Activeness-based Data Retention Recommender for HPC Facilities

Wei Zhang
X-Spirit.zhang@ttu.edu
Texas Tech University
Lubbock, Texas

KEYWORDS

data retention, recommender system, high performance computing

1 INTRODUCTION

Data retention policies at HPC facilities play a vital role in retaining valuable files and maintaining sufficient storage space. In addition to the classical time-based data retention solution [6], many data retention solutions were proposed over the years [2, 8, 9], but none of them were put into practical use due to the significant complexity in the retention criteria or the deployment. The current in-practice data retention policies only focus on the temporal properties of files [1, 3–5, 7] and hence may cause unnecessary data access interruption or even data loss to users. In this study, we propose an *activeness-based data retention action recommender (ActiveDR)* that generates user-centric data retention recommendations based on a holistic view of the user activeness. It is the first data retention solution that promotes the active and fruitful use of the HPC facilities. Our evaluation on the recommendation results shows that, as compared to the data retention policies in practice, adopting ActiveDR's data retention recommendation can significantly avoid undesired data loss for active users while reaching the same purge target. Also, we show that ActiveDR is a resource efficient data retention solution.

2 METHODOLOGY

As shown in Figure 1, ActiveDR considers the activeness of users of an HPC facility at the center of its design, including the operations that the users perform on the HPC system and the outcome that the users produce from the operations performed on the HPC system.

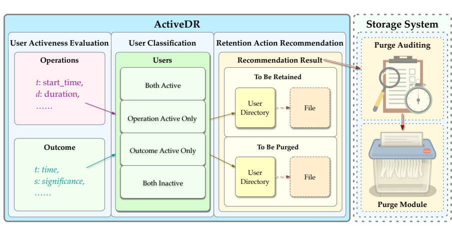


Figure 1: Overview of ActiveDR Design

Following the definition of some important metrics in our user activeness evaluation framework shown in Table 1, we first build a time-series activeness rank vector as shown in Figure 2.

Table 1: List of Mathematical Notations in User Activeness Evaluation Framework

Meaning	Notation
Set of m Periods	$P = \{p_0, ..., p_{m-1}\}$
Set of n Activity Types	$T = \{\lambda_0, ..., \lambda_{n-1}\}$
Set of k Activities	$A = \{a_0, ..., a_{k-1}\}$
Activeness of Activity a_t at time t	D_{a_t}
Average Activeness of All Activities A in each period	$D_{\overline{A}}$ or $Avg(D_A)$
Activeness Ratio of a Certain Period p	$b_p = D_{A_p}/D_{\overline{A}}$
Period Index of Activity a_t	$e = m - (a_t.ts - a_0.ts)/l + 1$ (ts denotes timestamp)

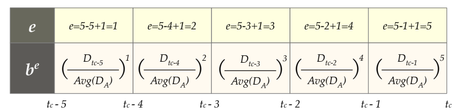


Figure 2: Time-series Activeness Rank Vector when $m_\lambda = 5$.

With the activeness rank vector, we can calculate the overall activeness rank of a particular activity type λ:

$$\Phi_\lambda = \prod_{e=1}^{m} (b_{p_e})^e \tag{1}$$

For other types of activities that can be categorized as operations or outcome, we perform the following calculation to derive the overall operation activeness Φ_{op} and the overall outcome activeness Φ_{oc}:

$$\Phi_{op} = \prod_{\lambda_{op}=1}^{m_{op}} \Phi_{\lambda_{op}} \quad and \quad \Phi_{oc} = \prod_{\lambda_{oc}=1}^{m_{oc}} \Phi_{\lambda_{oc}} \tag{2}$$

where λ_{op} denotes any operation activity and λ_{oc} denotes any outcome activity. Note that both Φ_{op} and Φ_{oc} will be within either the range $[0, 1)$ or the range $[1, +\infty)$.

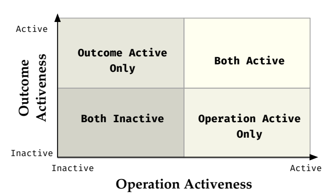


Figure 3: User Classification Matrix

Once the operation activeness and outcome activeness are evaluated, we can formulate a 2D matrix to classify all users into four categories (as depicted in Figure 3) and sort these users according to their activeness scores in each category. When sorting these users, we treat the operation activeness score Φ_{op} with higher priority.

ActiveDR generates recommendation on every d days. The lifetime ε_f of file f owned by this user can be calculated by the following equation:

$$\varepsilon_f = d \times \Phi_{op} \times \Phi_{oc} \quad (3)$$

where Φ_{op} and Φ_{oc} can be reset to 1.0 if they are less than 1.0. If a user is inactive for both the operations and the outcome, the lifetime ε_f of file f owned by this user is $\varepsilon_f = d$. When ActiveDR generating recommendation at time t_c, if the access time of file f is denoted as $atime_f$, as long as $t_c - atime_f > \varepsilon_f$, ActiveDR will generate a file purge action in the recommendation result for file f.

3 EVALUATION

We conducted our evaluation on the Cori Haswell computing nodes at NERSC. Our prototype implementation of ActiveDR can be found at [10].

To evaluate ActiveDR, we perform simulations of both ActiveDR and the in-practice fixed-lifetime data retention policy (a.k.a FLT).

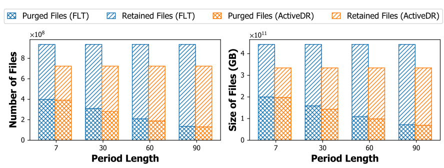


Figure 4: Overall Data Retention Effect

Figure 4 shows that both solutions lead to the same number of purged files and the same amount of data being purged. However, the number of retained files and the total size of them for ActiveDR is less than those of FLT.

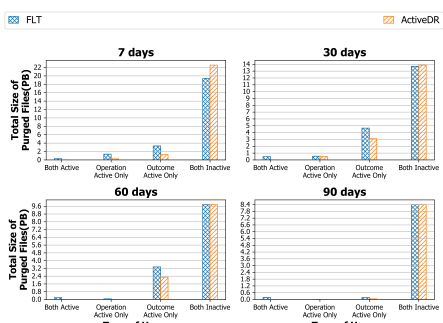


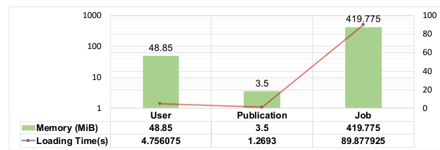
Figure 5: Total Size of Purged Files for the Users of Various Activeness Categories

Figure 5 demonstrates that ActiveDR purges fewer files for all active users and purges more files for "both inactive" users as compared to "FLT". ActiveDR does not lose the ability to purge files for inactive users, and it even performs better than the FLT does.

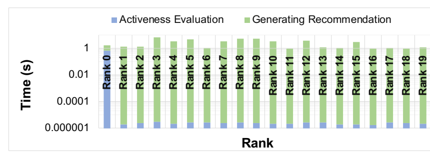
As shown in Figure 6, by adopting the data retention recommendation of ActiveDR, the number of users affected by data purge actions in all three active user groups is much smaller than that of the FLT approach.

Figure 7a, shows that ActiveDR only require about 500MB memory to execute and it only took less than 2 minute to load all user

Figure 6: Number of Users Affected by File Purge



(a) Memory Consumption during Activeness Evaluation



(b) Time for Activeness Evaluation and Recommendation Generating

Figure 7: Performance Evaluation

activities collected within 2 years, which is considered very efficient for HPC environment. Figure 7b shows that the activeness evaluation and the actual recommendation generating process is blazingly fast, which only took less than one second.

REFERENCES

[1] High Performance Computing Center at Texas Tech University. 2020. Data Policy - Use of HPCC Storage. https://www.depts.ttu.edu/hpcc/operations/datapolicy.php.

[2] Henry M. Monti, Ali R. Butt, and Sudharshan S. Vazhkudai. 2009. /Scratch as a Cache: Rethinking HPC Center Scratch Storage. In *Proceedings of the 23rd International Conference on Supercomputing* (Yorktown Heights, NY, USA) *(ICS '09)*. Association for Computing Machinery, New York, NY, USA, 350–359. https://doi.org/10.1145/1542275.1542325

[3] National Center for Atmospheric Research. 2020. GLADE File Space. https://www2.cisl.ucar.edu/resources/storage-and-file-systems/glade-file-spaces.

[4] National Energy Research Scientific Computing Center. 2020. NERSC Data Management Policy. https://docs.nersc.gov/data/policy/.

[5] Oak Ridge Leadership Computing Facility. 2020. OLCF Policy Guides - Data Management Policy. https://docs.olcf.ornl.gov/accounts/olcf_policy_guide.html.

[6] Arie Shoshani, Alexander Sim, and Junmin Gu. 2004. Storage Resource Managers. In *Grid Resource Management*. Springer, 321–340.

[7] Texas Advanced Computing Center. 2020. TACC Usage Policy. https://portal.tacc.utexas.edu/tacc-usage-policy.

[8] Lars Arne Turczyk, Oliver Heckmann, and Ralf Steinmetz. 2007. File Valuation in Information Lifecycle Management. In *Proceedings of the Thirteenth Americas Conference on Information Systems, Keystone, Colorado.*

[9] Fons Wijnhoven, Chintan Amrit, and Pim Dietz. 2014. Value-Based File Retention: File Attributes as File Value and Information Waste Indicators. *J. Data and Information Quality* 4, 4, Article 15 (May 2014), 17 pages. https://doi.org/10.1145/2567656

[10] Wei Zhang. 2020. ActiveDR (v1.0.2). https://doi.org/10.5281/zenodo.3878756