Open Ethernet Drive:
Evolution of Energy-Efficient Storage Technology

Hariharan Devarajan[*], Anthony Kougkas[*], Hsing Bung Chen[†], and Xian-He Sun[*]
[*]Illinois Institute of Technology, Department of Computer Science, Chicago, IL
[†]Los Alamos National Laboratory, Los Alamos, NM
{hdevarajan, akougkas}@hawk.iit.edu, sun@iit.edu, hbchen@lanl.gov

***Abstract*—An Open Ethernet Drive, also known as an OED, is a new technology that embeds a low-powered processor, a fixed-size memory and an Ethernet card on a disk drive (e.g. HDD or SSD). All major storage vendors have introduced their respective implementations with similar architectural designs. As the technology progresses into its second generation, the performance characteristics have improved substantially. In this study, we first demonstrate the differences between two generations of the OED technology. We run a variety of benchmarks and applications to thoroughly evaluate the performance of this device and its compatibility with the current ecosystem. Furthermore, we investigate the performance and energy footprint of the OED technology when used as a storage server and as an I/O accelerator. Evaluation results show that OED technology can be a reliable, scalable, energy and cost efficient storage solution. It is a viable replacement for storage servers offering competitive performance while consuming only 10% of the power that a typical storage node needs. Finally, this study shows that OED's can be used as I/O accelerators capable of executing data-intensive computations (such as sorting, compression/decompression, etc.) on local data, whereby the expensive data movement is minimized resulting in low power consumption.**

I. Introduction

Modern supercomputers produce and analyze data at an unprecedented rate. This growth of data is even faster than Moore's Law [1]. Many fields like social informatics [2], weather simulations [3] and computational biology [4] have become very data-driven. As we move towards the exascale era, data management is one of the greatest challenges which fuels the evolution of both hardware and software. ActiveStorage [5] and ActiveDisk [6] propose taking advantage of the embedded processors on the storage servers. These enhancements in technology focus to improve performance with a huge power requirement [7]. Up to 40% of the total energy consumed in these clusters is by the storage nodes [8]. This ratio would increase due to two factors (a) the power consumption of computing resources has been getting a lot of attention resulting in more efficient utilization; and (b) data deluge (expected to increase ten fold) will also increase the relative contribution to power consumption. The above factors have caused reduction in power consumption to become a central goal in data center's management.

Many researchers have worked on reducing power consumption on the storage nodes by proposing solutions based on either powering off storage devices or placing data smartly. MAID [9] employs cache disks with recently read data to improve access of hot data while powering down the archival storage. However, these scheme benefits from systems where data is rarely accessed (e.g. archival) which is not the case for most data centers. Popular Data Concentration [10] collects popular data together to conserve power on other nodes. This approach requires having a knowledge of the data access patterns of the application. Hibernator [11] combines lower disk clock speed and concentration of data to save power. This approach results in less availability of data and creating hot spots degrading the performance of the I/O system. Others, like Practical Power Management [12] and Pergamum [13], delay access to some storage devices to save power but this increases the latency for applications. Hence, it is imperative to look for more energy efficient technologies to reduce the power consumption of distributed storage systems.

One new technology that has the potential, to reduce the energy consumption in storage nodes, is the Open Ethernet Drive (OED). Its architectural characteristics enable ”data-centric” storage services. An OED device consumes only 10% of the power required by a typical storage node while maintaining reasonable performance [14]. The technology brings computation capabilities close to the data. Each OED can be treated as a storage node with the processor and RAM embedded close to the disk. This makes the OED an ”intelligent” drive with the capability of not only storing data but also executing certain data-centric computations. OED technology has evolved into its second generation with better hardware specifications while maintaining low power consumption. The device architecture is developed by several vendors each with their own minor variations. In HGST's implementation [15], each OED comes loaded with a Debian OS. This enables programmers to run scientific applications and software natively. The OED technology offers a cost-effective mechanism to distribute storage services across several low-powered processors like ARM-based instead of a few more powerful server-graded CPUs (e.g., Intel Xeons or AMD Opterons). Having the capability to manage data close to its origin enables several performance optimizations through the device. Apart from the specification differences, OED technology is not just using Ethernet as a new connection interface but it also moves the communications protocol from simple commands to read-and-write data blocks to a higher level of abstraction.

In this study, we explore the potential of this new generation of OED in HPC by evaluating the architecture. This is done by benchmarking the performance of the device and conducting an energy cost analysis while comparing it with its predecessor. Additionally, we explore the possibility of using this new technology as (a) storage services (i.e. parallel file system servers and/or key-value stores) and (b) I/O accelerators (i.e. in-situ analysis nodes, burst buffer nodes, etc.). In terms of I/O optimizers, the OEDs can be used for performing administrative tasks like compression/decompression, de-duplication and statistics. It could also be used as specialized storage entities in architectures like Decoupled Execution Paradigm (DEP) [16]. With respect to energy consumption, due to its architecture it consumes much less power than a typical server node. In this paper we provide all the metrics defining OED characteristics and behavior to explore this technology as a viable alternative for HPC storage infrastructures.

TABLE I: Hardware specifications

Feat	OED 1st Gen	OED 2nd Gen	Server Node
CPU	ARM 32bit 1-core (1GHz)	ARM 32bit 2-cores (2.2GHz)	2xAMD Opteron 8-cores (2.3GHz)
RAM	2GB DDR3 1-Channel 1600Mhz	1GB DDR3 2-Channel 1600Mhz	8GB DDR2 1-Channel 667Mhz
Disk	Megascale DC4000.B 4TB 7200rpm	Megascale DC4000.B 8TB 7200rpm	WD 2TB 7200rpm
Net	1 Gbit/s	1 Gbit/s	1 Gbit/s
OS	Debian 8.0	Debian 8.1	Ubuntu server 12.04
Kernel	3.14.3	3.9	2.6.28
Year	2014	2016	2011

II. Background

A. Motivation

Traditionally, performance improvements have been the main focus in HPC. However, power consumption of these systems has already become a major concern. The top three supercomputers in the Top500 list [17] operate at roughly 16,000 kW. This fact gave birth to the Green500 [18] list where supercomputers compete for the *power efficiency* metric. The top machine in this list demonstrates a power efficiency of 14.110 GFlops/watts. However, the top machine in the Top500 list (i.e., the fastest in the world) has a power efficiency of only 6.051 GFlops/watts. Among the many components in the supercomputers, storage is the next largest consumer of power after compute nodes. It consumes about 20% of the total power [19]. The Trinity supercomputer at Los Alamos National Lab built by Cray [20], and the Sequoia supercomputer of Lawrence Livermore National Laboratory built by IBM [21], are classical examples of such systems. Here, the storage solutions consume around 17-20% of the total power consumption [22], [23], [24]. The similar picture exists in generic data centers where the power consumption contributes to as much as 50% of the total cost of ownership of these sites [25]. Data centers alone in the U.S. consume 2% of the electricity consumption [26]. This energy consumption by the data centers will grow by 4% every year [26]. The percentage of power consumption of the storage infrastructure in HPC will increase as there is a continuous rise in data intensive applications. The supercomputers are designed to meet high and sustained bandwidth requirements under highly concurrent workloads. Recent research suggests that modern scientific applications show bursty I/O access patterns as they alternate between the computational and I/O phases [27], [28]. Since I/O happens periodically in bursts, storage systems in supercomputers waste a lot of power during I/O idle phases. Hence, as we move towards the exa-scale era, it is imperative to seek and employ more energy efficient hardware and software storage technologies.

B. Open Ethernet Drive Architecture

An Open Ethernet Drive, also known as an OED, has an ARM-based processor; a fixed-size RAM and an Ethernet card embedded onto a disk drive. It is designed to bring computation closer to the data. By connecting a number of OEDs together in some type of enclosure, a relatively capable cluster is created. It may be noted that the prototype devices we study in this paper are implemented by WD's HGST and any details provided will refer specifically to this implementation. Other vendors might add or remove features and/or hardware details. Each OED device runs Debian 8.0, offering a rich feature set of the familiar Linux ecosystem, which is already a dominant choice in scientific computing. This allows seamless integration of the storage medium with the tools needed to optimize and manage its use. A 32 bit ARM CPU clocked at 1 Ghz along with 2 GB of single channel DDR3 RAM are co-located with a 4 TB 7200 rpm hard drive. From the available RAM 300 MB are kept for the OS and system tools whereas the rest of the RAM is available for applications. A 1 Gbit/s Ethernet card completes the hardware specifications of such device that maintains a standard 3.5" HDD form factor. A serial port is also present to facilitate administrative tasks such as upgrading the software. HGST has presented a 4U enclosure (i.e., a JBOD), that contains 60 such drives offering a 240 TB total storage capacity. The components of this enclosure are hot-swap capable; it has also an embedded switched fabric. The internal network's bandwidth is 60 Gbit/s and there are four 10 Gbit/s connections for external connectivity. The OED hardware has evolved into its 2nd generation. Table I shows the specifications of both OED generations. As a reference, we list a typical server node's characteristics. The 2nd generation of OED comes with Debian 8.1 OS. It is embedded with a 32bit ARM v71 dual core CPU, clocked at 2.2 Ghz. It has a dual channel DDR3 RAM with a capacity of 1 GB. All these components are embedded on a 8 TB 7200 RPM hard drive with a 6.0Gb/s SATA connector. Additionally, a USB 2.0 port is now available to facilitate plugging of external devices.

Since its inception, OED technology is open sourced and made available to the public through OpenStack. Several companies have already presented use cases of OEDs. The Kinetic Open Storage Project [29] formed in 2015 under the aegis of the Linux Foundation dedicated to create an open standard around Ethernet-enabled devices. This collaborative project includes many industry leaders such as DELL, Seagate, Toshiba, and Western Digital. Several companies presented use cases of this technology and demonstrated its strengths and weaknesses. Mirantis is a company that delivers all the software for running OpenStack [30]. It deployed OpenStack's Swift object store, Ceph's OSDs and GlusterFS's bricks (i.e. the basic unit of storage) on top of an OED JBOD of 60 drives. Cloudian [31], a software-defined storage company, famous for its HyperStore smart scale storage platform, successfully deployed HyperStore servers on top of the OED technology. The deployment sought to answer two fundamental questions: is this even feasible and if yes, how well would it perform? They used Yahoo Cloud Serving Benchmark [32] to test the setup and concluded that all the tests were successful. They even reported that the OED architecture could offer a host of opportunities for optimizations on their applications. Finally, Skylable [33] is a company whose mission is to build a fast, robust and cost-effective object-storage solution. It had experimented and deployed their Skylable SX services on top of HGST's OED technology. The report was released after performing a series of tests from simple feasibility to resiliency and performance. They concluded that OEDs are the perfect building block for an energy efficient and horizontally scalable storage cluster.

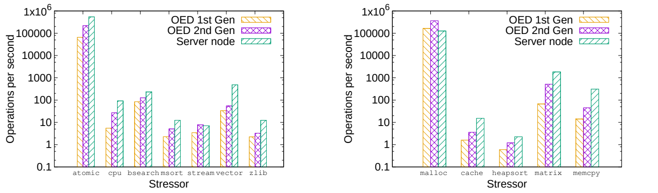
In our previous study [14], we evaluated the capabilities of the 1st generation of this technology. We concluded that the performance of the device, at that time, was not at par with a typical server node. However, it showed potential as it could perform several types of computation while consuming much less energy (i.e. 1/10th of the power needed by a typical server). Furthermore, its small form factor would lead to less cooling requirements and thus, it would potentially allow storage system deployments to consume less energy while maintaining normal operations. The hypothesis was untested in our previous work. Even though the hardware capabilities of 1st generation OEDs seem relatively lower compared to a modern HPC storage node, the 2nd generation shows promise with its renewed specifications. There are many benefits from this architecture like a) hardware costs, b) overall size, c) power and cooling requirements, and d) ease of maintenance.

It is our hypothesis that the HPC environment could benefit from the usage of OED technology in several ways: deploying storage services (e.g. parallel file system or key-value store servers) on top of an OED enclosure, applications could leverage the embedded resources and perform in-situ data analysis with lower cost. Similar to GPGPUs, an OED JBOD can be also used as an I/O accelerator (i.e. burst buffer nodes). In this study, our goal would be to examine if this technology and its evolution would be capable enough for the high-end computing and the use cases we propose. It is our belief that OED technology can provide a high-performance, reliable, and scalable storage solution while consuming a fraction of the energy required by today's storage infrastructures.

III. Approach

A. The evolution of OED technology

The OED technology has evolved into a better power-efficient hardware. In Section II-B we presented the differences in the specifications of the two generations with respect to a typical server node. The 2nd generation has a dual-core CPU with a higher clock frequency. This is very important as it allows faster computations, higher concurrency, and fewer contentions for processes. It has a smaller capacity of RAM (i.e. 1GB) but is dual channel and thus, allowing concurrent accesses to the memory banks. Hard drive capacity has also doubled with similar disk speeds. In summary, the 2nd generation OED is more powerful and shows great potential to alleviate the shortcomings of the 1st generation OED.



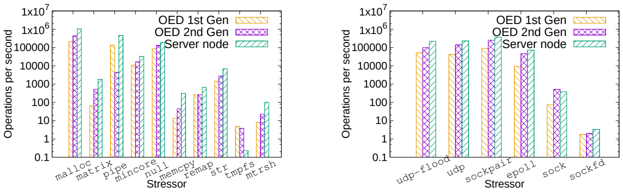
(a) CPU (logscale)

(b) CPU Cache (logscale)

Fig. 1: CPU evaluation

With that in mind, we strive to examine the performance differences between the two generations of OED drives. We run a collection of benchmarks that evaluate the capability of the internal components. Specifically, the components are categorized as follows a) CPU b) CPU-cache c) main memory (RAM) d) network and e) disk. For computing components (like CPU, cache, memory and network) , the benchmark suite we used is *Stress-ng* [34]. It is a popular benchmark designed to stress various physical subsystems of the computers as well as various operating system kernel interfaces. However, to test the storage component like disk performance, we use *IOR* [35]. It is a widely used benchmark for measuring I/O performance at both MPIIO and POSIX level. This parallel program performs, reads from and writes to the files under several conditions and reports the resulting throughput rates. We use fine-tuned cases to stress each of the above mentioned components to perform detailed study. Each component is tested with various stressors to stress different parts of these components. The aim of such tests is to study the individual capability of these systems and identify their place in the HPC ecosystem. It may be noted that for readability of some graphs they are presented in logarithmic scale which is mentioned wherever such a scale is used. Finally, the systems compared in this section are 1st generation OED, 2nd generation OED and a typical server node as a reference. In order to achieve a fair comparison between our diverse hardware, we have restricted the available memory and the CPU-core affinity of the server nodes to match the OEDs. Specifically, the available RAM is set to 1GB (i.e., same with the OED) and the CPU core count to 2. These restrictions only apply to results in this section.

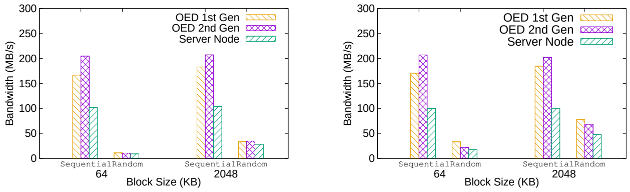
CPU: In this test, we selected stressors that cover a wide variety of operations like searching, sorting, vector math, compression and cpu operations. The choice of these stressors stem from the architecture of the OED drives. An OED is not meant to substitute a compute node CPU but to act as the *brain* of the disk drive. Therefore, we tested algorithms frequently used in data-centric computations. Figure 1(a) shows the results



(a) RAM (logscale)

(b) Network (logscale)

Fig. 2: RAM and Network evaluation



(a) Read

(b) Write

Fig. 3: Disk performance (access pattern)

expressed in operations per second with the Y-axis in log scale. It is clear that the CPU of the 2nd generation OED is faster than its predecessor by up to 5x. The server node performed from 1.6 to 4x higher than both OEDs which is expected as the CPU type is of x86 family and the clock frequency is higher. In summary, the newer ARM-based CPU of the 2nd generation OED brings the performance closer to the server node.

CPU Cache: In this test, we selected cache intensive stressors like matrix multiplication, sorting and few memory operations. Again, the choice of these stressors is inspired by the fact that OEDs were designed to be used in data-centric environments. Figure 1(b) shows the results expressed in operations per second with Y-axis in log scale. As it can be seen, the CPU-cache of 2nd generation is on average 2.7x faster than the 1st generation OED. The server node is 2.2x faster than 2nd generation and 4x faster than 1st generation OED. The improvement in cache performance of the 2nd generation is impressive and can lead to higher overall computational performance.

RAM: Main memory-intensive stressors like creation, copying, freeing and piping were selected in the following test. We chose these stressors to investigate the RAM's capability to perform in-memory I/O operations. Figure 2(a) shows the results as operations per second. Y-axis is in log scale. It is evident that 1st generation OED is, on an average, 1.5x slower than its successor. The server node, is 2x faster than the 2nd generation OED. The introduction of more channels in RAM for 2nd generation OED is the cause of this performance improvement.

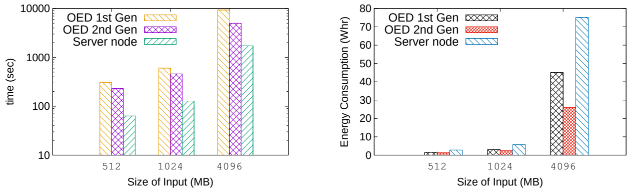
Network: In this test, we selected stressors performing operations like poll, socket and UDP operations. These stressors show the capability of the OEDs to handle network traffic and showcase the performance in a cluster environment. Figure 2(b) shows the performance in operations per second with Y-axis in log scale. Network-interface for both OED generations have similar performance as there is no change in hardware (i.e., same Ethernet card). However, due to other components, like CPU and RAM, both 2nd generation OED and the server node appears to be faster than the 1st generation OED. This result is encouraging since it demonstrates the networking capabilities of the OED technology especially in an enclosure where multiple drives will be part of the network fabriq.

Disk: In this last test of internal components, we examine the read and write bandwidth of the enclosed disk drive in the OEDs. This is important as I/O access is the biggest bottleneck in data-centric computing. Figure 3(a) and 3(b) show the results in bandwidth in MBs/sec. It is clear that 2nd generation OED is 1.4-1.8x faster than its predecessor and server node respectively for reads and 2-3x faster for writes. The actual disk drive inside the 2nd generation OED is physically faster than the other devices which accounts for its better performance.

It is noteworthy that the combination of all internal components can lead to an overall higher performance of each device. The 2nd generation OED has improvements in CPU and RAM. This can grant better I/O performance even with the same type of internal hard drive. These results show the OED technology has evolved in its hardware and it remains to be seen how would these devices perform under real applications. The OED technology is still in its infancy but demonstrates continued improvement with its 3rd generation expected to be even faster [36]. It is the small form factor and the computation capabilities for in-situ analysis that drives us to suggest that OEDs can be building blocks to a scalable, reliable, and energy-efficient storage solution.

B. *Proposed use cases for OED in HPC*

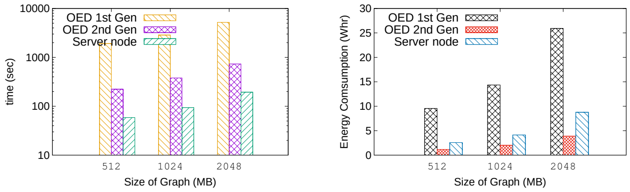
Driven by the performance characteristics of the OED technology we propose several use cases in an HPC environment. First, the bursty I/O behavior [37], [38] that most scientific applications demonstrate means that the storage layer in a supercomputing site is being used periodically [27], [39]. This motivates us to propose several JBODs of OED drives as a substitute for expensive and power hungry storage servers. The smaller size of OEDs means less space, less cooling needs, and less energy. Where OED lacks in performance, it can increase parallelism. Similar with RAID arrays, we can employ a large number of OEDs in enclosures and distribute them around the compute nodes. This will create a parallel file system (PFS) that can absorb incoming I/O. This will consume a fraction of the energy needed to maintain a typical cluster of few thousand full blown servers. Secondly, we could deploy a key-value store (KVS) service co-located with the parallel file system and support a wide variety of data-intensive applications and frameworks [40]. Our motivation comes from existing examples described in Section II-B. In-view of the above, we suggest that a JBOD of OEDs, with its Linux OS and the network capabilities, can easily run both the above services at the same time. Finally, past research suggested to offload data analysis and visualization kernels onto the storage layer (i.e., ActiveStorage, ActiveDisk, ActiveFlash, ActiveBuffers etc.). The ease of deployment, the networking, the ease of



(a) Performance (logscale)

(b) Energy Consumption

Fig. 4: External Integer Sorting



(a) Performance (logscale)

(b) Energy Consumption

Fig. 5: K-means Clustering

maintenance, the hot-swap-ability of the components and the scalability of the OED technology makes it an ideal alternative to be used as *I/O accelerators*. Burst buffers [41], active buffers or data analysis components are all good use cases for this new energy efficient technology. In the next section, we evaluate our hypothesis for all the above use cases. We ran real applications, and we deployed OEDs in numbers to test the performance and energy consumption in a cluster environment.

IV. Evaluation

A. Methodology

The evaluation of the OED technology is focused into two major categories: performance and energy consumption. To measure the devices performance, we divided our tests into two major aspects: individual device performance under real applications, OED cluster performance under proposed use cases. All performance results are either expressed in overall execution time (i.e., seconds) or bandwidth/throughput in MB/sec or Operations/second. For measuring the power consumption, we use the same test cases. We capture the power consumption in watts. We define *Energy Efficiency* metric as the product of the Active Power Consumed and Execution Time for running the application and it is expressed in Whr: $Energy\_Efficiency = P(Watts) * T(hours)$

Hardware Used: All experiments are conducted using HGST's prototype implementation of both generations of OED technology. The OED devices are compared with a typical server node, part of a 65-node SUN Fire Linux Cluster located in Illinois Institute of Technology. Note that we chose to use this cluster since we have physical access to it and therefore we were able to manually collect the power consumption readings. The specifications of the above machines are shown in Table I in Section II. For the clustered tests, we use up to 32 processes on client nodes and 8 nodes for deploying storage services (i.e. PFS or KVS). For the OED cluster we use a JBOD in Los Alamos National Laboratory (LANL) consisting of 30 OEDs of each generation. All the machines are connected with a Gigabit Ethernet switch which is sufficient to support our experiments without being a bottleneck. Finally, to measure the power consumption, we use a power logger by HOBO [42]. It is a watt meter with a capability of storing the active power consumed over a period of time. This power meter captures the power consumption of the entire machine and not of the internal components (i.e., CPU or RAM). We extracted the logs and analyzed them externally.

Software Used: A combination of our own application micro-kernels and some well-known open-sourced benchmarks are executed on all machines tested. In order to cover a wide variety of data-intensive workloads we selected one application from each of the following categories: integer out-of-core sorting [43] for algorithmic problems, K-means clustering [44] for machine-learning problems, and breadth-first search (BFS) from Graph500 [45] for graph exploration problems. Additionally, we run IOR [35], a famous I/O benchmark which measures I/O performance at both MPIIO and POSIX level. Also, we ran Yahoo Cloud Serving Benchmark [32], a widely used framework with variety of workloads for evaluating the performance of different key-value and cloud serving stores. For a parallel file system, we deployed OrangeFS 2.9.6 [46] (formerly known as PVFS2) and for the key-value store we used Redis 4.0.1 [47].

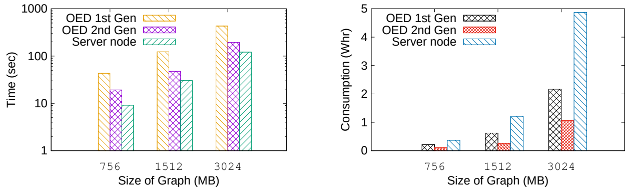
Experimental Setup: For IOR we performed direct I/O to test the disk performance eliminating read-ahead and caching from the OS and the disk driver. For the key-value store, cache size was set to 500M for both systems forcing the data to be flushed down to the disk. Lastly, all experiments have been executed 5 times and we report the average. Note that, for clarity,some of the presented figures are in logarithmic scale and are noted in the figure description appropriately.

B. Results with Real Applications

External Integer Sorting: The out-of-core integer sorting application takes a file of random integers as an input and performs arithmetic sorting according to the keys. If data cannot fit into memory, it performs sorting in phases. It first reads a chunk of the unsorted data into memory, sorts them, and writes back the intermediate lists sorted. Once this phase is done, it reads the intermediate chunks and merges them together to produce the final sorted output. In this test, we used three input files: a 512 MB file that fits entirely in memory, a 1024 MB equal the physical memory available, and a 4096 MB that will be sorted in phases (i.e. external sorting).

Figure 4(a) demonstrates the performance results. The 2nd generation OED shows a big performance improvement over its predecessor by performing sorting almost 2x faster. The server node performed sorting 5.3x and 2.8x faster than OED 1st and 2nd generation respectively. All input files are local to each machine's local disk drive.

We present the energy consumption results in figure 4(b). It can be seen that the 2nd generation is more energy efficient than the 1st generation OED. Most interestingly, both OEDs



(a) Performance (logscale)

(b) Energy Consumption

Fig. 6: Breadth-First Search

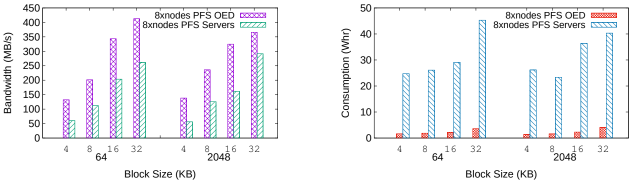
consume much less energy to perform the same task when compared with the server node. In detail, 1st generation OED needed 9288 seconds to complete the sorting of 4096 MB. At 17.44 Watts power consumption at full load, the energy efficiency of this device is 45 Whr. For the 2nd generation OED, it took 4984 seconds to complete the sorting and at 18.68 Watts at full load, it demonstrates an energy efficiency of 25.8 Whr. Finally, the server node took 1722 seconds to complete the same task and with 156 Watts power consumption the energy efficiency of this device is 75 Whr. The 2nd generation OED has **3x** better energy efficiency than the typical server node and thus, it is a viable alternative solution for local data-centric computations.

K-means clustering: The K-means clustering algorithm aims to find the evenly spaced sets of points in subsets of euclidean space and partition these subsets into well-shaped and uniformly sized convex cells. The algorithm starts with initial placement of some number k of points in each cell. It then repeatedly computes the centroid for each cell and moves the k points till it converges. This algorithm is a mixture of computation and I/O intensive phases. It reads points from the disk in several phases when it cannot fit them in memory. We used three point data sets with 512 MB, 1024 MB, and 4096 MB total size respectively.

Figure 5(a) shows the performance results. We observed that 1st generation OED ran the application in 5201 seconds for the out-of-core case of 4096 MB. The 2nd generation OED only needed 731 seconds, a **7x** boost in performance over its predecessor. Finally, the server node was faster than both the OEDs and completed the test in roughly 200 seconds.

Figure 5(b) shows the energy consumption results. The 2nd generation OED has an energy efficiency of 3.91 Whr which is **2.5x** better than the server node for the out-of core case of 4096 MB. Even though the server node performed the test faster than the OEDs, its 164 Watts power consumption at full load resulted in less energy efficiency. Also note that, the energy-efficiency of 1st generation OED is 25.91 Whr which shows that improvement in hardware for 2nd generation OED has resulted in 8x better energy-efficiency.

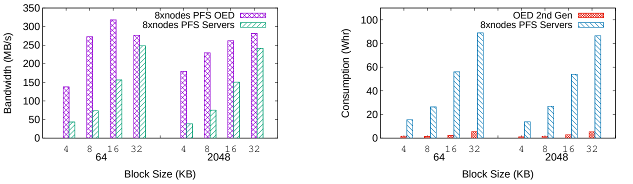
Breadth-First Search: Breadth-First Search is a graph traversal algorithm. It explores the vertices and edges of a graph beginning from a specified ”starting vertex”. It assigns each vertex a level number, starting with current and then visits its neighbors until it visits the entire graph. For out-of-core cases, the algorithm performs the exploration at one level at a time. The inputs to the application are graphs of 768 MB, 1536 MB, and 3072 MB of total size.



(a) Performance

(b) Energy Consumption

Fig. 7: OED as Parallel File System Servers (IOR-Read)



(a) Performance

(b) Energy Consumption

Fig. 8: OED as Parallel File System Servers (IOR-Write)

The same trend continues as it can be seen in figure 6(a) where the 2nd generation OED shows a big improvement over the 1st generation by performing the BFS algorithm **3x** faster. The server node completed the same test in 121 seconds which is 1.6x faster than the OED.

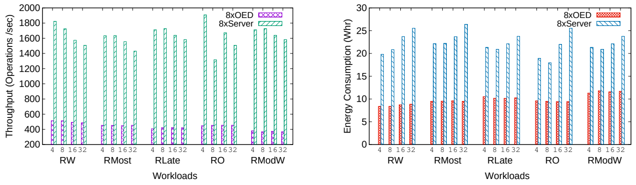
Figure 6(b) shows the energy consumption results. For the largest scale graph, 1st generation OED has a 2.16 Whr whereas the 2nd generation has 1.05 Whr. The server node comes last in the energy efficiency metric with only 4.86 Whr.

Concluding the above tests, we observe that the 2nd generation OED with its new hardware improvements sets a new trend where it demonstrates the best energy efficiency among all machines tested. It can perform computations faster than the 1st generation with a slight increase of the power consumption at full load (i.e., only 2 Watts more on average). The server node, even though the most powerful machine of all, it showed the highest power consumption and motivated us to explore the hypothesis further into a cluster environment.

C. Results from Clustered Environment

OED as Parallel File System Servers: Our first proposed use case for the OED technology in HPC is to utilize such devices as parallel file system servers. In this test, we aim to measure the aggregated bandwidth produced by a collection of OEDs and compare it with similar deployment in a cluster with typical server nodes. We run IOR with each process performing I/O of 2 GB. Block sizes range from small block size of 64 KB to 2 MB resembling typical workload from real applications. Figures 7(a) and 8(a) report the achieved bandwidth for read and write operations respectively. It can be seen that the bandwidth of the 2nd generation OED deployment is from 1.3x and up to **2.3x** higher than the bandwidth server nodes achieved. It is worthy to note that the highest bandwidth was achieved with 32 client processes since this test case saturates the storage layer.

Figures 7(b) and 8(b) show the energy efficiency score for both systems, read and write operations respectively. The



(a) Performance

(b) Energy Consumption

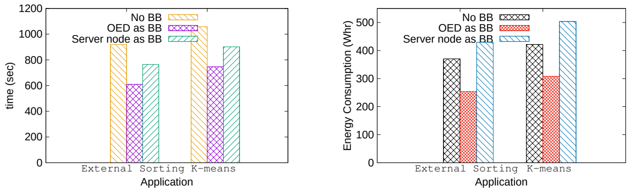
Fig. 9: OED as Key-Value Store Servers (YCSB)

low-powered nature of OED devices along with the better bandwidth that it demonstrated led to a much higher energy efficiency. Specifically, the power consumed by the entire collection of OEDs that run the PFS service is 154.6 Watts on full load which makes it **15x** more energy efficient than the typical server nodes.

OED as Key-Value Store Servers: Our second proposed use case for the OED technology in HPC is to utilize such devices as key-value store servers. In this test, we monitor the operation throughput of various workloads from YCSB as follows, a) Balanced: 50% reads and 50% writes. b) Read-mostly: 90% read and 10% writes c) Read-only: 100% read d) Read-latest: the most recently inserted records are the most popular. e) Read-modify-write: the client will read a record, modify it, and write back the changes. Figure 9(a) demonstrates the performance results for all the above workloads. We observe that server nodes are able to perform 3-4x faster than the OED. This is expected since the CPU on the server node is more powerful than the OED and Redis needs to calculate a hash value for each key in every operation.

In figure 9(b) we present the energy consumption results. Interestingly, even though the OED is slower than the server nodes, the small energy footprint allows them to still be more energy efficient with scores around 10 Whr on average between the tested workloads. The server nodes run for less seconds but consumed an order of magnitude more power and thus, achieved a 22 Whr average energy efficiency. Yet again, OEDs are more than **2.2x** more efficient.

OED as I/O Accelerators: In this last test we use external sorting and K-means clustering applications as our driver programs. We deploy a burst buffer system consisting of 4-OED devices or 4-server nodes. For both applications the flow is similar: reading input data from PFS, performing the algorithms (i.e., sorting or k-means), writing intermediate results, merging intermediate files, and writing the final output file to PFS. As a baseline, intermediate results are written/read to/from the remote PFS. In the burst buffer cases (noted in the graphs as type of device used followed by BB), all intermediate results are redirected to the buffer nodes that are physically closer to the computation nodes. The input size to both applications is 64 GB. In figure 10(a), the time is a compound time of all phases we mentioned above. We can see that for sorting, OEDs perform **20%** faster than the server nodes and that the existence of the BB layer boosts the overall performance by 33.5%. Similarly, for K-means OEDs are faster by **17%** compared to server nodes and 31% when compared with the baseline (i.e., no burst buffers).



(a) Performance

(b) Energy Consumption

Fig. 10: OED as I/O Accelerator

Finally, in figure 10(b) we can see the energy efficiency results. When we use OEDs as burst buffer nodes, the overall energy efficiency of the entire testbed is 254 Whr whereas the baseline is 370 Whr. Note that, when we use typical server machines as burst buffer nodes even though the overall execution time decreases by 17%, the energy efficiency of the entire testbed increased to 429 Whr since we introduced 4 extra server nodes as burst buffers.

V. Related Work

Idle Disk Spin Down: This is a technique used by most power saving models. The idea is simply reduce the spinning of the disk once it has been in idle mode for a while.There are several methods allowing disk to stay idle so that they can spin down. Massive Array of Idle Disk (MAID) [9] which employs disk cache with LRU policies. Hibernator models disk concentration according to popularity and variable disk speed to save power [11] where as popular data concentration collects popular data together to save power [10]. All these approaches have several drawbacks. (a) They mainly benefit those systems where they have rarely accessed data like archival; (b) to efficiently concentrate data they need to have an knowledge of the I/O patterns; and (c) Concentrating data has adverse effects on the performance as it reduces the bandwidth.

Exploiting redundancy: The diverted access uses replication [48]. It segregates all primary copies of data ona a partial set of disk to allow powering down of other disk. It shows the potential power gains but it can really hurt the performance of the system. Other works include study of using raid system to save power [49]. The solution applies to settings where the coding is limited to small groups of disks rather than a wider distribution of redundancy over a whole system.

ARM-based clusters: There are several works suggesting use of mobile devices in HPC which illustrate the potential of a mobile technology. The authors in [50] show an energy efficiency of 8.7x of their ARM-based cluster compared to a traditional cluster. The above work encourage us to suggest the use of the OED technology in HPC for power-efficient storage nodes.

VI. Conclusions

In this paper we presented the evolution of the Open Ethernet Drive technology through extensive evaluation. This new generation of the OED technology shows great advancement in hardware capabilities from its predecessor. Results suggest that OEDs can act as I/O accelerators and can replace typical storage nodes while maintaining normal operations. The overall energy efficiency of the OEDs is impressive and requires 2x-24x less energy for the use cases we tested. Additionally, the small form factor and compact enclosure as a JBOD are its unique characteristics. These mean that OEDs require far less space and hence lesser cooling needs which can lead to higher parallelism with lower power bills. As the ARM architecture progresses, the OED technology will have better computational capabilities which can lead to even higher performance while maintaining high energy efficiency. As a next step, we will compare OED technology with low-powered Xeon-based servers. We look forward to the next generation of such devices where a new storage hierarchy is said to be introduced. A small SSD drive on top of the HDD will give OEDs the ability to absorb incoming I/O faster while maintaining the core characteristics of the existing architecture.

References

[1] E. Russo, "Applying moores law to data growth," *Data Avail*, 2014.

[2] F. Y. Wang, K. M. Carley, D. Zeng, and W. Mao, "Social computing: From social informatics to social intelligence," *IEEE Intelligent Systems*, vol. 22, no. 2, pp. 79–83, March 2007.

[3] P. NYBERG, "The critical role of supercomputers in weather forecasting," 2013.

[4] R. S. J. McDermott, *Computational systems biology*. Springer, 2009.

[5] E. Riedel, G. A. Gibson, and C. Faloutsos, "Active storage for large-scale data mining and multimedia," in *Proceedings of the 24rd International Conference on Very Large Data Bases*, ser. VLDB '98. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998, pp. 62–73. [Online]. Available: http://dl.acm.org/citation.cfm?id=645924.671345

[6] A. Acharya, M. Uysal, and J. Saltz, "Active disks: Programming model, algorithms and evaluation," in *Proceedings of the Eighth International Conference on Architectural Support for Programming Languages and Operating Systems*, ser. ASPLOS VIII. New York, NY, USA: ACM, 1998, pp. 81–91. [Online]. Available: http://doi.acm.org/10.1145/291069.291026

[7] S. Mingay, "Green it: A new industry shock wave, gartner symposium/itxpo," *Gartner*, 2007.

[8] E. S. P. U.S. Environmental Protection Agency, "EPA Report on Server and Data Center Energy Efficiency," *Public Law 109-431*, 2007. [Online]. Available: http://www.energystar.gov

[9] D. Colarelli and D. Grunwald, "Massive arrays of idle disks for storage archives," in *Proceedings of the 2002 ACM/IEEE Conference on Supercomputing*, ser. SC '02. Los Alamitos, CA, USA: IEEE Computer Society Press, 2002, pp. 1–11. [Online]. Available: http://dl.acm.org/citation.cfm?id=762761.762819

[10] E. Pinheiro and R. Bianchini, "Energy conservation techniques for disk array-based servers," in *Proceedings of the 18th Annual International Conference on Supercomputing*, ser. ICS '04. New York, NY, USA: ACM, 2004, pp. 68–78. [Online]. Available: http://doi.acm.org/10.1145/1006209.1006220

[11] Q. Zhu, Z. Chen, L. Tan, Y. Zhou, K. Keeton, and J. Wilkes, "Hibernator: Helping disk arrays sleep through the winter," in *Proceedings of the Twentieth ACM Symposium on Operating Systems Principles*, ser. SOSP '05. New York, NY, USA: ACM, 2005, pp. 177–190. [Online]. Available: http://doi.acm.org/10.1145/1095810.1095828

[12] D. Narayanan, A. Donnelly, and A. Rowstron, "Write off-loading: Practical power management for enterprise storage," *Trans. Storage*, vol. 4, no. 3, pp. 10:1–10:23, Nov. 2008. [Online]. Available: http://doi.acm.org/10.1145/1416944.1416949

[13] M. W. Storer, K. M. Greenan, E. L. Miller, and K. Voruganti, "Pergamum: Replacing tape with energy efficient, reliable, disk-based archival storage," in *Proceedings of the 6th USENIX Conference on File and Storage Technologies*, ser. FAST'08. Berkeley, CA, USA: USENIX Association, 2008, pp. 1:1–1:16. [Online]. Available: http://dl.acm.org/citation.cfm?id=1364813.1364814

[14] A. Kougkas, A. Fleck, and X. H. Sun, "Towards energy efficient data management in hpc: The open ethernet drive approach," in *2016 1st Joint International Workshop on Parallel Data Storage and data Intensive Scalable Computing Systems (PDSW-DISCS)*, Nov 2016, pp. 43–48.

[15] HGST, "Openstack summit oed presentation," *Openstack*. [Online]. Available: http://www.slideshare.net/

[16] Y. Chen, C. Chen, X. H. Sun, W. D. Gropp, and R. Thakur, "A decoupled execution paradigm for data-intensive high-end computing," in *2012 IEEE International Conference on Cluster Computing*, Sept 2012, pp. 200–208.

[17] "Top 500, the list," https://www.top500.org/.

[18] "Green 500, the list," https://www.top500.org/green500/.

[19] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir, and H. Franke, "Reducing disk power consumption in servers with drpm," *Computer*, vol. 36, no. 12, pp. 59–66, Dec 2003.

[20] "Trinity." [Online]. Available: http://www.lanl.gov/projects/trinity/

[21] "Sequoia - advanced simulation and computing." [Online]. Available: https://asc.llnl.gov/computing_resources/sequoia/

[22] "Cray specs xc40." [Online]. Available: http://www.cray.com/sites/default/files/resources/CrayXC40Brochure.pdf

[23] "Cray sonexion 3000." [Online]. Available: http://www.cray.com/sites/default/files/SonexionBrochure.pdf

[24] "Bluegeneq." [Online]. Available: http://www.redbooks.ibm.com/redbooks/pdfs/sg247872.pdf

[25] N. Joukov and J. Sipek, "Greenfs: Making enterprise computers greener by protecting them better," *SIGOPS Oper. Syst. Rev.*, vol. 42, no. 4, pp. 69–80, Apr. 2008. [Online]. Available: http://doi.acm.org/10.1145/1357010.1352600

[26] J. G. Koomey, "Growth in data center electricity use 2005 to 2010." pp. 49, 24, 2011.

[27] P. Carns, K. Harms, W. Allcock, C. Bacon, S. Lang, R. Latham, and R. Ross, "Understanding and improving computational science storage access through continuous characterization," *Trans. Storage*, vol. 7, no. 3, pp. 8:1–8:26, Oct. 2011. [Online]. Available: http://doi.acm.org/10.1145/2027066.2027068

[28] Y. Kim, R. Gunasekaran, G. M. Shipman, D. A. Dillow, Z. Zhang, and B. W. Settlemyer, "Workload characterization of a leadership class storage cluster," in *Petascale Data Storage Workshop (PDSW), 2010 5th*, Nov 2010, pp. 1–5.

[29] "Kinetic open storage project," https://www.openkinetic.org/.

[30] "Openstack using oed architecture," http://goo.gl/P7u9e4.

[31] "Cloudian's hyperstore on oed architecture," http://goo.gl/eZOzsm.

[32] "Yahoo! cloud serving benchmark," https://github.com/brianfrankcooper/YCSB.

[33] "Skylablesx on oed architecture," http://www.skylable.com/pdf/hgst_use_case.pdf.

[34] "Stress-ng," http://kernel.ubuntu.com/~cking/stress-ng/.

[35] "Ior benchmark," https://goo.gl/YtW4NV.

[36] J. O'Reilly, "Future of the Ethernet drive," http://searchstorage.techtarget.com/feature/Future-of-the-Ethernet-drive-may-hinge-on-NVMe-over-Ethernet.

[37] Y. Kim, R. Gunasekaran, G. M. Shipman, D. Dillow, Z. Zhang, B. W. Settlemyer *et al.*, "Workload characterization of a leadership class storage cluster," in *Petascale Data Storage Workshop (PDSW), 2010 5th*. IEEE, 2010, pp. 1–5.

[38] N. Mi, A. Riska, Q. Zhang, E. Smirni, and E. Riedel, "Efficient management of idleness in storage systems," *ACM Transactions on Storage (TOS)*, vol. 5, no. 2, p. 4, 2009.

[39] "Leadership Computing Requirements for Computational Science," https://www.olcf.ornl.gov/wp-content/%20uploads/2010/03/ORNL%20TM-2007%2044.pdf.

[40] A. Kougkas, H. Eslami, X.-H. Sun, R. Thakur, and W. Gropp, "Rethinking keyvalue store for parallel i/o optimization," *The International Journal of High Performance Computing Applications*, vol. 31, no. 4, pp. 335–356, 2017. [Online]. Available: https://doi.org/10.1177/1094342016677084

[41] A. Kougkas, M. Dorier, R. Latham, R. Ross, and X.-H. Sun, "Leveraging burst buffer coordination to prevent i/o interference," in *e-Science (e-Science), 2016 IEEE 12th International Conference on*. IEEE, 2016, pp. 371–380.

[42] "HOBO Plug Load Data Logger UX 120," http://www.onsetcomp.com/products/data-loggers/ux120-018.

[43] "External Integer Sorting application," https://github.com/alveko/external_sort.

[44] "K-means clustering application," https://github.com/genbattle/dkm.

[45] "Graph500 - Breadth-first search (BFS) application," http://graph500.org/?page_id=12#sec-3_1.

[46] "OrangeFS parallel file system," http://www.orangefs.org/.

[47] "Redis key-value store," https://redis.io/.

[48] E. Pinheiro, R. Bianchini, and C. Dubnicki, "Exploiting redundancy to conserve energy in storage systems," in *Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems*, ser. SIGMETRICS '06/Performance '06. New York, NY, USA: ACM, 2006, pp. 15–26. [Online]. Available: http://doi.acm.org/10.1145/1140277.1140281

[49] D. Li and J. Wang, "Eeraid: Energy efficient redundant and inexpensive disk array," in *Proceedings of the 11th Workshop on ACM SIGOPS European Workshop*, ser. EW 11. New York, NY, USA: ACM, 2004. [Online]. Available: http://doi.acm.org/10.1145/1133572.1133577

[50] N. Rajovic, A. Rico, N. Puzovic, C. Adeniyi-Jones, and A. Ramirez, "Tibidabo: Making the case for an arm-based hpc system," *Future Generation Computer Systems*, vol. 36, pp. 322–334, 2014.