Title: AI-driven Storage Resource Provisioning and Operations: Revisiting Old Assumptions and Meeting New Expectations

Authors: Valentine Anantharaj (anantharajvg@ornl.gov, Oak Ridge National Laboratory), Rafael Ferreira da Silva (silvarf@ornl.gov, Oak Ridge National Laboratory), Ali Butt (butta@cs.vt.edu, Virginia Tech), and Sarp Oral (oralhs@ornl.gov, Oak Ridge National Laboratory), Devesh Tiwari (d.tiwari@northeastern.edu, Northeastern University)

Topic: Storage-system architecture design; Utilizing AI to improve I/O patterns;

Challenge: End-to-end I/O subsystems are complex in nature, especially at large scales. We are designing I/O subsystems based on historical data and assumptions, some even decades old, which may or may not hold true for a system targeted for 4-5 years into the future [3,4,5] and is expected to have an operational life of 5 years beyond that [2]. On top of that, the user workloads and I/O patterns are now changing in unpredictable ways, especially with the advent of AI in large-scale systems and this trend will continue with the integration of edge scientific experiments and instruments. Even today, on a large enough system, multiple applications are running concurrently (e.g., large-scale complex AI workflows that inherently couple various types of tasks such as short ML inference, multi-node simulations, long-running ML model training, etc. [1]), and these different applications are generating a mixed I/O workload utilizing traditional and specialized computing hardware (e.g., GPUs, quantum, neuromorphic chips) observed by the file and storage systems. On one hand, we have a wealth of log and telemetry data coming out of a large-scale compute and I/O system from across all layers of the OS and I/O software stack and hardware components (in terms of variety, velocity, and volume), simply beyond our current capabilities to meaningfully stitch together, analyze, and take action (design or operate). On the other hand, we are not getting enough and high fidelity data in real time from applications and I/O middleware libraries. We have new opportunities to design and operate better I/O subsystems given the data we have, but we are also missing fundamental comprehension of how applications are individually utilizing a given I/O subsystem or as a collection at the system level, and therefore failing to provide actionable feedback (real time or post mortem) to them on how they should improve their I/O behaviors.

Opportunity: To mitigate these data processing challenges, we argue that we need to meaningfully and intelligently reduce and filter the data. We further argue that to effectively operate a large scale storage system using a data driven approach we need to: (1) institutionalize and limit the number of “learning points”, and (2) use the learned models to “control” and achieve certain holistic system-level targets instead of individual-application focused metrics (e.g., system throughput, system-level I/O control congestion). These learning points can be placed on certain I/O servers and routers — instead of collecting data from every single source of the system along the end-to-end I/O path — learning points act as a representative sample and limit the data that needs to be ingested [3,4]. Data collected from these learning points are then fed to the “action controllers” [3,4]. These action controllers can essentially act as “recommendation implementers” to meet certain system-level objectives via better resource allocation (e.g., I/O bandwidth allocation, checkpointing frequency [6]). For example, we envision that these action controllers are embedded into job schedulers and I/O servers and routers to selectively co-schedule application traffic. These components will leverage control-theoretic property with the AI power to ensure that AI power is being harnessed but in a controlled way. This approach will also allow us to develop robust “learning points” and “action controllers” that can rely on extensive system-level benchmarks that can periodically calibrate these “learning points” and “action controllers” with ground truth [3,4]. Unfortunately, developing a representative system-level benchmark is difficult, but having this feedback-based approach (learning points and action controllers) will help us refine the benchmarking process itself and become more useful. In some sense, the benchmarking itself will become automatic and AI-driven, where it helps us achieve target objectives better (e.g., system throughput, I/O bandwidth allocation, checkpointing frequency). We believe that such an intelligent (data and model driven) system-level benchmark will allow us to design better and more cost-effective I/O subsystems.

We also have the opportunity to design a prediction system that would leverage both logs obtained from actual systems, and data that could be obtained from simulations of the system – i.e., a digital twin that could explore unforeseeable scenarios, or how the currently available technologies would perform on novel architectures. By

combining both types of data, it is possible to develop ML models (with acceptable confidence) that could be used to (1) identify current and upcoming system bottlenecks, and then (2) infer the design of novel technologies/solutions to address these challenges.

Timeliness and Maturity: Frontier at ORNL is being deployed today, and within the next two years El Capitan at LLNL and Aurora at ANL will be deployed. All these installations have I/O subsystems, speced and designed 4-5 years ago, are tuned for writing out large volumes of data, from multiple ranks, in the shortest possible time. These requirements may be based on, say writing a dump of the entire system memory in X seconds. This may capture the state of the application in restart and/or analysis files. One of the considerations in the past has been the MTBF of large systems. These stringent performance requirements also resulted in higher procurement costs and increased operational overhead. During the same time period commercial cloud service providers have developed and refined cost-effective approaches toward operational reliability. Over the past decade leadership class systems have become relatively more stable. The emerging class of AI and data intensive applications mostly require efficient and performant data ingress and egress operations. Besides, many digital twin (DT) applications are loosely coupling simulations, analytics, inferencing, synthesis and decision-support capabilities that could take advantage of native support for complex workflows and data management. Application developers and users prefer to think about data in a much more natural way that need not necessarily be I/O-centric. In some instances, the data may need to be desegregated (from files) and then reassembled in multiple ways to support various components of the digital twin applications. The bespoke workflows involved in DT applications result in complex I/O patterns that can occur concurrently during the course of the simulations and learning/inferencing phases. The emerging application needs for data management and DT workflows would need to be supported at multiple levels in the storage hierarchy. This requires intelligent provisioning and management of storage systems.

References:

[1] Ferreira da Silva, R., Casanova, H., Chard, K., Altintas, I., Badia, R. M., Balis, B., et al., A Community Roadmap for Scientific Workflows Research and Development, in 2021 IEEE Workshop on Workflows in Support of Large-Scale Science (WORKS), 81–90, 2021.

[2] Sarp Oral, Sudharshan S. Vazhkudai, Feiyi Wang, Christopher Zimmer, et.al., 2019. End-to-end I/O portfolio for the summit supercomputing ecosystem. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC '19). Association for Computing Machinery, New York, NY, USA, Article 63, 1–14.

[3] Tirthak Patel, Suren Byna, Glenn K. Lockwood, Nicholas J. Wright, Philip Carns, Rob Ross, and Devesh Tiwari, “Uncovering Access, Reuse, and Sharing Characteristics of I/O-Intensive Files on Large-Scale Production HPC Systems” USENIX FAST 2020.

[4] Tirthak Patel, Suren Byna, Glenn K. Lockwood, and Devesh Tiwari, “Revisiting I/O Behavior in Large-Scale Storage Systems: The Expected and the Unexpected”, Supercomputing (SC) 2019.

[5] Ross, Robert, Ward, Lee, Carns, Philip, Grider, Gary, Klasky, Scott, Koziol, Quincey, et.al., “Storage Systems and Input/Output: Organizing, Storing, and Accessing Data for Scientific Discovery”. Report for the DOE ASCR Workshop on Storage Systems and I/O. [Full Workshop Report]. United States: N. p., 2018. Web.

[6] Devesh Tiwari, Saurabh Gupta; Sudharshan S. Vazhkudai “Lazy Checkpointing: Exploiting Temporal Locality in Failures to Mitigate Checkpointing Overheads on Extreme-Scale Systems”, DSN 2014.