Green Storage: Parallel File Systems on ARM

Timm Leon Erxleben*, Kira Duwe*, Jens Saak†, Martin Köhler† and Michael Kuhn*
*Otto von Guericke University Magdeburg
Magdeburg, Germany
E-mail: timm.erxleben@ovgu.de, kira.duwe@ovgu.de, michael.kuhn@ovgu.de
†Max Planck Institute for Dynamics of Complex Technical Systems
Magdeburg, Germany
E-mail: saak@mpi-magdeburg.mpg.de, koehlerm@mpi-magdeburg.mpg.de

***Abstract*—Parallel distributed file systems are typically run on dedicated storage servers that clients connect to via the network. Regular x86 servers provide high computational power, often not required for storage management and handling I/O requests. Therefore, storage servers often use low core counts but still have a relatively high idle power consumption. This leads to high energy consumption, even for mostly idle file systems. Advanced Reduced Instruction Set Computer Machines (ARM) systems are very energy-efficient but still provide adequate performance for file system use cases. Leveraging this fact, we built an ARM-based storage system, on which we tested different parallel distributed file systems. We compare the performance and energy efficiency of x86 and ARM systems using several metrics. Analysis of the different file systems on the ARM system shows that energy efficiency highly depends on the architecture and the used file system. Results show that while our ARM-based approach currently provides less throughput per Watt for reads, it achieves an approximately 174 % higher write efficiency when compared to a traditional x86 Ceph cluster.**

Keywords*—*energy efficiency; parallel distributed file systems; x86; ARM

I. Introduction

Storage systems are scaled up steadily to satisfy increasing storage demands, leading to growing energy consumption [2]. High-Performance Computing (HPC) storage systems are currently built from regular x86 servers, whose computing power is not fully utilized by storage applications. Traditional x86 servers feature a relatively high power consumption even when idle: It is not uncommon to measure idle consumption of more than 100 W for just the processor, main memory, and mainboard. In comparison, low-power ARM computers are often required to stay below 5–10 W maximum consumption by design. To offset the high idle consumption of x86 servers, they have to be equipped with large amounts of storage devices, such as hard disk drives (HDDs) and solid-state disks (SSDs). However, depending on the used network interconnect, only a limited number of devices can be saturated. For instance, on a 100 Gbit/s network, two to three Non-Volatile Memory Express (NVMe) SSDs are enough to provide the necessary throughput and more devices cannot be used to their full extent. This proportion gets even worse on slower networks.

Therefore, we evaluate the use of low-energy ARM-based single-board computers (SBCs) as a replacement for traditional servers in storage systems. To assess the feasibility of an ARM-based storage system, we evaluated the ARM-based cluster using CephFS, OrangeFS, MooseFS and GlusterFS. We compared the performance and energy efficiency of the different configurations to an OrangeFS test cluster at the University of Hamburg, using different metrics and workloads. Furthermore, we compared it to a productive CephFS cluster running at the computer science faculty of the Otto von Guericke University Magdeburg, to validate the approach against modern hardware.

The contributions of our paper are:

1) We propose to apply the energy-delay product, typically used to evaluate the energy efficiency of computations, as a metric for storage systems as well to measure energy efficiency while still accounting for the performance needed by HPC applications.
2) We show that low-power ARM-based storage clusters can achieve throughput efficiencies comparable to, or even exceeding, traditional x86 systems.

This paper is based on a previous conference paper [1]. Since then, we analyzed and tested two additional file systems, MooseFS and GlusterFS, and interpreted the measured power consumption with respect to performance and energy efficiency. We also included one additional cluster with different hardware characteristics in our evaluations.

The remainder of the paper is organized as follows. In Section II, the used file systems are briefly described followed by a summary of related works in Section III. Section IV describes the benchmarks which were done and discusses metrics that can be derived from the measurement data. Next, in Section V all cluster setups, ARM and x86, are described, followed by the presentation of the results. Results and setups are discussed in Section VI. Section VII concludes the paper.

II. Background

This section introduces background on used technologies, such as the used parallel file systems. The information is taken from the respective file system's documentation if not referenced otherwise.

A. Ceph

Ceph [3] is a popular, clustered object store, which is highly scalable due to its Controlled Replication Under Scalable Hashing (CRUSH) placement algorithm, which enables all participating services, that can access the cluster map to locate and place objects [4]. A typical Ceph cluster is made of Object

Storage Devices (OSDs), monitoring and management services. All components may be redundant to enable automatic failover.

Apart from access through the library `librados`, many interfaces might be used. The POSIX access via CephFS, realized by additional Metadata Services (MDSs) interacting with Ceph storage pools, is particularly interesting for HPC systems. This is because POSIX is often used by scientific high-level I/O libraries like HDF5 or NetCDF and ensures portability of many applications. Even so, its semantics are mostly too strict for typical HPC I/O requirements and can impair performance [5]. CephFS has a rich feature set, including replication, multiple storage pools, file systems, snapshots, and high control over data placement.

B. OrangeFS

OrangeFS is a traditional parallel distributed file system designed for HPC [6][7]. Only one type of server is needed, which can handle both data and metadata, though it can be configured to handle only one type.

In OrangeFS, data is striped according to a distribution function that can be specified for each file. The default is to start at a random server and use all servers in a round-robin fashion with a stripe size of 64 KiB. Unlike Ceph, which uses its own object store *Bluestore* [8], OrangeFS relies on a separate local file system.

As of the current version, 2.9.8, there are no redundancy features for data that is not marked as read-only, though this is planned for OrangeFS version 3 [9]. Many interfaces may be used to interact with OrangeFS. Most popular choices include access via the OrangeFS Linux kernel module or direct access using the library `libpvfs2`. Noteworthy is the direct Message Passing Interface I/O (MPI-IO) support by using ROMIO's [10] Abstract-Device Interface for I/O (ADIO), for which OrangeFS provides an implementation [11].

C. MooseFS

MooseFS [12] is a POSIX-compliant parallel distributed file system designed for Big Data applications [13]. It makes use of different server types for metadata and data storage, monitoring and metadata backups.

Metadata is managed using the so-called master server for any accesses and several metalogger servers for backup purposes. Unlike in the other used parallel file systems, metadata is completely held in memory. Persistency is guaranteed by periodic on-disk backups and an on-disk journal for modifying operations.

Data is striped with a hard-coded size of 64 MiB and distributed to the chunk servers, which provide persistent storage using the underlying local file system. The distribution is random but prioritizes chunk servers with a lower load [14]. For data safety, each file has a replication goal.

D. GlusterFS

GlusterFS [15] is a parallel distributed file system for cloud storage and media streaming. Like OrangeFS, GlusterFS has only one type of server. All GlusterFS severs form a Trusted Storage Pool (TSP), which provides attached storage, called bricks, for volumes. Each volume creates its own namespace that clients can mount. Multiple volumes of different types may be created on top of a TSP.

The different volume types specify the distribution of data in the cluster. Distributed and replicated volumes distribute files without striping and are therefore not suitable for HPC applications. However, dispersed volumes make use of striping and provide data safety via redundant data blocks using erasure coding. This type of volume provides the parallel access needed to satisfy the performance requirements of parallel applications. The block size depends on the number of storage servers and the ratio of redundant blocks.

In GlusterFS, no separate metadata handling is needed because all participants can determine file positions by hashing. Unix file metadata, like access time or permissions, is stored in the inodes of the underlying file system. Other GlusterFS-specific metadata is stored using extended file attributes. In dispersed volumes, the metadata is duplicated to each file fragment, which contains all blocks of that file on this server. These file fragments are stored as a regular file on the servers.

Though GlusterFS is not specifically designed for HPC applications the use of erasure coding via dispersed volumes was interesting for the comparison with the other file systems.

III. State of the Art and Related Work

There have been various endeavors to measure and increase the energy efficiency of large systems, as energy consumption is becoming a possible constraint on HPC systems in the future. Many different aspects have to be considered, ranging from the system's energy efficiency to the scalability of the applications. As ARM processors aim to offer better energy efficiency, they have been heavily studied across the years [16–18]. Deployments, such as Fugaku [19], show that they can provide competitive performance and even work in exascale systems. Earlier research on systems like Tibidabo at Barcelona Supercomputing Center indicated that single instruction, multiple data stream (SIMD) instructions limited to single precision were a severe bottleneck for the performance [17][18][20].

Energy efficiency is also a relevant aspect in distributed systems, as examined for peer-to-peer systems. A survey by Brienza et al. [21] showed that often simple energy models were used, disregarding other hardware components like intermediate routers. An early approach, and still very prominent solution to energy savings in storage, is sending idle peers to sleep [22]. However, it introduces problems when the load varies. To have systems benefit from the increased energy efficiency, in the long run, applications have to be considered as well. The optimization towards energy efficiency comes indeed with its challenges for applications [20][23–25]. Reducing the performance of a single core, in order to cap the power consumption, means that scalability is of increased importance [20].

Gudu and Hardt evaluated the use of an ARM-based Ceph cluster, made of Cubieboards, as a replacement for traditional network-attached storage (NAS) controllers [26]. They measured the throughput of their cluster via Ceph's Reliable Autonomic Distributed Object store (RADOS) and RADOS Block Device (RBD) access and found that the Cubieboard cluster is a viable alternative to NAS controllers. However, the limited network capabilities were the bottleneck of the system.

Apart from using low-power hardware [27], there have been efforts to reduce the power consumption of existing HPC storage clusters [28][29]. For example, it was proposed to assign subsets of storage clusters to individual users and only run a specific subset at full power when an assigned user uses the compute-cluster [30].

Considering that local file systems are often part of the storage stack, their influence on energy efficiency and performance were analyzed, in [31], using simulated workloads of web, database, and file servers. It was found that the choice of file system and its configuration greatly influence performance and energy efficiency. However, no file system performed best for all workloads.

In contrast to Gudu and Hardt, we measure data throughput at the CephFS level and evaluate ARM-based clusters as a replacement for HPC storage clusters.

IV. Benchmark and Metrics

We measured the performance of the clusters for data-throughput oriented workloads. The benchmark comprised sequential, independent accesses from one to four clients using IOR v3.3 [32] with the POSIX backend, individual files per client and five iterations for each data point. The transfer size was set to 4 MiB, which corresponds to the default stripe size of CephFS and is aligned to the stripe size of the other parallel file systems. On the x86-based Ceph cluster, 96 GiB were written and read. The amount of data was reduced to 36 GiB for the other two clusters to keep run-times manageable.

For every iteration of the measurements, the power consumption of the storage cluster was measured using the methods as described in Section V. As a result, several energy efficiency metrics can be derived from the collected data. However, choosing a specific metric is not trivial, as there is no single optimal metric indicating energy efficiency [33].

We decided to compare the results obtained by using the **energy-delay product** (EDP) [34], **throughput per Watt** and **capacity per Watt** [35].

Throughput per Watt is a commonly used metric for evaluating and comparing storage energy efficiency. The transferred data may differ between systems, so it is well suited to compare systems that greatly vary in their performance. However, this metric alone is insufficient when analyzing and optimizing storage systems, as no insight into performance is given. Geveler et al. [23] found that for simulations, in some cases, energy savings might lead to performance drops. In such cases, they motivated using the EDP as a fused metric describing energy efficiency and performance at once. The EDP is computed as the product of the total energy E consumed while performing a task and the time t needed to complete the task (Equation (1)). Depending on the performance requirements, the time may be weighted [36]. As we want to focus on energy consumption, we set $w = 1$.

$$\text{EDP} = E \cdot t^w, \quad w \in \mathbb{N} \tag{1}$$

Though the energy-delay product was initially developed for hardware design, it is also useful when evaluating software, as done by Georgiou et al. [37]. Nevertheless, the amount of work needs to stay constant to compare different systems, so only the two ARM setups are compared using the EDP. Because its unit is hard to interpret and even changes with different weights, we normalized the EDP using the lowest value per comparison.

The third metric considered measures the capacity of the storage system per Watt. Because of growing storage demands and, therefore, growing storage systems, optimizing systems regarding this metric is critical for the cost-efficient and environmentally friendly operation of data centers.

V. Evaluation

In this section, the hardware and software setup is described, followed by an analysis of the respective clusters' theoretical peak performance and the presentation of the results.

A. Reference Cluster 1

The first reference cluster is a five node subset of a research cluster at the University of Hamburg. Each node has two Intel Xeon X5650 CPUs, each featuring six cores at 2.67 GHz, 11 GB RAM, and two Intel 82574L Gigabit Network Interface Cards (NICs). One node is equipped with a 250 GB Western Digital WD2502ABYS HDD [38], while the other nodes are equipped with a 250 GB Seagate ST3250318AS HDD [39]. A ZES Zimmer LMG 450 power meter was used to measure the power consumption of this setup. The five nodes consumed **460.21 W** on average in idle state with a standard deviation of 18.43 W, measured over one hour, with HDDs spun up. The clients used to benchmark this reference cluster were four servers of the same specification.

We used OrangeFS version 2.9.8 as file system for its straightforward setup and good comparability to the ARM-cluster. One node was used exclusively for metadata storage, while the other four nodes provided data storage. The used block size was the default of 64 KiB. The same configuration of OrangeFS was later used on the ARM-based cluster.

B. Reference Cluster 2

The second reference cluster is a four-node subset of the productive Ceph cluster running at the computer science faculty at the Otto von Guericke University using Ceph 16.2.7 deployed as containers. Three nodes of the subset are part of the Supermicro AS 2124BT-HNTR [40] multi-node system, each of which is equipped with four Intel P4510 NVMe SSDs [41]. The fourth server is a Gigabyte R282-Z94 [42] equipped with one Intel P4510 NVMe SSD and eight Samsung

MZQL23T8HCJS-00A07 NVMe SSDs [43]. All nodes are connected by 100 Gbit Ethernet, with a separate 100 Gbit network for communication between Ceph OSDs. Though Ceph does not exclusively use the nodes, they are idle most of the time. The average idle power consumption of the four nodes was measured to be **699.3 W**. This power measurement was done on a Sunday since the servers are mostly idle on the weekend. It lasted for one hour, starting at 14:00, and had a standard deviation of 13.98 W. While running, the benchmark power consumption peaked at 1,057 W. The existing monitoring solution, gathering power samples over IPMI every 15 seconds, was used to collect power samples.

For each SSD, two Ceph OSDs are deployed. The Ceph monitor and a standby metadata service are located at the Gigabyte server, while the active metadata service runs on one of the Supermicro servers. Ceph pools use the default replication settings and, therefore, produce three replicas of the data and return to the client after two replicas are written. The clients used for the benchmark were four servers equipped with an AMD Epyc 7443, with 24 cores at 2.85 GHz, 128 GB RAM, and 100 Gbit Ethernet.

C. ARM Cluster

a) Cluster Setup: The low-power cluster is built of six Odroid HC4 nodes featuring the Amlogic S905X3 SoC, with four cores at 1.8 GHz, 4 GiB DDR4 RAM, two SATA-3 ports, and a 1 Gbit NIC [44] (see Figure 1). We decided to use the Odroid HC4 instead of more typical SBCs like the Raspberry Pi [45] due to its native SATA ports. Four of the nodes, nodes A1–A4, are equipped with two 1 TB WD Black HDDs [46] and one, node C, is equipped with two 512 GB Samsung V-NAND SSD 860 PRO SSDs [47]. One exception was made for GlusterFS, where the SSDs were exchanged with two more HDDs. This decision will be discussed later on. The last remaining node, node B, has no disks and is intended for monitoring purposes. All nodes are connected to a Netgear GS110EMX switch [48].

The ARM-cluster nodes run on Armbian Buster 21.08.8, which uses Linux 5.10.81-meson64. Armbian [49] is a Linux distribution based on Debian, which is modified and optimized for use on SBCs. The pre-installed *Petitboot* was erased from the HC4's flash memory to use the *uboot* bootloader, which is part of the Armbian image.

The complete cluster, including the switch, is powered by an MW HRP450-15 PSU [50] and consumes 56.36 W, measured over one hour with a standard deviation of 0.14 W, in idle state, with HDDs spun up. For comparison with the reference cluster, which does not include the switch in the power measurements, we subtracted the average idle power of the switch, which was measured to be 15.46 W, with a standard deviation of 1.13 W over one hour. The adjusted idle power consumption of the ARM cluster, therefore, is **40.9 W**. The highest peak in power consumption measured while running the benchmark was 63.68 W, which was observed for writes with four clients and MooseFS.

For power measurements, the ZES Zimmer LMG 450 [51] was used to measure the power consumption of the PSU for the whole cluster. The power meter was connected to a BananaPi M1 via USB, which collects samples with 20 Hz.

The clients used to perform the benchmark were four Dell Precision 3650 Tower workstations [52] each with an Intel Core i7-11700 CPU with 8 cores at 2.5 GHz, 8 GB RAM, and a 1 Gbit NIC. They were connected via the network infrastructure of the Max Planck Institute Magdeburg.

We used Debian Bullseye 5.10.46-5 on the clients, which uses Linux 5.10.0-8. OpenMPI 4.1.0 with the included MPI-I/O implementation OMPIO was installed from the Debian Buster Repository for parallel benchmarks. All storage nodes and clients use Network Time Protocol (NTP) to synchronize their clocks with node C.

The network topology is visualized in Figure 1, where dotted lines depict the devices included in the power measurement.

b) Parallel File System Configuration: We compared four different parallel file systems on the ARM cluster: CephFS, OrangeFS, MooseFS and GlusterFS. Though not all of them are originally designed for the same purpose and workloads, they can be configured to perform reasonably well for the parallel coordinated access. The different architectures and features of the file systems were useful to determine the capabilities of the ARM-based cluster.

We used Ceph version 14.2.21, which is available in the Buster backports repository. One OSD was deployed for each storage device. Node C, which was equipped with SSDs, additionally ran one MDS. The Ceph monitor and management daemon ran on node B, which has no disks attached. The two storage pools needed for CephFS used different CRUSH rules to distribute objects. While the data pool used all HDDs and managed replicas on the node level, the metadata pool used the two SSDs and managed replicas on the OSD level. Both pools were configured to use 64 placement groups, to produce two replicas and to return immediately after one replica is written. The relaxed replication settings allowed a fairer comparison with the other file systems.

We built OrangeFS version 2.9.8 with GCC version 8.3.0 and LMDB 0.9.22 from the Buster repository. As explained above, OrangeFS has only a single type of daemon, which was running on all nodes with disks. Metadata was stored by the daemon, which was deployed on node C, while the other nodes stored the data. As OrangeFS offers no data redundancy for data that is not read-only, ZFS version 2.0.3 was used to mirror disks locally.

We used MooseFS version 3.0.115, which is available in the Buster backports repository. The chunk servers were deployed on nodes A1–A4, the master server on node C and the monitoring server on node B. As file deletes are always asynchronous in MooseFS, deleted files will be removed in the background after a certain time, called trash time. We made sure to avoid overlapping reads and writes with background file deletions by keeping the default trash time of 24 hours and timed our benchmarks accordingly.

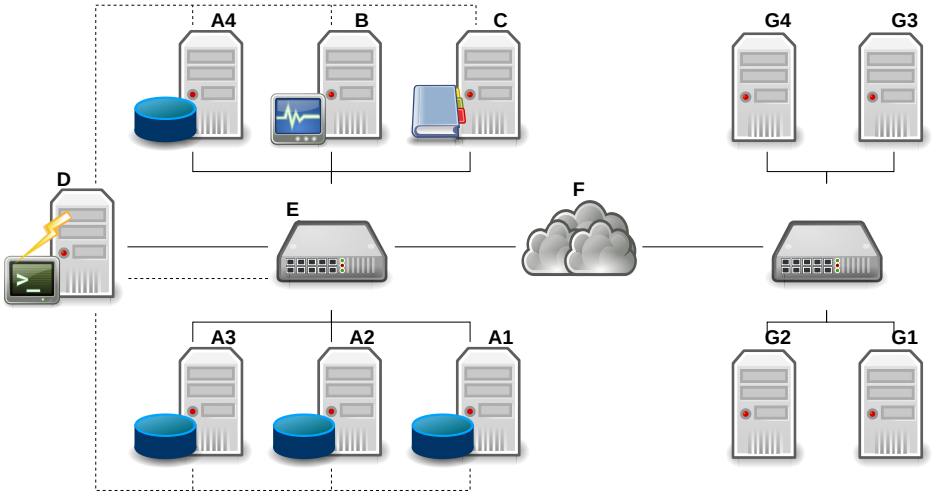

Figure 1. This graphic shows the network topology of the ARM-based cluster. The storage nodes (A1–A4), the management node (B), the metadata node (C), and the BananaPi with the power meter (D) are connected to the Netgear switch (E). The dotted lines indicate which devices are included in power measurements. The storage cluster is connected to the clients (G1–G4) via the Max Planck Institute Magdeburg network infrastructure (F).

We built GlusterFS version 9.2 with GCC version 8.3.0. All nodes with disks were part of a trusted storage pool. We used one disk as brick per node, which was formatted with XFS according to the recommendations from the GlusterFS documentation [15]. Additionally, we exchanged the SSDs on node C for HDDs of the same type as on nodes A1–A4, as GlusterFS cannot benefit from multiple storage tiers within a single volume. The dispersed volume was created using all nodes in the TSP and a redundancy count of one, resulting in a stripe size of 2 KiB. These changes for GlusterFS did not change the theoretical peak performance, which is then bound by the network throughput of the clients.

D. Theoretical Peak Performance

As can be seen in Table I, the theoretical peak performance (TPP) of the ARM cluster is limited by the network throughput of each node, which is not as high as the aggregated throughput of all storage devices of the node. The same applies to reference cluster 1.

Because no measurements could be made in the productive reference cluster (reference cluster 2), the maximum throughput of the components is taken from the respective datasheets. Adding together the TPP of the two node types, its TPP is **44.04 GiB/s**.

This analysis neglects metadata operations, which are, reasonably, assumed not to limit the data throughput of the cluster, for a few files in use. Furthermore, the table only presents the performance for writes. However, as the network already limits peak performance for the ARM cluster, and aggregated throughput of the SSDs in Supermicro nodes of the reference cluster is close to the network speed, the same applies, approximately, to reads.

E. Results

The results of the performance and energy efficiency metrics are shown in Figures 2 to 4. Each measurement was repeated five times and the error bars depict the standard deviation of those samples. The samples for the throughput per Watt metric are computed by dividing each performance measurement by the mean power consumption of its measurement iteration. As explained above, the EDP is normalized by the lowest value per comparison.

Figure 5 shows box plots of the distributions of measured power samples during the last measurement iteration with 4 clients on the ARM cluster, to gain further insight into the power consumption of the different parallel file systems.

The capacity metric shown in Figure 6 was computed using the idle power consumption of the clusters and the raw storage capacity. Nevertheless, the usable storage capacity depends on the respective software setup. The ARM cluster achieved **0.178 TiB/W** and the productive reference cluster (reference 2) **0.066 TiB/W**. Reference cluster 1 is excluded from this metric, because it is not designed as a storage cluster and therefore not equipped with many high capacity storage devices.

VI. Discussion

In this section, we discuss general aspects of our experiments followed by a discussion on the results.

All results need to be seen in relation to the respective systems' cost, as the ARM cluster nodes and disks cost only about € 1,350, while the reference cluster nodes and disks cost around € 40,000. In addition, the reference cluster 2 only uses NVMe SSDs, while the ARM-based cluster uses HDDs for data object storage. Due to the low sampling rate of the power measurements for the reference cluster 2, some spikes in the energy consumption are possibly missed, resulting in an underestimation. In contrast, power measurements on the ARM-based cluster can be expected to overestimate the actual power consumption of the nodes and disks, as only the average idle power consumption of the switch is subtracted.

During previous experiments on a BananaPi M1 single-board computer cluster, the deployment of traditional parallel

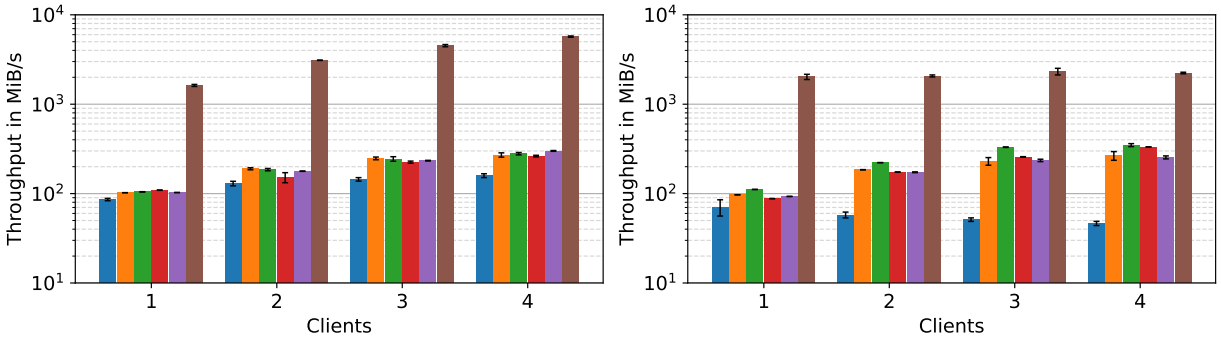

Figure 2. Throughput for reading (left) and writing (right)

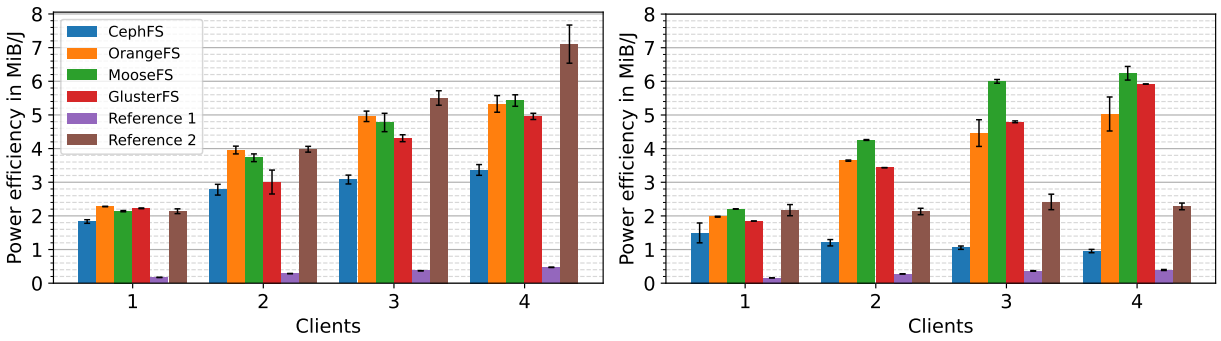

Figure 3. Power efficiency in throughput per Watt for reading (left) and writing (right)

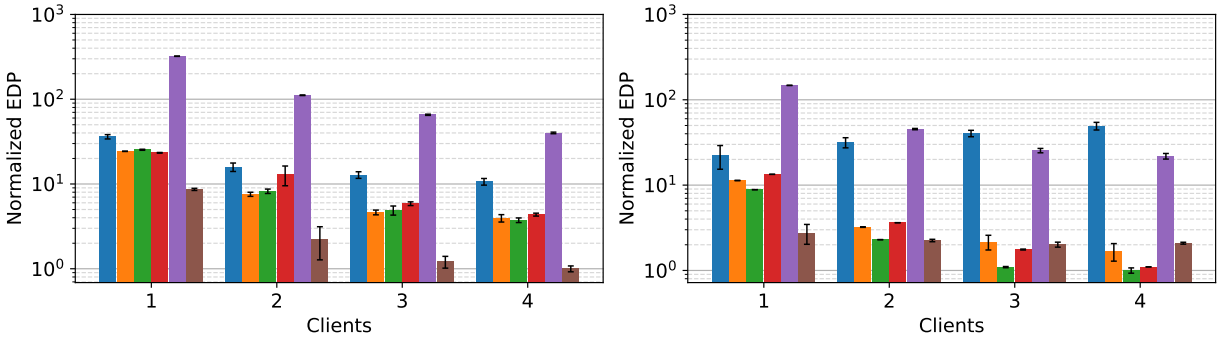

Figure 4. Normalized energy-delay product for reading (left) and writing (right)

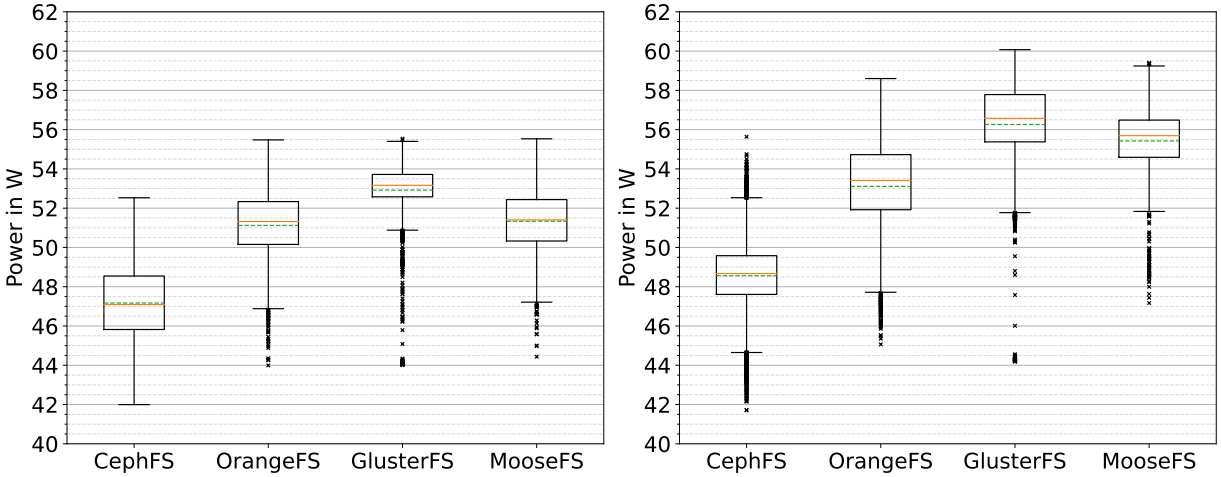

Figure 5. Distribution of power samples for the last measurement iteration with 4 clients for reading (left) and writing (right). In addition to the quartiles, the mean is depicted as green dashed line.

TABLE I. THROUGHPUT OF COMPONENTS RELEVANT FOR THEORETICAL PEAK PERFORMANCE (TPP) THROUGHPUT

Cluster	Network	Throughput Storage Devices	Storage Devices per Node	# Nodes	TPP
ARM	111.34 MiB/s	140.09 MiB/s	2	4	445.36 MiB/s
Reference 1	112.22 MiB/s	115.35 MiB/s	1	4	448.88 MiB/s
Reference 2 - Supermicro	11.64 GiB/s	2.70 GiB/s	4	3	32.4 GiB/s
Reference 2 - Gigabyte	11.64 GiB/s	2.70 GiB/s / 3.73 GiB/s	1+8	1	11.64 GiB/s

TABLE II. MAXIMUM THROUGHPUT OF ALL MEASUREMENT ITERATIONS IN MIB/S AND PERCENT OF TPP.

System	Write / % TPP	Read / % TPP
ARM - CephFS	95.22 / 21.38	172.12 / 38.65
ARM - OrangeFS	289.23 / 64.94	296.82 / 66.65
ARM - MooseFS	365.57 / 82.08	291.41 / 65.43
ARM - GlusterFS	333.26 / 74.83	268.60 / 50.31
Reference 1	266.48 / 59.37	305.52 / 68.06
Reference 2	2322.47 / 5.15	5705.00 / 12.65

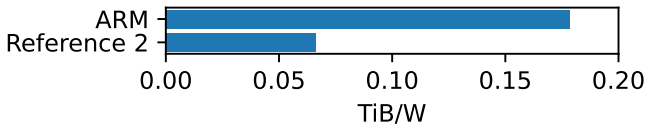

Figure 6. Storage capacity per Watt

file systems proved difficult on the unusual hardware. Tested file systems were CephFS, OrangeFS and BeeGFS. Both CephFS and BeeGFS needed small patches to run on the setup. OrangeFS could not run the client on ARM 32-bit using the upstream kernel module. Additionally, we observed low read throughput if no direct I/O was used. For four clients reading a 2 GiB file each, only 12.41 MiB/s could be achieved. Consequently, measurements on OrangeFS are done with direct I/O.

Our prototype cannot compete with the throughput of the productive reference cluster 2. However, using reference cluster 1 the performance of the ARM-based cluster seems comparable. We used these two references for different purposes. While reference cluster 1 has nearly the same TPP, and is therefore good to compare the performance of the different file systems on different architectures, it is made of legacy hardware and not built as a storage cluster. Because of this, it is not beneficial as a reference for the energy efficiency. Consequently, we used the second reference cluster, which is made of modern hardware, to validate the ARM-based cluster in terms of energy efficiency. For real world HPC applications, more storage nodes would need to be added to the ARM-based cluster to achieve higher throughput. This cluster was built as a proof-of-concept for throughput efficiency and to gain insight in ARM single-board computer storage clusters.

The different read and write sizes on both setups were chosen to achieve reasonable run-times of the benchmarks on both settings. Neither throughput nor throughput efficiency are influenced by the different amounts of transferred data if run-times are long enough.

A. Performance

All clusters show good throughput scaling when adding more clients. Exceptions occur for writes. On the second reference cluster, one client achieves close to the observed maximum performance, and no further improvement can be seen when adding more clients. On the CephFS setup, on the ARM-based cluster, the situation is even worse, as performance drops for more clients. Both Ceph-based systems only reached a fraction of the theoretical peak performance, as can be seen in Table II. For the ARM cluster, this is most likely related to data replication over the public network. This hypothesis is supported by the fact that Ceph OSDs reported slow operation warnings due to waiting times for sub-operations. As pointed out by Just [53], the Ceph OSD service utilizes many threads, which could lead to performance issues for a few cores, as context switches introduce additional overhead. Ceph's behaviour is strongly influenced by the number of placement groups per OSD [3]. While a higher ratio of placement groups to OSDs ensures a balanced data distribution, management of each placement group consumes memory and CPU time. To minimize overhead, we set both pools to 64 placement groups. The number of placement groups per OSD also influences recovery behavior for larger clusters as more placement groups need to be replicated in case of a server crash. Further experiments are needed to evaluate different placement group counts and placement group to OSD ratios for productive usage of Ceph on large ARM clusters.

Nevertheless, replication cannot explain the performance drop for the second reference cluster, which needs further investigation. One impacting factor for reads was that only one process per client was used, resulting in only one network stream, insufficient to saturate the network. This decision was made for comparability with the ARM cluster.

Both Ceph-based systems might be impacted by CephFS' lazy deletes [3], which are done asynchronously by an MDS and probably overlapped with reads and writes, resulting in lower throughput.

OrangeFS performs better than CephFS on ARM in nearly all measurements. In contrast to CephFS, the OrangeFS daemon is lightweight and does not use many threads. Therefore, context switches introduce less overhead on low core counts. Because no replication is done between nodes, less data needs to be transferred via the network, and the management of replicas does not consume resources. The downside is that faults of nodes can lead to data loss. Even though performance is higher compared to CephFS, only about 60 % of the TPP (see Table II) can be achieved.

MooseFS behaved similar to OrangeFS overall. However, it achieved the maximum of write throughput of all measurements at 82.08% of TPP, see Table II. Nevertheless, a disadvantage from the perspective of a single user might be the asynchronous deletion of files in the background, because

this overlapping workload reduces the performance of parallel I/O. Different parameters on how background operations are performed are available in MooseFS. Further tuning of the system is needed to show if this problem can be mitigated.

Although GlusterFS is not designed for these parallel coordinated accesses its overall performance was comparable to OrangeFS on the ARM-based cluster. Its performance does not seem to be impaired by the small stripe size of only 2 KiB. However, it also had an advantage compared to the other file systems because one additional server was available for data storage resulting in lower I/O stress per node. On the other hand, due to redundant data blocks added by erasure coding, each node received the same amount of data to write as with OrangeFS and MooseFS.

While most of its volume types are not suitable for HPC workloads, dispersed volumes enable parallel access to multiple servers within one file and ensure data safety using erasure coding. In contrast to replication, erasure coding will not duplicate all of the data blocks, but add redundant blocks. Thus, it is more space efficient and puts less stress on the network than replication.

The disadvantage of this volume type in GlusterFS is that the stripe size depends on the number of bricks in the volume and the desired redundancy count. Scaling up such systems, without changing the stripe size, can be accomplished by combining volume types. Multiple dispersed volumes can be part of a single distributed volume, which would distribute whole files to the different dispersed volumes.

All tested file systems on the ARM cluster can certainly be tuned for higher throughput. Many settings of different storage layers influence their behavior. On top of that, the interactions between the layers are non-trivial. For example, let us look at OrangeFS: Tuning the stripe size and the record size of ZFS can be a first optimization. Compared to the defaults of other parallel file systems, OrangeFS has a relatively low default stripe size of 64 KiB. Further benchmarks should be done to evaluate bigger stripes, which could result in larger disk accesses depending on server-side cache size and cache times. As shown by traces of MPI-IO calls and OrangeFS' internal Trove layer, which does the actual disk I/O, single client-side write calls can result in multiple server-side Trove write calls [54]. Those should align to ZFS record sizes, if possible, to minimize read-modify-write cycles. Additionally other local file systems (e.g., XFS, BTRFS) and layers for local disk mirroring (e.g., LVM, MDADM) could be evaluated.

B. Energy Efficiency

While the ARM-based cluster was hardly comparable with the second reference cluster in terms of performance, its energy efficiency and throughput per Watt was similar to or even exceeded the second reference for all file systems except CephFS. In contrast, the first reference cluster shows devastating energy efficiency. The reason for this is the high energy consumption while only operating on a 1 Gbit/s network and therefore showing similar performance to the ARM-based cluster.

Apart from the first reference cluster, the energy efficiency plots resemble the performance plots in the relations between the file systems. This similarity suggests that energy efficiency on the ARM-based cluster was mostly determined by performance and resulting run-times of the operations.

To determine whether there are more differences between the systems than performance and resulting run-times, we took a look at the measured power consumption during the last measurement iteration with 4 clients, which can be seen in Figure 5. File systems that showed a higher throughput and energy efficiency, also consumed more power during the benchmarks. This pattern indicates that a higher hardware utilization leads to better performance and ultimately higher energy efficiency. CephsFS, for example, was possibly not able to fully utilize the disks on the nodes, due to the OSDs' overhead, which led to lower power consumption of disks, but also to lower throughput. GlusterFS, on the other hand, had the highest power consumption, which, in combination with its high throughput, suggests a high hardware utilization. Even so, GlusterFS' power consumption is slightly higher, here, because two SSDs were swapped for HDDs.

Overall, the ARM cluster's low idle power and maximum power consumption allow for usage of the cluster in places or situations where power restrictions apply, enabling the usage as a mobile storage solution.

In terms of capacity per Watt, the ARM cluster is superior to the second reference cluster, achieving 2.68 times more TB per Watt. However, this result could easily be changed by using higher capacity disks on both clusters. For a more sophisticated comparison between the system architectures in this regard, the power consumption of the server nodes should be measured separated from the disks as done by Gudu and Hardt [26] resulting in a storage controller energy efficiency metric. Nevertheless, this metric is useful for optimizing existing storage solutions.

C. Energy-Delay Product

Compared to the other metrics, the EDP, as shown in Figure 4 is a fused metric that measures performance and energy efficiency at once. The use of this metric for tuning storage systems enforces that balanced configurations are found. Neither performance nor energy-saving efforts are neglected in favor of the other. One example is given by the first reference cluster and CephFS on the ARM-based cluster, see Figure 4. While CephFS had low performance but also a low power consumption its EDP is lower at first. Nevertheless, with more clients the first reference cluster starts to gain advantage because performance starts to out weigh energy consumption.

Even so, for practical applications the weight of the EDP has to be chosen carefully. To evaluate HPC applications one would likely choose higher weights to put more focus on performance. Another problem is imposed by great fluctuations of the measured EDP for repeated measurements, which are even amplified when a larger weight is chosen. This could be mitigated by using shorter benchmarks and more repetitions.

VII. Conclusion and Future Work

We evaluated different file systems for HPC workloads on two reference clusters, based on traditional x86 servers, and an ARM-based low-power cluster. We compared the results in terms of throughput and efficiency. The ARM cluster is able to provide more than twice as much TB per Watt compared to the reference cluster and can achieve similar throughput efficiency. OrangeFS, MooseFS and GlusterFS have been shown to perform better than CephFS on the ARM cluster. Due to the low idle power consumption and low power peaks, ARM-based storage solutions are helpful in situations where power restrictions apply, for example, when used as a mobile storage cluster. In summary, we have shown that the energy efficiency of storage solutions depends significantly on both the used architecture and the file system. Lightweight solutions can reduce energy consumption and thus cost, which is becoming increasingly important due to the exponentially growing volumes of data.

As a next step, we will evaluate the ARM-based cluster using other workloads that are of interest. Examples of such workloads are metadata-focused workloads and mixed workloads that would be produced by multiple users accessing the storage cluster in an uncoordinated manner. Such workloads will show whether ARM-based storage clusters with many small nodes can generally replace traditional storage clusters, or whether they are more suitable for smaller or special purpose systems. For this reason, throughput scaling of the ARM cluster while adding more storage nodes also needs to be measured.

Acknowledgment

This work is partly funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 417705296. More information about the CoSEMoS (Coupled Storage System for Efficient Management of Self-Describing Data Formats) project can be found at https://cosemos.de.

References

[1] T. L. Erxleben, K. Duwe, J. Saak, M. Köhler, and M. Kuhn, "Energy efficiency of parallel file systems on an ARM cluster," in *ENERGY 2022, The Twelfth International Conference on Smart Grids, Green Communications and IT Energy-aware Technologies*, vol. 12. IARIA, 2022, pp. 42–48.

[2] J. G. Koomey, "Worldwide electricity used in data centers," *Environmental Research Letters*, vol. 3, no. 3, 2008, DOI: 10.1088/1748-9326/3/3/034008.

[3] Ceph authors and contributors, "Ceph Documentation," https://docs.ceph.com/en/latest, 2021, [retrieved: 04, 2022].

[4] S. A. Weil, S. A. Brandt, E. L. Miller, D. D. E. Long, and C. Maltzahn, "Ceph: A Scalable, High-Performance Distributed File System," in *7th Symposium on Operating Systems Design and Implementation (OSDI '06), November 6-8, Seattle, WA, USA*, B. N. Bershad and J. C. Mogul, Eds. USENIX Association, 2006, pp. 307–320.

[5] C. Wang, K. Mohror, and M. Snir, "File system semantics requirements of hpc applications," in *Proceedings of the 30th International Symposium on High-Performance Parallel and Distributed Computing*, ser. HPDC '21. New York, NY, USA: Association for Computing Machinery, 2021, pp. 19–30, DOI: 10.1145/3431379.3460637.

[6] M. M. D. Bonnie *et al.*, "OrangeFS: Advancing PVFS," in *USENIX Conference on File and Storage Technologies (FAST)*, 2011.

[7] OrangeFS Development Team, "OrangeFS Documentation," http://docs.orangefs.com/, [retrieved: 05, 2022].

[8] K. Duwe and M. Kuhn, "Using Ceph's BlueStore as Object Storage in HPC Storage Framework," in *CHEOPS@EuroSys'21*. ACM, 2021, pp. 3:1–3:6, DOI: 10.1145/3439839.3458734.

[9] J. Edge, "The OrangeFS distributed filesystem," https://lwn.net/Articles/643165/, 2015, [retrieved: 04, 2022].

[10] R. Thakur, W. Gropp, and E. Lusk, "A Case for Using MPI's Derived Datatypes to Improve I/O Performance," in *Proceedings of SC98: High Performance Networking and Computing*. ACM Press, November 1998, DOI: 10.1109/SC.1998.10006.

[11] M. Vilayannur, R. Ross, P. Carns, R. Thakur, A. Sivasubramaniam, and M. Kandemir, "On the performance of the POSIX I/O interface to PVFS," in *12th Euromicro Conference on Parallel, Distributed and Network-Based Processing, 2004. Proceedings.*, 2004, pp. 332–339, DOI: 10.1109/EMPDP.2004.1271463.

[12] A. Kruszona-Zawadzka, "MooseFS 3.0 User's Manual," https://moosefs.com/Content/Downloads/moosefs-3-0-users-manual.pdf, 2017, [retrieved: 04, 2022].

[13] Tappest sp. z o.o., "MooseFS Website," https://moosefs.com, 2022, [retrieved: 05, 2022].

[14] Z. Baojun, P. Ruifang, and Y. Fujun, "Analyzing and improving load balancing algorithm of MooseFS," *International Journal of Grid and Distributed Computing*, vol. 7, no. 4, pp. 169–176, Aug. 2014, DOI: 10.14257/ijgdc.2014.7.4.16.

[15] GlusterFS Development Team, "GlusterFS Documentation," http://docs.gluster.org/, [retrieved: 05, 2022].

[16] Z. Ou, B. Pang, Y. Deng, J. K. Nurminen, A. Ylä-Jääski, and P. Hui, "Energy- and Cost-Efficiency Analysis of ARM-Based Clusters," in *Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012)*. IEEE Computer Society, 2012, pp. 115–123, DOI: 10.1109/CCGrid.2012.84.

[17] E. L. Padoin, D. A. G. de Oliveira, P. Velho, and P. O. A. Navaux, "Evaluating Performance and Energy on ARM-based Clusters for High Performance Computing," in *41st International Conference on Parallel Processing Workshops, ICPPW 2012, Pittsburgh, PA, USA, September 10-13, 2012*. IEEE Computer Society, 2012, pp. 165–172, DOI: 10.1109/ICPPW.2012.21.

[18] N. Rajovic, A. Rico, N. Puzovic, C. Adeniyi-Jones, and A. Ramírez, "Tibidabo: Making the case for an

ARM-based HPC system," *Future Generation Computer Systems*, vol. 36, pp. 322–334, 2014, DOI: 10.1016/j.future.2013.07.013.

[19] M. Sato *et al.*, "Co-Design for A64FX Manycore Processor and "Fugaku"," in *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, 2020, pp. 1–15, DOI: 10.1109/SC41405.2020.00051.

[20] D. Göddeke *et al.*, "Energy efficiency vs. performance of the numerical solution of PDEs: An application study on a low-power ARM-based cluster," *Journal of Computational Physics*, vol. 237, pp. 132–150, 2013, DOI: 10.1016/j.jcp.2012.11.031.

[21] S. Brienza, S. E. Cebeci, S. S. Masoumzadeh, H. Hlavacs, Ö. Özkasap, and G. Anastasi, "A Survey on Energy Efficiency in P2P Systems: File Distribution, Content Streaming, and Epidemics," *ACM Computing Surveys*, vol. 48, no. 3, pp. 36:1–36:37, 2016, DOI: 10.1145/2835374.

[22] G. Lefebvre and M. J. Feeley, "Energy efficient peer-to-peer storage," Technical Report TR-2003-17. Department of Computer Science, University of British Columbia, Tech. Rep., 2000.

[23] M. Geveler, B. Reuter, V. Aizinger, D. Göddeke, and S. Turek, "Energy efficiency of the simulation of three-dimensional coastal ocean circulation on modern commodity and mobile processors," *Computer Science - Research and Development*, vol. 31, no. 4, pp. 225–234, 2016, DOI: 10.1007/s00450-016-0324-5.

[24] F. Mantovani *et al.*, "Performance and energy consumption of hpc workloads on a cluster based on arm thunderx2 cpu," *Future Generation Computer Systems*, vol. 112, pp. 800–818, 2020, DOI: 10.1016/j.future.2020.06.033.

[25] M. Ponce *et al.*, "Deploying a top-100 supercomputer for large parallel workloads: The niagara supercomputer," in *Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (Learning)*, ser. PEARC '19. New York, NY, USA: Association for Computing Machinery, 2019, DOI: 10.1145/3332186.3332195.

[26] D. Gudu and M. Hardt, "ARM Cluster for Performant and Energy-Efficient Storage," in *Computational Sustainability*, ser. Studies in Computational Intelligence, J. Lässig, K. Kersting, and K. Morik, Eds. Springer, 2016, vol. 645, pp. 265–276, DOI: 10.1007/978-3-319-31858-5_12.

[27] 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)*, 2016, pp. 43–48, DOI: 10.1109/PDSW-DISCS.2016.012.

[28] L. Zhang, Y. Deng, W. Zhu, J. Zhou, and F. Wang, "Skewly replicating hot data to construct a power-efficient storage cluster," *Journal of Network and Computer Applications*, vol. 50, pp. 168–179, 2015, DOI: 10.1016/j.jnca.2014.06.005.

[29] X. Ruan *et al.*, "ECOS: An energy-efficient cluster storage system," in *2009 IEEE 28th International Performance Computing and Communications Conference*, 2009, pp. 79–86, DOI: 10.1109/PCCC.2009.5403814.

[30] C. Karakoyunlu and J. A. Chandy, "Techniques for an energy aware parallel file system," in *2012 International Green Computing Conference, IGCC 2012, San Jose, CA, USA, June 4-8, 2012*. IEEE Computer Society, 2012, pp. 1–5, DOI: 10.1109/IGCC.2012.6322247.

[31] P. Sehgal, V. Tarasov, and E. Zadok, "Evaluating Performance and Energy in File System Server Workloads," in *8th USENIX Conference on File and Storage Technologies, San Jose, CA, USA, February 23-26, 2010*, R. C. Burns and K. Keeton, Eds. USENIX, 2010, pp. 253–266.

[32] H. Shan and J. Shalf, "Using IOR to Analyze the I/O Performance for HPC Platforms," in *In: Cray User Group Conference (CUG'07)*, 2007.

[33] S. Rivoire, M. A. Shah, P. Ranganathan, C. Kozyrakis, and J. Meza, "Models and metrics to enable energy-efficiency optimizations," *Computer*, vol. 40, no. 12, pp. 39–48, 2007, DOI: 10.1109/MC.2007.436.

[34] M. Horowitz, T. Indermaur, and R. Gonzalez, "Low-power digital design," in *Proceedings of 1994 IEEE Symposium on Low Power Electronics*, 1994, pp. 8–11, DOI: 10.1109/LPE.1994.573184.

[35] D. Chen *et al.*, "Usage centric green performance indicators," *SIGMETRICS Perform. Evaluation Rev.*, vol. 39, no. 3, pp. 92–96, 2011, DOI: 10.1145/2160803.2160868.

[36] J. H. Laros III *et al.*, "Energy Delay Product," in *Energy-Efficient High Performance Computing: Measurement and Tuning*. London: Springer London, 2013, pp. 51–55, DOI: 10.1007/978-1-4471-4492-2_8.

[37] S. Georgiou, M. Kechagia, P. Louridas, and D. Spinellis, "What Are Your Programming Language's Energy-Delay Implications?" in *Proceedings of the 15th International Conference on Mining Software Repositories*, ser. MSR '18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 303–313, DOI: 10.1145/3196398.3196414.

[38] Western Digital Corporation, "WD2502ABYS Datasheet," https://products.wdc.com/library/SpecSheet/ENG/2879-701281.pdf?_ga=2.204934934.742886585.1651008103-1759292557.1651008103, 2008, [retrieved: 04, 2022].

[39] Seagate Technology LLC, "Seagte ST3250318AS Product Manual," https://www.seagate.com/staticfiles/support/disc/manuals/desktop/Barracuda%207200.12/100529369c.pdf, 2009, [retrieved: 04, 2022].

[40] Super Micro Computer, Inc., "Supermicro AS 2124BT-HNTR Datasheet," https://www.supermicro.com/en/Aplus/system/2U/2124/AS-2124BT-HNTR.cfm, 2020, [retrieved: 04, 2022].

[41] Intel Corporation, "Intel P4510 Datasheet," https:

//ark.intel.com/content/www/us/en/ark/products/122579/intel-ssd-dc-p4510-series-4-0tb-2-5in-pcie-3-1-x4-3d2-tlc.html, 2018, [retrieved: 04, 2022].

[42] GIGA-BYTE Technology Co., "Gigabyte R282-Z94 Datasheet," https://www.gigabyte.com/Enterprise/Rack-Server/R282-Z94-rev-100#Specifications, 2021, [retrieved: 04, 2022].

[43] Samsung, "Samsung MZQL23T8HCJS-00A07 Datasheet," https://semiconductor.samsung.com/ssd/datacenter-ssd/pm9a3/mzql23t8hcjs-00a07/, 2021, [retrieved: 04, 2022].

[44] HARDKERNEL CO., LTD., "Odroid HC4 Datasheet," https://wiki.odroid.com/odroid-hc4/hardware/hardware, 2021, [retrieved: 04, 2022].

[45] Raspberry Pi Ltd., "Raspberry Pi 4 Model B specifications," https://www.raspberrypi.com/products/raspberry-pi-4-model-b/, [retrieved: 10, 2022].

[46] Western Digital Corporation, "WD Black WD10SPSX Datasheet," https://documents.westerndigital.com/content/dam/doc-library/en_us/assets/public/western-digital/product/internal-drives/wd-black-hdd/product-brief-western-digital-wd-black-mobile-hdd.pdf, 2020, [retrieved: 04, 2022].

[47] Samsung, "Samsung V-NAND SSD 860 PRO Datasheet," https://www.samsung.com/semiconductor/global.semi.static/Samsung_SSD_860_PRO_Data_Sheet_Rev1_1.pdf, 2018, [retrieved: 04, 2022].

[48] NETGEAR, Inc., "Netgear GS110EMX Datasheet," https://www.netgear.com/images/datasheet/switches/webmanagedswitches/GS110EMX_GS110MX.pdf, 2021, [retrieved: 04, 2022].

[49] Armbian, "Armbian Odroid HC4," https://www.armbian.com/odroid-hc4/, 2022, [retrieved: 04, 2022].

[50] MEAN WELL, "MW HRP 450-15 Datasheet," https://www.meanwell.com/webapp/product/search.aspx?prod=HRP-450, 2021, [retrieved: 04, 2022].

[51] ZES ZIMMER Electronic Systems GmbH, "ZES Zimmer LMG 450 Brochure," https://www.zes.com/en/content/download/286/2473/file/lmg450_prospekt_1002_e.pdf, 2010, [retrieved: 04, 2022].

[52] Dell Inc., "Dell Precision 3650 Tower Hardware Specification," https://www.delltechnologies.com/asset/en-us/products/workstations/technical-support/precision-3650-spec-sheet.pdf, 2021, [retrieved: 04, 2022].

[53] S. Just, "Crimson: A new ceph OSD for the age of persistent memory and fast NVMe storage," Presentation at Linux Storage and Filesystems Conference (Vault '20), Santa Clara, CA, Feb. 2020, https://www.usenix.org/conference/vault20/presentation/just [retrieved: 04, 2022].

[54] T. Ludwig, S. Krempel, J. Kunkel, F. Panse, and D. Withanage, "Tracing the MPI-IO Calls' Disk Accesses," in *Recent Advances in Parallel Virtual Machine and Message Passing Interface*, B. Mohr, J. L. Träff, J. Worringen, and J. Dongarra, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 322–330, DOI: 10.1007/11846802_45.