A Dynamic Multi-Tiered Storage System for Extreme Scale Computing

Hariharan Devarajan, Anthony Kougkas, and Xian-He Sun
Illinois Institute of Technology
hdevarajan@hawk.iit.edu, akougkas,sun@iit.edu

I. Extended Abstract

The traditional compute-centric architecture of high-performance computing (HPC) systems has led to the separation of compute and storage nodes, which are generally interconnected via a shared network infrastructure. In this architecture, applications make data access a secondary concern. However, with science moving towards data-driven discoveries in various fields such as astronomy, high-energy physics, climate modeling, and earthquake engineering, applications require immense support from the storage sub-system to perform their tasks and have been shown to spend a significant amount of time performing I/O operations[1]. As we move towards the exascale era, current storage architectures cannot easily scale to handle the extreme computing requirements [2]. Therefore, the I/O bottleneck problem presents a significant challenge for scientific applications on modern supercomputers that must be solved by the next generation of storage systems.

In the age of data-driven discoveries, the diversity of modern applications ranging from the simulation of natural phenomena to artificial intelligence involved in many aspects of our day-to-day lives has seen staggering growth. These applications come in various shapes and sizes, ranging from measuring heart rate with a small IoT device to running a weather simulation on the largest supercomputer on the planet. They exhibit a diverse set of I/O requirements, both hardware and software, which are vital to their accuracy and performance. However, this diversity puts tremendous stress on existing storage sub-systems, as they have to manage a plethora of applications that often have conflicting I/O requirements. Moreover, most applications are executed as a part of a long-running workflow process, which further exacerbates the problem, since most storage systems cannot adapt to the changing requirements stemming from different applications. Therefore, a storage system that can dynamically match and adapt to the changing application requirements is necessary.

Domain-specific applications in both scientific and cloud computing demonstrate a spectrum of conflicting I/O requirements that puts additional stress on storage subs-systems. These communities have developed a plethora of storage solutions that have significantly diverged from one another in various ways. Additionally, there is extensive growth and development of new I/O technologies that have defined new standards of I/O performance for applications. This growth has led to many heterogeneous storage architectures where different devices are presented as storage accelerators or additional capacity to support a diverse set of application requirements. As workflows converge, with both scientific and big data applications working together in increasing frequency, the underlying storage system is not capable of simultaneously handling the conflicting I/O requirements from these different classes of applications. Hence, a storage system that can transparently manage the heterogeneity of the underlying storage is required.

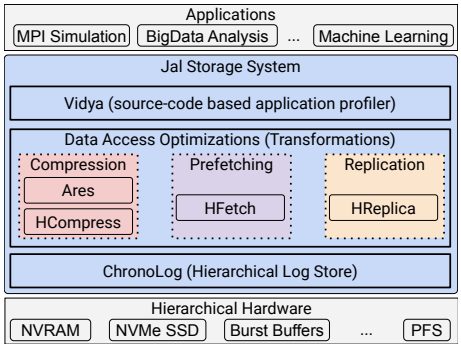

Fig. 1. Jal high-level design.

In order to support the growing diversity of I/O requirements and the heterogeneity of storage architectures, a new type of a storage system must be designed to achieve a holistic solution to the I/O bottleneck problem. We propose that *dynamic storage re-configuration and malleability* should be introduced to the next-generation of I/O systems. There are several challenges associated to the design and implementation of a dynamically re-configurable storage system:

1) How to understand and characterize the cause of application I/O behavior?
2) How to match diverse application requirements with storage configurations?
3) How to design a dynamically re-configurable multi-tiered storage system?

A. Jal Storage System

We propose Jal, a novel, dynamically re-configurable, multi-tiered storage system that can mitigate the challenges mentioned above. Jal targets high-performance computing (HPC) and extreme-scale environments. We present a layered approach to the Jal storage system, where each layer is responsible for mitigating a challenge. We use three layers as shown in Figure 1:

- *An abstract application model*: defines an automated methodology to extract an application's I/O requirements by understanding its code structure. In order to understand

the cause of the application's I/O behavior, the model matches the observed I/O behavior directly to a specific code block. To achieve this, Jal uses a combination of compiler-driven and machine learning techniques.

- *An abstract data model*: explores how different I/O requirements from the application can be translated into storage system requirements. In this layer, Jal analyzes various data access optimizations (e.g., data prefetching, compression, and replication) to understand how, for each case, an application's I/O requirements can be mapped to the underlying storage sub-system.
- *An abstract storage model*: aims to provide a re-configurable storage abstraction that can unify and manage several heterogeneous storage resources and tune them based on the characteristics required from the above layer.

The ultimate goal of this work is to develop a new, *high-performance, dynamically re-configurable, and multi-tiered storage solution*.

To achieve the above goal, we need to first understand, characterize, and tune applications' I/O behaviors. Existing tools use either offline profiling or online analysis to get insights into an application's I/O patterns. However, there is a lack of a clear metric to characterize an application's I/O. Moreover, these tools are application-specific and do not account for multi-tenant systems. In this work, we present Vidya [3], an I/O profiling framework that can predict an application's I/O intensity using a new formula called Code-Block I/O Characterization (CIOC). Using CIOC, developers and system architects can tune an application's I/O behavior and better match the underlying storage system to maximize performance. Evaluation results (shown in poster Figure 6) show that this approach can predict an application's I/O intensity with a variance of **0.05%**. Additionally, this approach can profile applications with a high accuracy of **98%** while reducing profiling time by **9x** compared to existing tools. We further show how this approach can drive I/O optimizations boosting applications' performance by up to **3.7x**.

Additionally, modern HPC storage solutions include fast node-local and/or globally shared storage resources to alleviate data access bottleneck for applications. Also, several middleware libraries (e.g., Hermes[4]) are proposed to transparently move data between these storage tiers. Furthermore, scientists have proposed various data access optimizations in the form of Data compression, Data Prefetching, and Data Replication. These optimizations and hardware technologies, if used together, can benefit from each other. The effectiveness of data access optimization can be enhanced by selecting different algorithms according to the characteristics of the different storage tiers, and the multi-tiered storage hierarchy can benefit from transparent data access optimization. To this end, we designed and implemented various data access optimizations [5], [6], [7] that can improve an application's performance by harmoniously leveraging both multi-tiered storage and data access optimization. We have developed various algorithms that facilitate the optimal matching of data optimization algorithms to the tiered storage. Our evaluation (as in poster Figure 8) shows that our optimization can improve the performance of scientific applications by **7x** when compared to other state-of-the-art compression frameworks.

Modern applications, spanning from Edge to Cloud to High-Performance Computing (HPC), produce/process log data and create a plethora of workload characteristics that rely on a common storage model: the distributed shared log. Applications such as key-value stores, message brokers, metadata, coordination, and file system namespace services showcase the power of a shared log abstraction and how it can facilitate several application requirements efficiently. However, many of these applications' log requirements are often conflicting with one another. Furthermore, these applications run concurrently, making it harder to satisfy these requirements under the same system. In this work, we present the design and implementation of ChronoLog [8], a new distributed shared log store that uses physical time to provide total ordering on a log. ChronoLog also offers the ability to process the log with partial reads via range queries. It is designed to offer high performance via I/O isolation (tail and historical operations are handled separately), elastic storage capabilities, and a new 3D log distribution. Evaluation results (in poster Figure 7) show that eliminating a centralized synchronization point can boost performance to new highs. ChronoLog can achieve millions of tail operations per second and can outperform existing log stores by an order of magnitude.

The Jal storage system utilizes all of these components to form a dynamic, re-configurable, multi-tiered, storage system that can achieve optimal matching between application requirements and diverse storage technologies.

REFERENCES

[1] K. Wang, A. Kulkarni, M. Lang, D. Arnold, and I. Raicu, "Using simulation to explore distributed key-value stores for extreme-scale system services," in *SC'13: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. IEEE, 2013, pp. 1–12.

[2] D. Zhao, D. Zhang, K. Wang, and I. Raicu, "Exploring reliability of exascale systems through simulations." in *SpringSim (HPC)*, 2013, p. 1.

[3] H. Devarajan, A. Kougkas, P. Challa, and X.-H. Sun, "Vidya: Performing Code-Block I/O Characterization for Data Access Optimization," in *2018 IEEE 25th International Conference on High Performance Computing (HiPC)*, Dec 2018, pp. 255–264.

[4] A. Kougkas, H. Devarajan, and X.-H. Sun, "Hermes: a heterogeneous-aware multi-tiered distributed I/O buffering system," in *Proceedings of the 27th International Symposium on High-Performance Parallel and Distributed Computing*. USA: ACM, 2018, pp. 219–230.

[5] H. Devarajan, A. Kougkas, L. Logan, and X.-H. Sun, "Hcompress: Hierarchical data compression for multi-tiered storage environments," in *2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, May 2020.

[6] H. Devarajan, A. Kougkas, and X.-H. Sun, "An intelligent, adaptive, and flexible data compression framework," 2019.

[7] H. Devarajan, A. Kougkas, , and X.-H. Sun, "HFetch: Hierarchical Data Prefetching for Scientific Workflows in Multi-Tiered Storage Environments," in *2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, May 2020, pp. 62–72.

[8] A. Kougkas, H. Devarajan, K. Bateman, J. Cernuda, N. Rajesh, and X.-H. Sun, "Chronolog: A distributed shared tiered log store with time-based data ordering," *Proceedings of the 36th International Conference on Massive Storage Systems and Technology (MSST 2020)*.