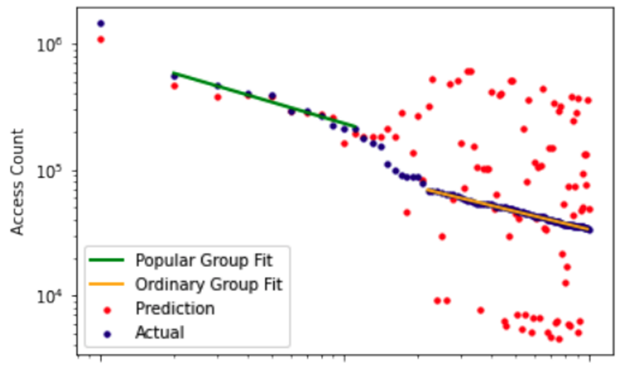


Figure 7: Predicted vs. actual access counts for August 2020-March 2021. The 100 most popular datasets sorted according to their actual access counts.

Figure 7 shows the predicted access values vs the actual access values for the 100 most popular datasets in the first date range from Oct. 2020 to Apr. 2021. The most popular dataset is accessed much more than the second most popular dataset, while the access counts of the next ten most popular datasets follow a power law with the exponent of −0.57. The access counts of many commonly accessed datasets follow the same power law show in Figure 7 for the majority of the top 100 popular datasets. This power law has an exponent of −0.47. This corroborates the pattern shown in Figure 5, where there is a small group of highly popular datasets, and their accesses are more predictable.

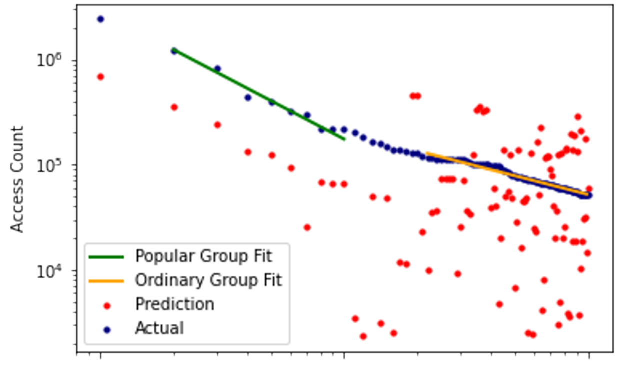


Figure 8: Predicted vs. actual access counts for January 2022-January 2023. The 100 most popular datasets sorted according to their actual access counts.

To further validate the prediction model, we used it to predict the number of accesses for the 100 most popular datasets in the second date range from Jan. 2022 to Jan. 2023. Figure 8 shows these results. As with the first date range, the model is able to predict access values for the small group of highly popular datasets, although it is inaccurate for less popular datasets, and its predictions for popular datasets are less accurate than with the first date range. However, it should be noted that the predicted values for the popular datasets appear to differ from the actual values by a constant factor, meaning the model is still able to make reasonably accurate forecasts. The access counts of the 2nd through 11th most popular datasets follow a power law with an exponent of -1.27, and the access counts for the majority of the remaining top 100 datasets follow a power law with an exponent of -0.59. For both date ranges, the model is able to identify which datasets fall into the small group of very popular datasets. Therefore, the model can be used to select which datasets should be pinned in dCache.

6 Conclusion

By analyzing logs from two separate date ranges, we identified the existence of a clear access pattern on ATLAS disk cache. The majority of datasets are accessed a relatively small number times a day, and there are a few datasets that are accessed many more times. Pinning the popular datasets in dCache will lead to a more effective cache replacement policy for dCache, since it will guarantee the popular data is available from the disk cache. To understand the predictability of access counts, we showed a neural network could predict the access counts of datasets. Tests show that the predictions are more accurate for the popular datasets, which means the predictions are useful for advising dCache operators on what datasets to pin. The popular datasets appear to follow a different power-law distribution in their popularity than the remaining datasets, we are interested in understanding this better better. Furthermore, we are also interested in exploring different prediction algorithms to improve the prediction accuracy to make the pinning advice more reliable. We are also exploring simulations to evaluate the effectiveness of different caching policies.

7 Acknowledgments

This work was supported by the Office of Advanced Scientific Computing Research, Office of Science, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231, and also used resources of the National Energy Research Scientific Computing Center (NERSC).

References

[1] G. Behrmann, P. Fuhrmann, M. Grønager, J. Kleist, *A distributed storage system with dCache*, Journal of Physics: Conference Series **119** (2008)

[2] M. Ernst, P. Fuhrmann, M. Gasthuber, T. Mkrtchyan, C. Waldman, *dcache, a distributed storage data caching system*, Journal of Physics: Conference Series (2001)

[3] Y. Wang, K. Wu, A. Sim, S. Yoo, S. Misawa, *Access Patterns to Disk Cache for Large Scientific Archive*, in *ACM International Workshop on Systems and Network Telemetry and Analytics* (2020), pp. 37–40

[4] P.K. Patra, M. Sahu, S. Mohapatra, R.K. Samantray, *File access prediction using neural networks*, IEEE Transactions on Neural Networks **21**, 869 (2010)

[5] R.W. Watson, R.A. Coyne, *The Parallel I/O Architecture of the High-Performance Storage System (HPSS)*, in *Proceedings of the 14th IEEE Symposium on Mass Storage Systems* (1995), ISBN 0818670649

[6] R. García, J.P. de Castro, M.J. Verdú, E. Verdú, L.M. Regueras, P. López, *An Adaptive Neural Network-Based Method for Tile Replacement in a Web Map Cache*, in *International Conference on Computational Science and Its Applications (ICCSA)* (2011), pp. 76–91

[7] V. Fedchenko, G. Neglia, B. Ribeiro, *Feedforward neural networks for caching: N enough or too much?*, ACM SIGMETRICS performance evaluation review **46**, 139 (2019)

[8] M. Hashemi, K. Swersky, J.A. Smith, G. Ayers, H. Litz, J. Chang, C. Kozyrakis, P. Ranganathan, *Learning memory access patterns*, CoRR (2018), 1803.02329

[9] D.P. Kingma, J. Ba, *Adam: A method for stochastic optimization*, arXiv (2017), 1412.6980