Rethink Energy-Efficient Storage Stack for Exascale Computing in the Hardware-Accelerated I/O Era

Ming Liu, *University of Wisconsin-Madison, mgliu@cs.wisc.edu*
Topic: *system software and emerging technologies*

1. Abstract
The rising storage volume and data movement demand of Exascale computing makes conventional HPC storage servers dense and power-hog. Emerging domain-specific accelerators come to the rescue, enabling hardware-accelerated I/O processing. However, conventional stacks can barely benefit from these capabilities due to their decade-old "smart-sender dumb-receiver" design philosophy. We radically revisit this issue and propose a revamped storage stack that can automatically refactor the I/O processing over on-path accelerators based on energy appraisal. This position paper serves as a "call to arms" for developers, vendors, and practitioners to build the next-generation energy-efficient I/O stack.

2. Challenge: Storage Servers Become Dense and Power-Hungry
Storage servers in HPC clusters today are based on general-purpose processors. They usually enclose one or two x86 CPUs and a dozen HDDs/SSDs, run the Linux storage stack with remote storage protocol (such as NVMe-over-Fabric), deploy a parallel file system (like Lustre) or other storage services, and provide various data abstractions for scientific applications.

Lately, storage servers have become dense and power-hungry. To satisfy the skyrocketing data volume and application demand, they are equipped with a rising number of fast (i.e., PCIe Gen5/6) and physically compacted (like EDSFF) NVMe drives, whose I/O density has increased dramatically. As a result, to fully use the storage bandwidth and sustain a higher throughput (e.g., several to tens of millions of IOPS), one needs dozens of cores to busy-drive the I/O parallelism, yielding dramatic power consumption. This trend has been demonstrated in recent commodity storage boxes, consuming more than five hundred watts.

3. Opportunities: Hardware-Accelerated I/O
The last few years have seen several hardware innovations in the storage landscape. People develop a number of domain-specific accelerators [1] along the I/O data path to improve the performance and efficiency of I/O processing (Figure 1-a). We highlight them below:

- ***Open and configurable SSD:*** SSD-class devices have traditionally been opaque and provide limited internal visibility. However, new and upcoming SSD models offer greater transparency and control over internal operations. For example, ZNS/FDP SSDs introduce a new command set for block partitions and write alignments, Open-Channel SSDs delegate data placement and I/O scheduling to the host, not to mention computational SSDs enclosing execution engines for application-specific offloading;
- ***Low-cost and low-power storage platform:*** People have attempted to deploy specialized and low-cost storage targets, as opposed to today's server-based ones. For example, there are SmartNIC-based JBOF ("just-a-bunch-of-flash") designs, wherein a standalone (server-less) SmartNIC houses NVMe drives and is responsible for chaining the network and storage protocols. An EBOF ("ethernet-bunch-of-flash) solution streamlines the storage target by baking this chaining logic in an ASIC. Both solutions deliver orders of magnitude higher IOPS per Joule (Figure 1-b);
- ***Transparent and reconfigurable storage adapter:*** They enable interposing application-defined storage logic. This allows client application software to operate in an unmodified form but still enjoy the benefits of virtualization and in-line transformations that are critical to improving performance and energy efficiency;

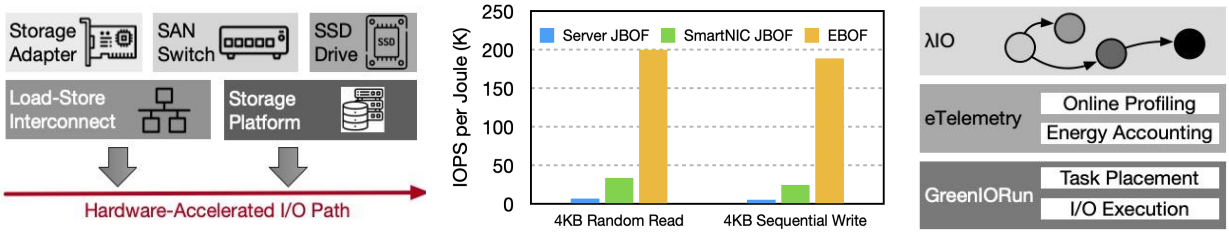

Figure 1: *(a) shows different hardware accelerators; (b) reports the energy efficiency of three storage targets when handling 4KB reads/writes; (c) sketches the proposal.*

- ***Programmable SAN switching fabric:*** RMT switches allow for extending the scheduling and isolation control even within the network fabric. One can develop an effective in-network scheduler to transmit I/Os at an application-specified deadline;
- ***Load-store Interconnect:*** They facilitate memory-semantic communications and can dramatically simplify the design and execution of remote storage protocol at the initiator and target side. Further, these interconnects expose flexible command extensions;

In sum, storage infrastructure is becoming increasingly open and programmable. By effectively harnessing these hardware-accelerated I/O opportunities, one can dramatically improve the energy efficiency of the storage stack for Exascale computing.

4. Vision: An End-to-End Auto-Refactored I/O Stack Based on Energy-Appraisal

Albeit promising, these emerging technologies completely change the system integration interfaces we have held for many years. *Existing storage stacks employ a "smart-sender and dumb-receiver" design philosophy that implements sophisticated I/O handling logic on the host processor and views the storage drive as a simple block device with just read/write I/O interfaces.* This squanders the above hardware-accelerated I/O opportunities because the I/O data path becomes computational and can hold partial or even full I/O processing logic from the host. Thus, we should rethink how to architect the storage stack in this new era.

Our idea is to build an end-to-end auto-refactored I/O stack based on fine-grained energy appraisal. The envisioned system can (a) continuously perform energy accounting at the per-IO granularity; (b) reconstruct the I/O data path on the fly and map tasks to a suitable hardware substrate; (c) orchestrate the I/O execution to maximize throughput and energy efficiency without exceeding the computing limit of the corresponding devices. Realizing such a design is non-trivial and raises a number of questions. We propose a system design (Figure 1-c):

- **λIO**: a local microservice framework on the host side that translates each I/O processing into a chain of self-contained small lambda subtasks. This raises the following questions: how to maintain a (semi-)transparent interface to applications, how to capture application semantics, and how to design the programming system;
- **eTelemetry**: a distributed telemetry system that performs always-on energy profiling and serves as the basis for I/O execution. It requires us to tackle: how to measure power for each device, how to maximize the measurement precision, how to develop a power model when there is a need, and how to attribute the power to different I/Os.
- **GreenIORun:** an energy-aware I/O orchestration engine that maps lambda subtasks to different storage accelerators and takes care of the entire I/O data plane. We should address: how to max out energy efficiency and performance, what is the multi-tenancy guarantee, and how to reduce the runtime execution overheads.

5. References:

[1] Dally, William J., Yatish Turakhia, and Song Han. "Domain-specific hardware accelerators." Communications of the ACM 63, no. 7 (2020): 48-57.