Towards Unified Intelligent High Performance Computing Storage Systems

Dong Dai, University of North Carolina at Charlotte, ddai@uncc.edu

Topic: This position paper focuses on a new storage-system architecture design.

Challenge: High-Performance Computing (HPC) faces significant I/O challenges. The applications' data volume and their needs for high-speed data accessing are growing fast. HPC storage is also expected to deliver low latency and high IOPS for emerging machine learning and artificial intelligence workloads [3].

To fulfill such requirements, modern HPC systems deploy multi-tiered storage stacks, which may include client-side on-demand file systems (e.g., GeKkoFS or BeeOND), burst buffers (e.g., DataWarp or IME), together with the traditional parallel file systems. These storage layers obtain distinct latency, bandwidth, capability, data visibility, and data durability attributes. As a result, how to leverage them most effectively and to obtain the best performance becomes a critical job to the end-users.

This places significant burdens on end-users as they need to know available storage options in the HPC systems, performance, capacity, data visibility, and durability. In addition to the usability challenge, letting users manually configure the multi-tiered HPC storage systems often misses significant runtime optimization opportunities [5]. For instance, without deep understandings of the I/O patterns, users may select sub-optimal I/O configurations (e.g., data size, location, or durability); without a global view of the entire storage system, multiple applications and users may compete for the same storage resource, resulting in a limited performance [7]; without a real-time view of the storage resource statuses, users can not dynamically schedule or tune I/O operations in runtime to improve the performance [2].

As the HPC storage becomes more complex and heterogeneous, effectively and productively using it will simply become more challenging; the gained performance will be unsatisfactory as well. This problem roots in the isolated design of current HPC storage systems and their high dependencies on users' manual tuning and configurations. Therefore, We see a need to shift its design from *isolated multi-tired storage* to *unified storage*; from *users' manual configuration* to *intelligent automatic management*.

Opportunity: The key to addressing the previously described challenges is to build *a unified, intelligent HPC storage system* that automatically delivers high I/O performance to end-users.

If we consider the heterogeneous storage devices are just running a slim layer of software to store data objects and communicate with each others, then to achieve such a unified and intelligent storage, we will need an intelligent data management system to conduct most of the configuration and tuning work, such as determining data locations, directing data buffering and moving, scheduling I/O requests, without any users' hints or manual configurations. We expect such data management decisions would be made by machine learning components based on historical patterns and real-time storage system status, effectively delivered by an extremely high-performance metadata layer. The machine learning components will also dynamically tune the I/O operations based on the real-time system status, together with achieving optimal performance.

To enable such a unified and intelligent storage system in HPC, we need to address multiple challenging questions that are barely touched in previous work.

- First, what metadata is necessary, adequate, and accurate for data management tasks, such as data location selection, visibility and consistency tuning, and dynamic IO scheduling? Many of these metadata are about the runtime status of other HPC components, such as batch job schedulers. How can they be integrated with the POSIX namespace to enable unified and easy metadata access?
- Second, how to support the common metadata operations and the data management functionalities efficiently to match the scale and speed requirements of modern HPC systems? The upcoming Exascale machines may deliver IO maximally at billions of random-read IOPS [1], which far exceed the IOPS

that SSDs and HDDs. Persistent memory seems promising. However, can it deliver such a speed to work with future systems?

- Third, how to improve the intelligence of data management such that it can coherently manage various IO tasks to achieve extreme performance? Existing studies touched a small part of these IO tasks such as data location, pre-fetching, or asynchronous data movement but lack understanding of how they work together and miss runtime I/O tuning. So, how the runtime I/O tuning can be done by machine learning models?

Timeliness or maturity: Building such a unified, intelligent HPC storage system becomes feasible now for two reasons: 1) the availability of new persistent memory (PMEM) devices to support needed fast metadata operations; 2) the progress in deep learning to accurately capture I/O patterns. Here we focus on explaining the maturity of persistent memory, while the machine learning part has been proven in many recent studies.

Feasibility of persistent memory. Persistent memory (PMEM) (particularly, Intel Optane DC Persistent Memory [4]) is a new kind of memory device that provides near-DRAM data access latency, higher capacity, lower-price, and data persistence. Its higher density, lower cost, and near-zero standby power cost make it a perfect choice for implementing our proposed global metadata management layer. For instance, for 128-byte small random reads, a single fully populated PMEM server can deliver 60 million IOPS [6]. We know that DOE's upcoming Exascale machine Frontier will deliver around 5TB/s read/write bandwidth and 2 million random-read IOPS. Its on-demand file systems on compute nodes will maximally deliver 75TB/s read bandwidth and 15 billion random-read IOPS. Then theoretically, we just need 25 PMEM servers to deliver 15B IOPS. Although this is just a theoretical calculation, it still shows the feasibility of persistent memory to deliver needed fast metadata management.

References

[1] OLCF ANNOUNCES STORAGE SPECIFICATIONS FOR FRONTIER EXASCALE SYSTEM. https://www.olcf.ornl.gov/2021/05/20/olcf-announces-storage-specifications-for-frontier-exascale-system/.

[2] D. Dai, Y. Chen, D. Kimpe, and R. Ross. Two-Choice Randomized Dynamic I/O Scheduler for Object Storage Systems. In *International Conference for High Performance Computing, Networking, Storage and Analysis, SC*, volume 2015-Janua, pages 635–646, 11 2014.

[3] G. K. Lockwood, D. Hazen, Q. Koziol, R. S. Canon, K. Antypas, J. Balewski, N. Balthaser, W. Bhimji, J. Botts, J. Broughton, et al. Storage 2020: A vision for the future of hpc storage. 2017.

[4] Optane. Intel Optane Persistent Memory. https://www.intel.com/content/www/us/en/products/docs/memory-storage/optane-persistent-memory/optane-dc-persistent-memory-brief.html, 2019. Accessed: 2019-11.

[5] Y. Qian, X. Li, S. Ihara, A. Dilger, C. Thomaz, S. Wang, W. Cheng, C. Li, L. Zeng, F. Wang, et al. Lpcc: Hierarchical persistent client caching for lustre. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–14, 2019.

[6] T. Tristian and L. Travis. Analyzing the performance of intel optane dc persistent memory in app direct mode in lenovo thinksystem servers, 2019.

[7] M. R. Wyatt, S. Herbein, K. Shoga, T. Gamblin, and M. Taufer. Canario: Sounding the alarm on io-related performance degradation. In *2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 73–83. IEEE, 2020.