

Understanding Data Access Patterns for dCache System

Julian Bellavita¹, Caitlin Sim¹, Kesheng Wu², Alex Sim²
Shinjae Yoo³, Hiro Ito³, Vincent Garonne³, Eric Lancon³

University of California, Berkeley¹, Lawrence Berkeley National Laboratory²
Brookhaven National Laboratory³

{jbellavita,caitlinsim}@berkeley.edu¹, {kwu,asim}@lbl.gov², {sjyoo,hito,vgaronne1,elancon}@bnl.gov³

At Brookhaven National Lab, the dCache storage management system is used as a disk cache for large high-energy physics (HEP) datasets primarily from the ATLAS experiment[1]. Storage space on dCache is considerably smaller than the full ATLAS data collection. Therefore, a policy is needed to determine what data files to keep in the cache and what files to evict. A good policy is to keep frequently needed files *in the future*. In this work, we use the current and past file access information to predict the number of file accesses in the next day. The analysis tasks from the ATLAS experiment often access a predefined dataset as a group. Therefore, this study predicts how many times a dataset will be accessed in the future rather than each individual file.

HEP collaborations like ATLAS generate files in groups known as datasets and each of these groups (datasets) is produced by a *task* (such as an physical experiment and a simulation) with a Task ID, or TID. The dCache system operators are considering policies specified in TIDs rather than individual files. For example, if a dataset (with a specific TID) is expected to be very popular in the next few days, it might make sense to ping all files of the dataset in disk.

To investigate how datasets tend to be accessed, we first performed K-means clustering on 9 months' worth of dCache operational logs. Figure 1 shows the results of clustering the datasets according to their present and next day access counts. 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. Pinging the small group of very popular datasets in dCache would achieve our goal of a popularity-based cache policy.

The neural network was trained using a dataset containing information for 9 months' worth of dCache transactions. We process the raw dCache logs into daily access statistics with the next day's access count as the target variable for learning. The neural network was built using PyTorch; it uses 2 dense layers, the Tanh activation function, and the ADAM optimizer.

Figure 2 shows the predicted access values vs the actual access values for the 100 most popular datasets. 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

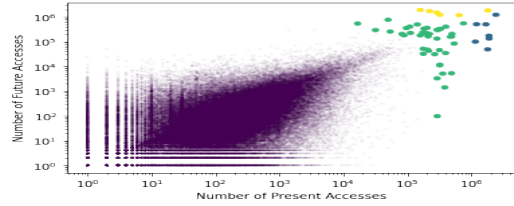


Fig. 1. K-means clustering with $k=4$. A small number of datasets are accessed much more frequently than others and their access counts might be predictable.

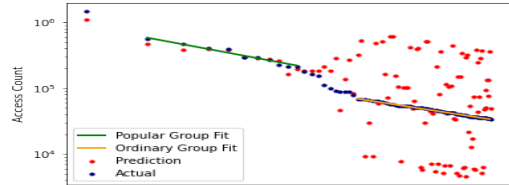


Fig. 2. Predicted vs. actual access counts in the next day. The 100 most popular datasets sorted according to their actual access counts.

-0.57, and an RMSE of 5.7×10^5 . The access counts of many commonly accessed datasets follow the same power law show in Figure 2 for the majority of the top 100 popular datasets. This power law has an exponent of -0.47, and an RMSE of 2.0×10^6 . This corroborates the pattern shown in Figure 1, where there is a small group of highly popular datasets, and their accesses are more predictable. Since the most popular few datasets are accessed much more frequently than others, pinging them in the disk cache could simplify the cache replacement decisions without sacrificing the overall disk cache effectiveness.

In summary, our results show that the popularity of the most popular datasets are predictable. It is therefore possible to ping these datasets in dCache, yielding a more effective cache policy. Future work will develop, simulate, and benchmark cache policies based off of the method presented here.

REFERENCES

- [1] Y. Wang, K. Wu, A. Sim, S. Yoo, and S. Misawa, "Access patterns to disk cache for large scientific archive," in *Proceedings of the 2021 on Systems and Network Telemetry and Analytics*, 2020, pp. 37–40.