Towards Energy Efficient Data Management in HPC: The Open Ethernet Drive Approach

Anthony Kougkas, Anthony Fleck, Xian-He Sun
Illinois Institute of Technology, Department of Computer Science, Chicago, IL
{akougkas, afleck}@hawk.iit.edu, sun@iit.edu

***Abstract*—An Open Ethernet Drive (OED) is a new technology that encloses into a hard drive (HDD or SSD) a low-power processor, a fixed-size memory and an Ethernet card. In this study, we thoroughly evaluate the performance of such device and the energy requirements to operate it. The results show that first it is a viable solution to offload data-intensive computations on the OED while maintaining a reasonable performance, and second, the energy consumption savings from utilizing such technology are significant as it only consumes 10% of the power needed by a normal server node. We propose that by using OED devices as storage servers in HPC, we can run a reliable, scalable, cost and energy efficient storage solution.**

***Keywords*—Open Ethernet Drives, Data Management, Data-Intensive Computing, Energy Efficiency, Benchmarking, Performance Evaluation**

I. Introduction

Modern supercomputers capture, classify, analyze, process, and store data in an unprecedented rate. The amount of data is growing significantly faster than Moore's law [1]. This data explosion, driven by other advances in technology , gave birth to *data-intensive computing* making it the fourth paradigm towards scientific discovery [2]. Many fields such as astronomy, meteorology, social computing, bioinformatics, and computational biology have become highly data-driven [3], [4], [5]. It is clear that as we move to the exa-scale era, data management will be one of the greatest challenges. The evolution of software and hardware led system designers to new computer architectures to tackle this issue. ActiveDisks [6] and ActiveStorage [7] proposed to take advantage of the embedded processors on the storage servers. Burst Buffers [8] were introduced to quickly absorb bursty I/O from the compute nodes onto fast flash storage or NVRAM and asynchronously push them back to the archival storage system which in most cases is a parallel file system (PFS) such as PVFS [9], Lustre [10] or other. However, even though the performance and the efficiency of the systems was improved, most of these architectures rely on thousands of storage servers running on full-blown nodes as the backend storage solution. The energy consumption to support such data-intensive HPC workloads remains as high as ever and the related energy costs are a big concern. Department of Energy (DOE) High-Performance Computing (HPC) centers aim to keep system utilization as high as they possibly can, however, most data centers waste 90% of their power consumption because of the expensive data management. In fact, data centers worldwide consume roughly 30 GW [11] and it is projected to grow higher.

One new technology that could possibly alleviate the high energy consumption of data handling is the *Open Ethernet Drive* (OED) architecture. Initially designed for the cloud computing environment, OED architecture enables the migration of "data-centric" storage services as close to the storage as possible and it could potentially deliver improved storage efficiency, flexibility and enable new capabilities to the data center of the future [12]. An OED is an "intelligent" storage device that consists of a processor, RAM, Ethernet, and a hard drive or SSD. Two companies, Seagate and Western Digital's subsidiary HGST have developed and presented their prototypes with a few significant differences. Seagates Kinetic Open Storage platform [13] replaces that primitive block I/O interface with a key/value API and the traditional SAS or SATA interfaces with a pair of Gigabit Ethernet ports and clearly targets the cloud environment. HGST's implementation [14] could be characterized as more general as each OED comes loaded with a Linux OS offering developers a more open field by letting them run their own native code. OED architecture is based on the assumption that it is more cost effective to spread a workload like a storage system across thousands of low-cost processors (e.g. ARM-based) than to run it on a few more powerful server-graded CPUs (e.g. Intel Xeons or AMD Opterons). Despite the differences, OED architecture is not just using Ethernet as a new connection interface; it is also moving 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 usage of OEDs in HPC by evaluating the architecture, benchmarking the performance of an OED device, and conducting an energy cost analysis. We propose that a possible integration of such technology in the HPC infrastructure is meaningful in two aspects: *optimizing* the I/O performance and *reducing* the energy consumption. In terms of optimizing the I/O performance, OEDs can be used in multiple ways such as active I/O aggregators inside the compute cluster performing various administrative operations on the data (i.e. compression/decompression, deduplication, statistics e.t.c.), or as active burst buffers [15] and ActiveFlash arrays [16], [17], or even as specialized storage entities in architectures like Decoupled Execution Paradigm (DEP) [18], [19]. In terms of energy consumption, OEDs could reduce the energy required to deploy a PFS by entirely replacing storage servers. Since there is a Linux OS on each OED, one could easily deploy PFS servers on them maintaining the parallelism while also driving down the total energy consumption and most likely the monetary costs. In this paper we present all the necessary metrics to better understand the OED technology and

978-1-5090-5216-5/16 $31.00 © 2016 IEEE
DOI 10.1109/PDSW-DISCS.2016.11

we aim for setting the ground work for further exploration.

The rest of the paper is organized as follows. Section II presents the OED architecture and the technical specifications of a prototype OED device manufactured by HGST. Our evaluation methodology is presented in Section III and the results in Section IV. The conclusions of this study and the future work are presented in Section V.

II. Background and Motivation

A. Open Ethernet Drive Architecture

An OED device is designed to bring computation closer to the data by embedding an ARM-based processor and some RAM onto the drive itself. By connecting a number of OEDs together in some type of enclosure, a relatively capable cluster is created. Note that, the prototype device we study in this paper is by HGST's implementation and any details provided will refer specifically to this implementation. Other vendors might add or remove features and 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 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 and the rest is available to the 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 or configuring it in an enclosure. HGST has presented a 4U enclosure, called JBOD, that contains 60 such drives offering a 240 TB total storage capacity. This enclosure's components are hot-swap capable. The enclosure 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. Even though these hardware capabilities seem relatively lower compared to a modern HPC storage node, there are a lot of benefits from this architecture such as hardware costs, overall size, power and cooling consumption, and ease of maintenance.

B. Use Cases

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. Mirantis, a company that delivers all the software for running OpenStack, collaborated with HGST to demonstrate the deployment of various software-defined technologies on the OED architecture [20]. Specifically, they 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, a software-defined storage company famous for its HyperStore smart scale storage platform, successfully deployed HyperStore servers on top of the OED technology. This test aims to answer two fundamental questions: is this even feasible and if yes, how well would perform? They used Yahoo Cloud Serving Benchmark [21] to test the setup and concluded that all the tests were successful [22]. OED architecture could offer a lot of opportunities for optimizations on their applications. Finally, Skylable, a company whose mission is to build a fast, robust and cost-effective object-storage solution had the opportunity to experiment and deploy their Skylable SX services on top of HGST's OED technology [23]. According to the released report, after performing a series of tests from simple feasibility to resiliency and performance, they concluded that HGST's OEDs are the perfect building block for an energy efficient and horizontally scalable storage cluster running Skylable SX.

TABLE I: Hardware specifications

Feature	OED	Personal Computer	Server Node
CPU	ARM 32bit 1-core (1Ghz)	AMDAthlon X4 4-cores (3.7GHz)	2xAMD Opteron 8-cores (2.3GHz)
RAM	2GB DDR3 1600Mhz	16GB DDR3 2400Mhz	8GB DDR2 667Mhz
Disk	Megascale DC4000.B 4TB 7200rpm	Seagate Barracuda 1TB 7200rpm	WD 250GB 7200rpm
Network	1 Gbit/s	1 Gbit/s	1 Gbit/s
OS	Debian 8.0	Ubuntu 14.04	Ubuntu server 9.04
Kernel	3.14.3	4.4.0-34	2.6.28
Year	2014	2015	2009

All of the above mentioned use cases of the OED technology concern cloud environments and specifically object store services with Amazon's S3 API. We believe that HPC environment could similarly benefit from the usage of OED technology by deploying PFS servers on top of it. Applications could also leverage the embedded resources and perform in-situ data analysis with lower cost [24]. In the next section, we present our evaluation methodology of the prototype OED device. Our goal is to examine if such technology is capable enough for the high-end computing.

III. Methodology

The evaluation of the OED device is focused into two major categories: performance and energy consumption. To measure the device's performance we divided our tests into four major aspects: CPU, memory, disk, and network card performance. For measuring the energy consumption, we used a *wattmeter*, a special instrument for measuring the supply rate of electrical energy.

Hardware used: The prototype OED device implemented by HGST was compared with a common-use commercial computer and also with a server node, part of a 65-node SUN Fire Linux cluster at Illinois Institute of Technology (IIT). The hardware specifications for all machines used are shown in Table I.

Software used: A combination of our own micro-benchmarks and some well-know open-sourced benchmarks were executed on all machines tested. Specifically, for CPU testing we used SysBench [25], a modular, cross-platform and

multi-threaded benchmark tool for evaluating OS parameters that are important for a system running a database under intensive load and Stress-ng [26], a suite designed to stress test a computer system with its various physical subsystems as well as the various OS kernel interfaces. It has a wide range of CPU specific stress tests that exercise floating point, integer, bit manipulation and control flow. Stress-ng was also used to test the performance of the main memory along with PMBW benchmark [27]. For testing the I/O capabilities we used both Stress-ng and SysBench benchmark suites with the appropriate kernels. Finally, for network performance we used Iperf [28], an open-sourced tool for active measurements of the maximum achievable bandwidth on IP networks and aslo the Stress-ng network module. Lastly, we tested all machines with three real applications: an out-of-core sorting algorithm, a vector addition, and a kernel of calculating descriptive statistics of a given input [29].

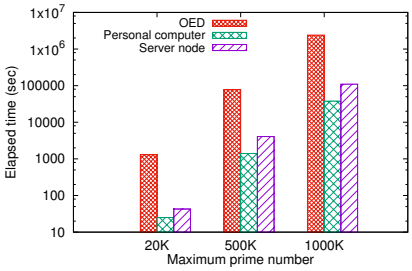
Setup: We restricted the available memory of the personal computer and the server node to the size of OED's RAM (i.e., 2 GB). We used three different dataset sizes, 1GB (half the available RAM), 2GB (equal to RAM size), and 4GB for out of core computations. We ran all the tests single-threaded on one CPU core since the OED has only one core. All machines were connected to the same network, IIT's internal and external network. Note that for clarity of the presented figures in the next section, some graphs are in logarithmic scale. In such case, it is reported in the figure description accordingly. Lastly, all experiments were executed 5 times and we report the average.

IV. Evaluation Results

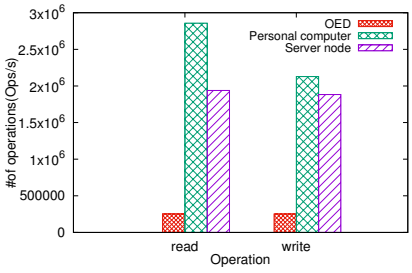
A. Benchmarks

CPU: In these tests, the clock speed and the CPU generation are important factors that determine how fast a processor completes a certain computation. Figure 1 (a) demonstrates the total time that each system needed to calculate all prime numbers up to a given threshold (i.e., 20000, 500000, and 1000000). It is clear that the OED has the weakest processor and it needed approximately 50x more time when compared to the personal computer (PC) and 30x time to the server node. In figure 2 we present the Stress-ng results for all the CPU-related modules spanning from binary search, context switching, spinning on sqrt(rand()), matrix and vector operations, and quick sort to CPU cache, and others. The lower CPU clock frequency and the ARM architecture are responsible for the significantly lower performance of the OED which performs 16x on average and up to 50x slower than the PC and 8.5x on average and up to 24x slower than the server node.

Memory: Figure 1 (b) shows the total number of memory read and write operations each system was able to perform in the specified time window. The PC was able to perform 11x more reads and 8x more writes than the OED. The server node was also faster by 7x even though the RAM modules specification are lower than those of the OED. This is caused by the OED's lower processor clock speed and the fewer number of memory bus lanes. We ran PMBW benchmark that tests two very basic functions found in any data processing: sequential scanning and pure random access. Table II summarizes the



(a) CPU prime number calculations (logscale)



(b) Main memory read/write

Fig. 1: SysBench results.

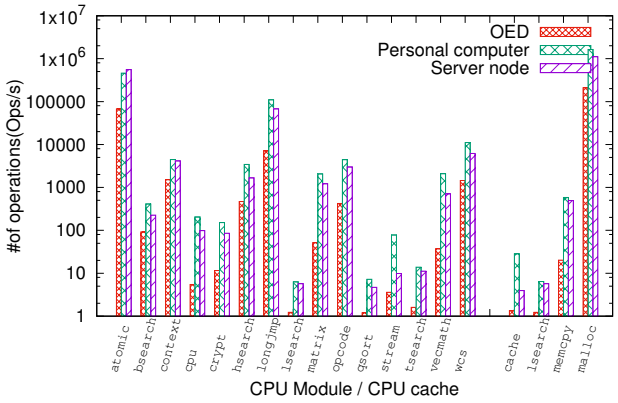


Fig. 2: CPU performance with Stress-ng (logscale).

results. OED is again slower compared to both the PC and the server node. In figure 3 we present Stress-ng results with various memory stressors such as malloc, memcpy, mmap, and remap. OED results are, on average, 12x lower than the PC and 5x compared to the server node.

Disk: In terms of pure disk performance the OED performs better from both other systems. In figure 4 (a) we present SysBench results. The disk bandwidth for this test was 3.38 MB/s for the OED, 0.98 MB/s for the PC, and 0.75 MB/s for the server node. The OED also performed 3.5x and 4.5x more operations and requests per minute when compared to

TABLE II: PMBW benchmark results

Memory Results	OED	Personal Computer	Server Node
max bandwidth	8 GiB/s	60 GiB/s	35 GiB/s
average bandwidth	4.2 GiB/s	24 GiB/s	8.9 GiB/s
min latency	0.5 ns	0.2 ns	0.3 ns
average latency	3.5 ns	2.1 ns	2.5 ns

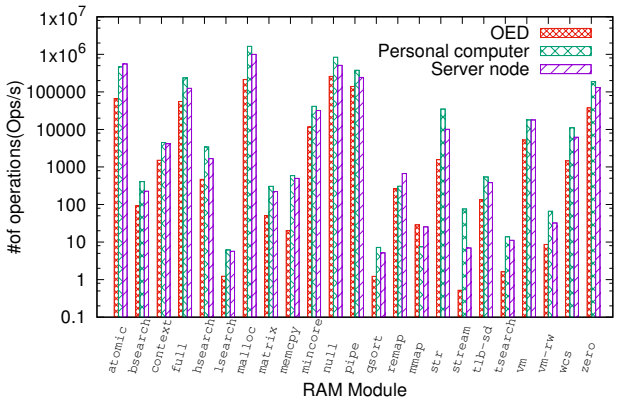
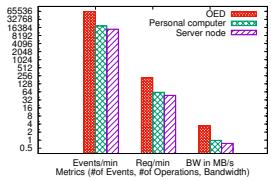


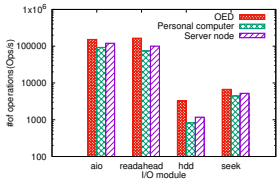
Fig. 3: Main memory performance with Stress-ng (logscale).

the PC and server node respectively. The above mentioned numbers might seem very low for a disk however, this test demands from the disk to react to hundreds of thousands of events and respond to hundreds of requests thus stressing the disk to its limits. Figure 4 (b) shows the I/O capabilities of each system in terms of operations per second as Stress-ng reports. OED performs on average 2.3x and 1.7x faster than the PC and server node respectively. Overall, OED hard drive is more capable and it might be the advanced disk technology that HGST impemented on this drive that is responsible for this result. The PC has an off-the-shelf commercial disk and the server node is equipped with a server-graded disk that is more than 7 years old.

Network: Since the OED is equipped with an onboard Ethernet card we ran a few network-related tests. In figure 5 (a) we report the network card performance through Stress-ng benchmark. We observe that the PC's Ethernet card is from 2-6x faster than the OED's and the server node's is from about 1-4x faster. The network cards' specifications are similar but the smaller form factor of the OED might have restricted any internal buffers size. In the next test, we had all systems connected to the same external line to the Internet through IIT's network and we tested the popular network benchmark Iperf. We performed the tests in early morning hours to minimize congestion of the network and we killed all unnecessary processes that might have used network resources. We varied the TCP window size as a quick optimization. OED's network performance was again lower than the other two systems as it can be seen in figure 5 (b).

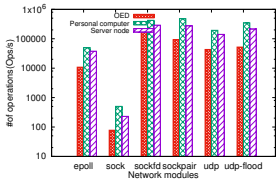


(a) Disk performance with SysBench (logscale).

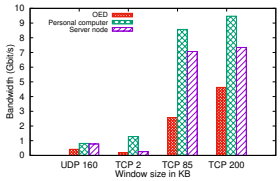


(b) I/O characteristics with Stress-ng (logscale).

Fig. 4: Disk and I/O benchmarks.



(a) Network performance with Stress-ng (logscale).



(b) Network capabilities (Iperf)

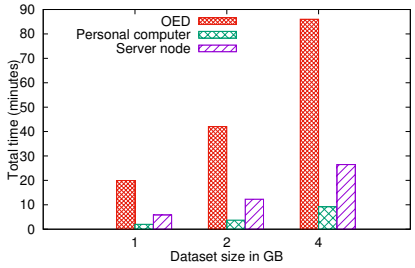
Fig. 5: Network benchmarks.

B. Real applications

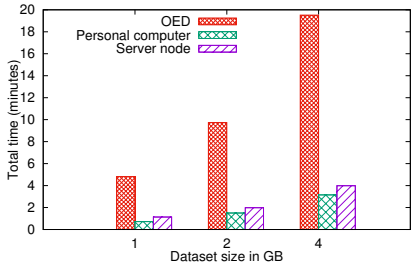
We tested the OED device at executing real application's code. We created three datasets of 1, 2, and 4 GB of random integers. The applications we ran are out-of-core sorting, vector addition, and descriptive statistics. Figure 6 (a) shows the result for sorting. We see that the OED took 9x more time than the PC and 3x more time than the server node. The OED performance is limited considering the weaker individual components, specifically the CPU. Nonetheless, OED completed the test successfully. In figure 6 (b) we present the results for vector addition. In this simple application, two vectors are read from the disk and are added together to produce a new vector. OED's performance is consistent but still 6x and 4x slower than the PC and server node respectively. Finally, results from the descriptive statistics application are shown in figure 6 (c). This application computes and prints summary statistics (count, first quartile, mean, median, third quartile, variance, interquartile range, standard deviation, min, max) for a given data set. It is therefore more cpu-intensive and the results prove that. OED took about 9x and 7x more time to complete the test compared to the PC and server node.

C. Energy consumption

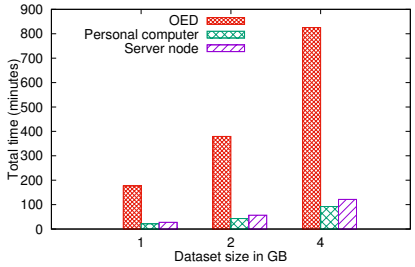
So far we presented our evaluation of the OED device in terms of pure performance in the internal components and the performance while running several applications. In this subsection we study a very important aspect of computing: power consumption. Performance in computer systems comes with a cost of high energy consumption for powering high-performance hardware and for cooling the system down. OED



(a) Sorting



(b) Vector addition



(c) Descriptive statistics

Fig. 6: Real applications' results

comes with a 3.5" form factor which helps in two ways, size and minimal need for cooling. Additionally, the internal hardware components are extremely power friendly which was one of the initial goals in designing such a device. We measured the power consumption (W) with a wattmeter in various stages: booting up, staying idle, performing integer sorting, and simple disk-related operations such as moving data by copying. Figure 7 demonstrates the results. When booting up, the OED consumes just 20 W while the PC and server node consume 175 W and 200 W respectively. When the system is idle OED consumes 16 W and while running sorting goes up to 16.6 W. That is significantly lower compared to the 165 W that the server node needed to run sorting. Copying data results are consistent with the trend that OED is consuming approximately 10% of the energy needed by a normal computer system.

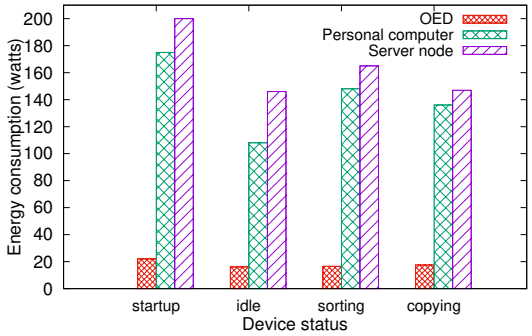


Fig. 7: Energy consumption comparison

V. Conclusions and Future Work

In this paper, we evaluated the capabilities of the Open Ethernet Drive architecture and explored a potential integration in the HPC hardware stack. We first presented the OED architecture and then we comprehensively benchmarked a prototype device. The results indicate that the performance of such a device is not yet on par with a normal server node but are limited to this specific hardware implementation. An interesting finding of this study is that such a device can perform computations while consuming a very small fraction of energy when compared to a normal server. This will allow system deployments that require less energy to maintain normal operations since OEDs do not need the same cooling degree and the device itself consumes less energy to power its hardware components.

As future work, we plan to do a scalability study. We have already successfully installed MPICH 3.2 and OrangeFS 2.9.5 (i.e., PVFS2) on the OED and we plan to further expand our testing with MPI benchmarks and applications. We plan to deploy a parallel file system on multiple OEDs and compare it to a deployment of the same PFS over cluster server nodes. We already have access to a 2nd generation of OEDs by HGST which are more powerful and thus, we will monitor how this technology is evolving. Finally, we will model the OED's performance and we will implement a simulator that we plan to open source and let others use it in their system architecture designs.

As the computation capability of OED increases, we can leverage the technology by installing OED JBODs close to the compute cluster to act as active burst buffer nodes. Applications can offload some of the data-intensive computations on the OEDs in a decoupled execution fashion. Asynchronicity and non-blocking I/O will boost OEDs performance while consuming much less energy. In a world where mobile computing becomes more powerful every six months, where smartphones employ 8-core processors running at 2.5 Ghz clock speed, where Ethernet networks become faster and faster, we strongly believe that the OED technology will push the boundaries of running an HPC system requiring less power and possibly smaller monetary costs for deploying an energy efficient data management solution.

ACKNOWLEDGMENT

The authors would like to acknowledge Los Alamos National Laboratory for providing the prototype OED hardware and also for giving them access to a x60-OED JBOD located at LANL. Specifically, we want to thank Dr. Hsing Bung Chen from LANL and Dr. Fu Song from University of North Texas for providing us helpful instructions about the OED devices. Lastly, we want to thank PhD student Kun Feng from our lab in Illinois Tech for helping setting up the network for our tests.

REFERENCES

[1] P. Ranganathan, "The data explosion," 2011.

[2] T. Hey, S. Tansley, K. M. Tolle *et al.*, *The fourth paradigm: data-intensive scientific discovery*. Microsoft research Redmond, WA, 2009, vol. 1.

[3] 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, 2007.

[4] A. Lesk, *Introduction to bioinformatics*. Oxford University Press, 2013.

[5] J. McDermott, R. Samudrala, R. Bumgarner, K. Montgomery, and R. Ireton, *Computational systems biology*. Springer, 2009.

[6] A. Acharya, M. Uysal, and J. Saltz, "Active disks: Programming model, algorithms and evaluation," *ACM SIGPLAN Notices*, vol. 33, no. 11, pp. 81–91, 1998.

[7] E. Riedel, G. Gibson, and C. Faloutsos, "Active storage for large-scale data mining and multimedia applications," in *Proceedings of 24th Conference on Very Large Databases*. Citeseer, 1998, pp. 62–73.

[8] "Large Memory Appliance/Burst Buffers Use Case," https://asc.llnl.gov/CORAL-benchmarks/Large_memory_use_cases_llnl.pdf.

[9] P. H. Carns, W. B. Ligon, III, R. B. Ross, and R. Thakur, "PVFS: a Parallel File System for Linux Clusters," in *Proceedings of the 4th annual Linux Showcase & Conference - Volume 4*. Berkeley, CA: USENIX Association, 2000.

[10] S. Donovan, G. Huizenga, A. J. Hutton, C. C. Ross, M. K. Petersen, and P. Schwan, "Lustre: Building a File System for 1000-Node Clusters," Citeseer, 2003.

[11] W. Harrod, "A journey to exascale computing," 2012.

[12] "Openstack summit oed presentation," http://www.slideshare.net/hgststorage/open-stack-summit-presentation.

[13] "Hgst's open ethernet drive architecture," https://www.hgst.com/company/innovation-center.

[14] "Seagates kinetic open storage platform," http://goo.gl/2iCg5O.

[15] C. Chen, M. Lang, L. Ionkov, and Y. Chen, "Active burst-buffer: In-transit processing integrated into hierarchical storage."

[16] H. Sim, Y. Kim, S. S. Vazhkudai, D. Tiwari, A. Anwar, A. R. Butt, and L. Ramakrishnan, "Analyzethis: an analysis workflow-aware storage system," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. ACM, 2015, p. 20.

[17] S. Boboila, Y. Kim, S. S. Vazhkudai, P. Desnoyers, and G. M. Shipman, "Active flash: Out-of-core data analytics on flash storage," in *012 IEEE 28th Symposium on Mass Storage Systems and Technologies (MSST)*. IEEE, 2012, pp. 1–12.

[18] 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*. IEEE, 2012, pp. 200–208.

[19] H. Eslami, A. Kougkas, M. Kotsifakou, T. Kasampalis, K. Feng, Y. Lu, W. Gropp, X.-H. Sun, and T. R. Chen, Yong, "Efficient disk-to-disk sorting: a case study in the decoupled execution paradigm," in *Proceedings of the 2015 International Workshop on Data-Intensive Scalable Computing Systems*, 2015.

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

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

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

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

[24] K. W. D. Z. T. L. I. R. Iman Sadooghi, Geet Kumar, "Albatross: an efficient cloud-enabled task scheduling and execution framework using distributed message queues." in *Proceedings of the IEEE 12th International Conference on eScience*. IEEE, 2016, p. 10.

[25] "Sysbench benchmark," https://launchpad.net/sysbench.

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

[27] "Pmbw memory benchmark," https://panthema.net/2013/pmbw/index.html#top.

[28] "Iperf network benchmark," https://iperf.fr/.

[29] "Descriptive statistics benchmark," https://github.com/loicseguin/desc.