Assessing Data Deduplication Trade-offs from an Energy and Performance Perspective

Lauro Beltrão Costa*, Samer Al-Kiswany*, Raquel Vigolvino Lopes‡ and Matei Ripeanu*
* *Electrical and Computer Engineering Department*
The University of British Columbia
Vancouver, BC, Canada
{*lauroc,samera,matei*}*@ece.ubc.ca*
‡ *Departamento de Sistemas e Computação*
Universidade Federal de Campina Grande
Campina Grande, PB, Brazil
raquel@dsc.ufcg.edu.br

***Abstract*—The energy costs of running computer systems are a growing concern: for large data centers, recent estimates put these costs higher than the cost of hardware itself. As a consequence, energy efficiency has become a pervasive theme for designing, deploying, and operating computer systems. This paper evaluates the energy trade-offs brought by data deduplication in distributed storage systems. Depending on the workload, deduplication can enable a lower storage footprint, reduce the I/O pressure on the storage system, and reduce network traffic, at the cost of increased computational overhead. From an energy perspective, data deduplication enables a trade-off between the energy consumed for additional computation and the energy saved by lower storage and network load. The main point our experiments and model bring home is the following: while for non energy-proportional machines performance- and energy-centric optimizations have break-even points that are relatively close, for the newer generation of energy proportional machines the break-even points are significantly different. An important consequence of this difference is that, with newer systems, there are higher energy inefficiencies when the system is optimized for performance.**

I. Introduction

Storage systems have evolved to employ techniques that enable trade-offs over performance metrics such as throughput, reliability, and generated I/O overhead. While the trade-off space over these traditional performance metrics has been extensively studied over the past decades, performance with regard to energy efficiency attracted less attention. As a result, determining the balance between system performance and its energy bill is a complex and unexplored task.

Data deduplication [1]–[8] is one storage system optimization that provides a trade-off between compute and I/O overheads. It consumes additional CPU cycles to detect data similarity and, in return, reduces the storage footprint, pressure on the storage system, and network traffic.

While the impact of data deduplication on traditional metrics (e.g., data throughput, storage footprint) is well understood [1]–[3], previous studies leave an important gap: energy consumption analysis. They overlook two important issues. First, while data deduplication increase the CPU load, it may reduce the network and storage devices' load. As a result, it is unclear under what scenarios it will lead to energy savings, if any. Second, the performance impact of energy-centric tuning of the storage system is unexplored.

To gain experience with the methodological and practical difficulties of such analysis, this work targets fixed-block data deduplication as a first case-study. We explore deduplication for its popularity – a wide class of data-intensive systems employ deduplication from backup systems, virtual machine image repositories, to checkpointing optimized storage systems; and for its simplicity to design experiments that explore the characteristics of various workloads.

The contribution of this work is threefold. First, an empirical evaluation of energy consumption for data deduplication (§III). This evaluation quantifies the energy consumption on two hardware platforms with different characteristics in terms of power proportionality and identifies deduplication's break-even point - threshold that separates the cases when it is worth or not deduplicating data from an energy or performance perspective (§IV). Second, it shows that break-even points for performance and for energy efficiency are different (§IV). Third, a simple energy consumption model that makes it possible to reason about the benefits of deduplication and offers an approximation for the energy break-even point (§V).

This work is related to a rapidly growing body of works on deploying energy efficient systems. It joins others in which the focus is to understand the energy consumption of compression techniques in different scenarios [9], [10]. To the best of our knowledge, this work is the first to study the impact of data deduplication on storage system energy consumption. Besides, it also considers different generations of machines to demonstrate the impact of the new power proportional hardware (i.e., hardware whose power consumption is proportional to the utilization level)

978-1-4577-1221-0/11/$26.00

[11] on energy consumption.

The empirical evaluation and energy model suggest that, as storage systems and their components become increasingly energy proportional, the energy and throughput break-even points will shift farther apart. This trend has an important consequence: optimizations for energy efficiency and performance will likely conflict. As a result, storage system designers and deployers will have to make conscious and informed decisions about the metric to optimize for.

II. Background and Related Work

This work directly relates to efforts in design/evaluation of data deduplication solutions and energy efficient systems.

A. Data deduplication

Data deduplication is a method to detect and eliminate similarities in the data. Briefly, deduplication works as follows: when a new file is stored, the storage system divides the file into blocks, computes identifiers for each block based on its content (e.g., by hashing the data), compares the identifiers obtained with the identifiers of the blocks already stored, and persistently stores only the new blocks (i.e., those with different identifiers). Similar blocks are not stored, saving storage space, reducing the I/O load, and also reducing network load. Experiences show that space savings can be as high as 60% for a generic archival workload [1], 85% for application checkpointing [3], or 95% for a virtual machine image repository [7].

A number of research and commercial systems employ various forms of deduplication targeting two main goals: (i) *reducing the storage footprint*, such as Venti [1] and Foundation [2] optimized for archival, Mirage [5] optimized for storing virtual machine images, and DEBAR [6] optimized for enterprise-scale backup services; and (ii) *improving performance* by reducing the pressure on the persistent storage or the volume of data transferred over the network, including low bandwidth file system [8], web acceleration [12] content-based caching [13], and high performance storage system StoreGPU [4].

While previous work focuses on the storage footprint and run time related benefits in systems of different scales, the impact of deduplication on energy consumption is not clear. On the one side, detecting similarity introduces computational overhead to compute the hashes of data blocks. On the other side, if there is detectable similarity in the workload, the computational overhead can be offseted by less storage or network effort. This work explores this trade-off from an energy standpoint.

B. Energy Optimized Systems

With non power proportional hardware (i.e., hardware that draws the same power regardless of the utilization level), energy efficiency [11] is tightly coupled with high resource utilization. Consequently, to increase energy efficiency, previous work recommends increasing system utilization through mechanisms such as application consolidation [14], or through runtime optimizations to reduce per-task energy consumption (e.g., [15]).

Hardware has become increasingly power proportional. This trend opens new opportunities for energy efficiency for the software stack: it enables shifting work from the most energy efficient component (most power proportional) of a computer system to the less energy efficient component (e.g., from disk to CPU) to reduce the total volume of energy consumed for a specific task [14].

Recently, Chen *et al.* [9] and Kothiyal *et al.* [10] have investigated the trade-off of compressing or not data in the context of MapReduce applications and data centers, respectively. Similar to this work, they concluded that compression is not always the best choice in terms of energy consumption, as it depends on the workload. Ma *et al.* [16] investigate deduplication performance using a commodity low power coprocessor for hash calculations. Unlike this work, they do not explore the trade-off of exchanging I/O operations by extra CPU load, and do not quantify deduplication energy savings.To the best of our knowledge, this work is the first to study the energy impact of deduplication and to put in perspective the impact of new generations of computing systems that have different energy proportionality profiles.

III. Methodology

To start investigating the impact of deduplication on energy consumption we chose an empirical approach: monitoring a distributed storage system (MosaStore [17]) that adopts deduplication and subjecting it to a checkpointing-like workload. The main advantage of this methodology is that it anchors the investigation with data obtained from real systems subjected to a realistic workload. The main drawback is limiting the exploration space: only deduplication using fixed-size blocks and only a distributed storage scenario (as these are core assumptions made by MosaStore).

The rest of this section presents the key aspects of the storage system used, the generated workload, the deployment platform, and our solution to estimate energy consumption.

A. Deduplication in MosaStore

MosaStore [17] is an object-based distributed storage system that can be configured to enable workload-specific optimizations. More relevant to this study, MosaStore can be configured to employ deduplication to eliminate similarities between blocks of consecutive versions of the same file. MosaStore has three main components: a metadata manager, storage nodes, and the client's system access interface (SAI) that provides a POSIX API.

Each file is divided into fixed-size blocks stored on the storage nodes. For each file, the metadata manager maintains a block map which contains block-level information

including the identifier (hash value) for each block. The SAI implements the client-side deduplication mechanism. To write to a file, the SAI first retrieves the file's previous-version block map from the manager, divides the new version of the file into blocks, computes the hash value for each block, and searches the file's previous-version block map for the same hash values. The SAI sends to the storage nodes only the blocks not found in the previous-version block map, and reuses the blocks already stored. Once the write operation completes, the SAI commits the new file's block-map to the metadata manager. Other deduplication systems [1], [2] use similar techniques.

B. Checkpointing: An Application Use Case

Checkpointing is representative for workloads that can benefit from deduplication as there can be significant similarity between successive checkpoint images. We have collected and analyzed [3] checkpoint images produced using VM-supported checkpointing (using Xen), process-level checkpointing (using the BLCR checkpointing library [18]), and application-based checkpointing. Depending on the checkpointing technique, the time interval between checkpoints, and the deduplication technique used, the detected similarity between consecutive files varied between no similarity to 82% similarity (for BLAST bioinformatics application checkpointed using BLCR at 5min intervals).

This study uses synthetic workloads that mimic checkpointing workloads: The workload generators produces files at regular time intervals and controls the similarity ratio between consecutive file versions (from 0 to 100% similarity).

C. Evaluation Testbed

We evaluate performance and energy consumption on two classes of machines which we label *'new'* and *'old'*:

- *'new'* machines (Dell PowerEdge 610) are equipped with Intel Xeon E5540 (Nehalem) @ 2.53GHz CPU (launched Q1'09, max TDP 80W), 48GB RAM, 1Gbps NIC, and two 500GB 7200 rpm SATA disks. Nehalem is a new Intels architecture that exhibits major improvements in power efficiency. Indeed, a machine consumes 86W in idle mode and 290W at peak utilization;
- *'old'* machines (Dell PowerEdge 1950) are equipped with Intel Xeon E5395 (Clovertown) @ 2.66GHz CPU (launched Q4'06, max TDP 120W), 8GB RAM, 1Gbps NIC, and two 300GB 7200 rpm SATA disks. A machine consumes 188W in idle mode and 252W at peak.

All machines run Fedora 14 Linux OS (kernel 2.6.33.6). MosaStore uses the same configuration in all experiments. To simplify power measurements, we use only two machines: the storage node on one machine; and the manager, the SAI, and the workload generator on a second machine. The machines are connected by a Dell PowerConnect 6248 10Gbps switch.

D. Evaluating Performance and Energy Consumption

We use two WattsUP Pro [19] power meters to measure the energy consumption for each of the two machines. They measure energy at the wall power socket, capturing the energy consumption for the entire system. A third machine collects the measurements from the meters via a USB interface. The meters provide a 1W power resolution, 1Hz sampling rate, and $\pm 1.5\%$ accuracy.

For energy, since the meters give only a 1Hz maximum sampling rate, we collect the measurements every second during an experimental batch (from the beginning of the first write until the completion of the last write). The power is given in watts (joules/s) for the last measurement interval, giving the energy consumed during the last measurement interval. For each machine, we sum energy consumption estimates to obtain the total amount of energy consumed during the experiment batch. We report the average energy consumed per write by dividing the total energy consumption by the number of checkpoint image writes.

The evaluation considers the energy consumed by all storage system components: manager, storage node and client SAI. On the client node, however, the workload generator runs together with the storage system component. Since it is not possible to isolate the consumption only for the storage system path, the evaluation conservatively reports the energy consumed for the whole system, including the workload generator [2].

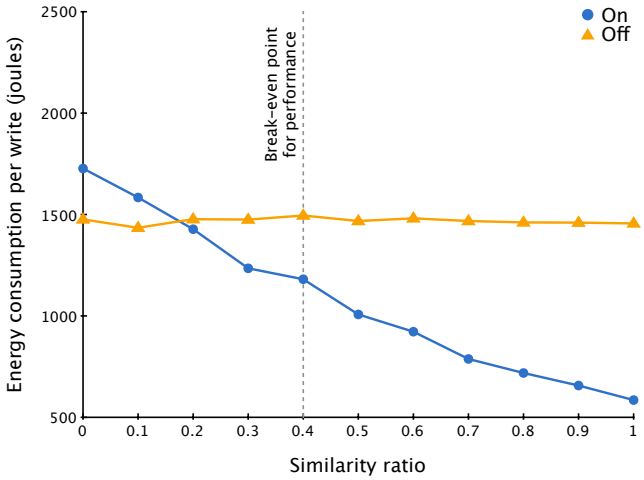
In the plots, each point presents the average value for a file write operation (for energy or time) calculated over a batch of 50 writes at a fixed similarity level. The experiments consider different data sizes (32, 64, 128, 256 and 512MB) while varying the similarity level (0% - 100%, with increments of 10%). Although the time and energy required for each operation varies, the overall relation of energy consumption, time, and similarity is the same regardless of the data sizes. Thus, the plots show results just for 256MB.

IV. Evaluation

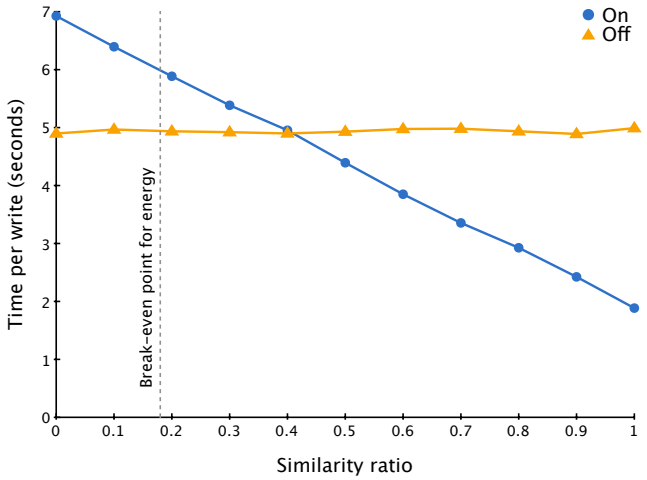
We aim to estimate energy consumption, checkpoint image write performance, and the performance and energy break-even points for platforms with different energy proportionality properties. To this end we execute the same synthetic application while varying the level of data similarity (this is equivalent to varying the frequency of the checkpointing operation).

Figures 1(a) and 1(b) present the energy consumption and, respectively, the average write time per checkpoint on the *'new'* testbed. The main point to note is that the break-even points for energy and performance are different: for similarity lower than 18%, hashing overheads are not compensated by the energy savings in I/O operations,

[2]Indeed, we have noted that the workload generator is lightweight compared to the client SAI.

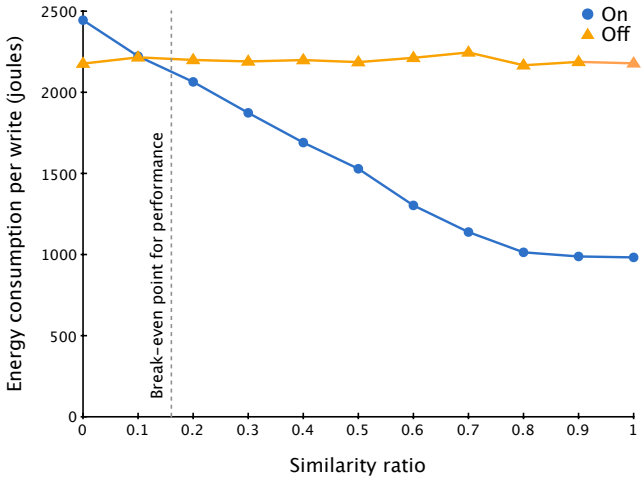


(a) Average energy consumption

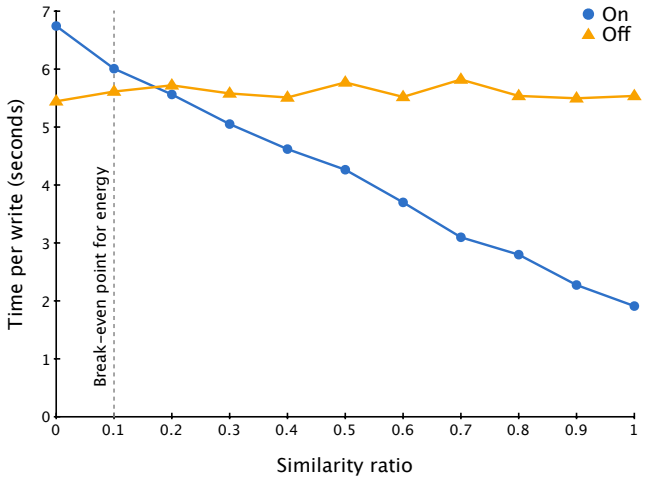


(b) Average time to write

Figure 1. Average energy consumed and time to write a 256MB file for different similarity levels in the *'new'* testbed. Note: Y axes do not start at 0.



(a) Average energy consumption



(b) Average time to write

Figure 2. Average energy consumed and time to write a 256MB file for different similarity levels in the *'old'* testbed. Note: Y axes do not start at 0

thus enabling deduplication brings benefits only when the workload has a similarity rates higher than 18%. From a performance perspective, enabling deduplication makes sense only for higher similarity levels (higher than 40%).

A second point to note is that the relative gains enabled by deduplication differ for energy and time. For energy, the highest consumption level (at 0% similarity) is 2.9x larger than the lowest (at 100% similarity). For the time to write a checkpoint, this ratio is 3.6x. Although deduplication enables energy savings starting at lower similarity than it enables them for time, at 95% of similarity deduplication saves almost the same rate of energy and time (around 50%).

Figures 2(a) and 2(b) present the energy consumption and, respectively, the average time per checkpoint write for the *'old'* testbed. Compared to the *'new'* testbed, the break-even point for energy (at 10% similarity) is closer to the one for performance (at 16% similarity). For this testbed there are similar differences in the relative gains that deduplication enables.

One important factor to note is that, although the two testbeds have almost the same performance profile (as evidenced by the checkpoint write performance), they have different energy profiles. The energy consumption per write with deduplication turned off is about 45-50% higher in the *'old'* testbed (even though the writing time is only 10% higher). With deduplication turned on and a high similarity rate, the differences are even more striking: about 2x higher energy consumption in the *'old'* testbed for about

the same write time (Figure 1(a)). The reason is that the newer generation machines (with Nehalem CPUs) are more power proportional and save energy by matching the level of resources enabled (e.g., switching on/off cores) to the offered load.

The main point the experiments bring home is the following: while for non energy-proportional machines performance- and energy-centric optimizations have break-even points that are relatively close, for the newer generation of energy proportional machines the break-even points are significantly different. An important consequence of this difference is that, with newer systems, there are higher energy inefficiencies when the system is optimized for performance. The experiments presented above quantify these inefficiencies: on the *'old'* testbed, optimizing for performance leads to an up to 5% energy inefficiency (in the 10-16% similarity rate interval). For the *'new'* testbed, optimizing for performance leads to an up to 20% energy inefficiency (in the 18-40% similarity rate interval).

V. Modelling Data Deduplication Trade-offs

"All models are wrong; but some are useful." – G.E.P. Box (1976)

Section IV shows that deduplication can bring energy and/or performance savings if enough similarity exists in the workload. It also shows that the break-even points for energy and performance are different and depend on the characteristics of the deployment environment. In these conditions, system administrators need a tool to support their configuration decisions related to deduplication.

This section proposes a simple model to guide the storage system configuration. The model can be used in two ways. First, to identify whether deduplication will lead to energy savings, for a given level of similarity. Second, to estimate the energy impact of upgrading the system: for example, by adding SSDs (to increase energy efficiency and performance) or by adding energy-efficient accelerators (e.g., GPUs) to support deduplication [4].

Two guidelines direct the design of the model. First, the model should be simple to use and do not require extensive machine benchmarking or the use of power meters. Ideally, it should be seeded with power information available on technical data-sheets of various components. Second, the model should be simple and intuitive, even at the cost of lower accuracy, since such models have higher chances to be adopted and used to guide decisions in complex settings.

Let us define the following variables:

- P_I, P_C, P_{IO} are the power consumed by the machine in idle state, peak CPU load, and peak I/O (disk and network) load, respectively.
- E_H is the *extra* energy consumed by the machine to compute the hash value for one data block. It is roughly approximated by $E_H = (P_C - P_I) \times T_H$, where T_H is the time for hashing a single block.
- E_{IO} is the energy consumed for the transfer of one block on the storage path - including all system calls, sending the block from the client machine, and receiving it at the storage node and storing it on disk. $E_{IO} = 2 \times (P_{IO} - P_I) \times T_{IO}$. The factor 2 appears since there are one client and one storage node involved in the storage path, where T_{IO} is the time for sending and storing a block.
- S is the similarity ratio of the data;
- B is the total number of blocks to be written.

For each write operation, the extra energy needed to compute the hash values is $E_H \times B$. The energy saved by reducing the stress on the storage path is $E_{IO} \times B \times S$. Every time the energy savings are higher than the additional energy spent to compute hash values ($E_{IO} \times B \times S > E_H \times B$ *if* $S > \frac{E_H}{E_{IO}} = \frac{(P_C - P_I) \times T_H}{2 \times (P_{IO} - P_I)} \times T_{IO}$), then it is worth turning deduplication on. Note that this choice is independent of the data volume B.

The administrator can easily use this formula to guide her deduplication related decisions. The parameters needed can be easily benchmarked (T_H and T_{IO}), or can be provided by system assemblers in technical sheets (P_I, P_C, and P_{IO}), and estimated or extracted from the workload history (S).

The above modeling exercise highlights our main theme: for past, non-energy proportional systems optimizing for energy and for performance are similar. Thus, the decision to optimize for energy depends only on the relative runtime to hash or store a data block and the similarity level present in the workload. Power proportionality brings new factors into this equation: the relative position of power consumed when idle, under maximum I/O load and under maximum compute load. Once a system is power proportional (that is, if P_I is significantly lower than P_C and/or P_{IO}) and it draws different power levels at peak CPU vs. at peak I/O load ($P_C \neq P_{IO}$) a richer trade-off space emerges.

To evaluate the accuracy of the simple model, we compare its prediction of the energy break-even point with that of an oracle. In this case, the oracle is the measurements of the two testbeds from §IV. To benchmark the testbed and estimate P_I, P_C, and P_{IO} we ran an idle workload as well as a CPU, and disk and network intensive workload and measured the consumed power for each workload separately.

Benchmarking the testbed and plugging its characteristics into the model indicates that the energy break-even points are at 21.4% similarity for the *'new'*testbed and at 18.1% similarity for the *'old'* testbed. This is close to the oracle: the actual measurements indicate 18% and, respectively, 10% for the energy break-even points. Despite the model simplicity, it estimates the break-even point with relatively good accuracy. It only fails to predict when the similarity ratio is between 18-21.4% in the new testbed or between 10-18.1% in the old. When the similarity is in these ranges deduplication

configured using the model consumes less than 10% extra energy compared to the optimal configuration (Figures 1(a) and 2(a)).

Note that designing an accurate fine-granularity model for the storage system energy consumption is a complex task: the main reason is that it is hard to decouple the energy consumption of different components in the system, and to decouple the energy consumed by the application running in the system.

VI. Conclusion

While for non-energy proportional computer systems, energy- and performance-centric optimizations do not conflict, the recent trend towards increasingly energy-proportional systems opens new trade-offs that make the design space significantly more complex as optimizations for these two criteria start to diverge.

To better understand this issue, this work focuses on data deduplication. We evaluate the energy consumption and the performance of data deduplication in storage systems on two different generations of machines. This evaluation supports and quantifies the above intuition: the more power proportional a system, the higher the opportunities to trade among different resources and the larger the gap between performance- and energy-centric optimizations.

We also propose an energy consumption model that highlights the same issues and, in spite of its simplicity, can be used to reason about the energy and performance break-even points when configuring a storage system.

Acknowledgements

This research was supported in part by the DOE ASCR X-Stack program under grant DE-SC0005115 and the Institute for Computing, Information and Cognitive Systems (ICICS) at UBC. We want to thank Emalayan Vairavanathan for his great effort in the implementation of MosaStore.

References

[1] S. Quinlan and S. Dorward, "Venti: A new approach to archival data storage," in *USENIX Conf. on File and Storage Tech. (FAST '02)*.

[2] S. Rhea, R. Cox, and A. Pesterev, "Fast, inexpensive content-addressed storage in foundation," in *USENIX 2008 Annual Technical Conf.*, 2008.

[3] S. Al-Kiswany, M. Ripeanu, S. S. Vazhkudai, and A. Gharaibeh, "stdchk: A checkpoint storage system for desktop grid computing," in *Int. Conf. on Distributed Computing Systems*, ser. ICDCS, 2008.

[4] A. Gharaibeh, S. Al-Kiswany, S. Gopalakrishnan, and M. Ripeanu, "A gpu accelerated storage system," in *ACM Int. Symposium on High Performance Distributed Computing*, ser. HPDC, 2010.

[5] D. Reimer, A. Thomas, G. Ammons, T. Mummert, B. Alpern, and V. Bala, "Opening black boxes: using semantic information to combat virtual machine image sprawl," in *ACM Int. Conf. on Virtual execution environments*, ser. VEE, 2008.

[6] T. Yang, H. Jiang, D. Feng, Z. Niu, K. Zhou, and Y. Wan, "Debar: A scalable high-performance deduplication storage system for backup and archiving," in *IEEE Int. Symposium on Parallel Distributed Processing (IPDPS)*, 2010.

[7] A. Liguori and E. V. Hensbergen, in *Workshop on I/O Virtualization*, M. Ben-Yehuda, A. L. Cox, and S. Rixner, Eds.

[8] A. Muthitacharoen, B. Chen, and D. Mazières, "A low-bandwidth network file system," in *Symp. on Oper. Sys. Princ.(SOSP'01)*.

[9] Y. Chen, A. Ganapathi, and R. H. Katz, "To compress or not to compress - compute vs. IO tradeoffs for mapreduce energy efficiency," in *Proc. of the first ACM SIGCOMM workshop on Green networking*, ser. Green Networking '10, 2010, pp. 23–28.

[10] R. Kothiyal, V. Tarasov, P. Sehgal, and E. Zadok, "Energy and performance evaluation of lossless file data compression on server systems," in *International Systems and Storage Conference*.

[11] L. A. Barroso and U. Hölzle, "The Case for Energy-Proportional Computing," *IEEE Computer*, vol. 40, no. 12, pp. 33–37, 2007.

[12] S. Ihm, K. Park, and V. S. Pai, "Wide-area network acceleration for the developing world," in *USENIX annual technical Conf.(USENIXATC'10)*.

[13] R. Koller and R. Rangaswami, "I/O Deduplication: Utilizing Content Similarity to Improve I/O Performance," in *USENIX Conf. on File and Storage Tech. (FAST '10)*, 2010.

[14] S. Harizopoulos, M. A. Shah, J. Meza, and P. Ranganathan, "Energy efficiency: The new holy grail of data management systems research," in *CIDR*, 2009.

[15] P. Sehgal, V. Tarasov, and E. Zadok, "Evaluating performance and energy in file system server workloads," in *USENIX Conf. on File and Storage Tech. (FAST '10)*.

[16] L. Ma, C. Zhen, B. Zhao, J. Ma, G. Wang, and X. Liu, "Towards fast de-duplication using low energy coprocessor," in *Int. Conf. on Networking, Architecture, and Storage*.

[17] S. Al-Kiswany, A. Gharaibeh, and M. Ripeanu, "The case for a versatile storage system," *SIGOPS Oper. Syst. Rev.*, vol. 44, pp. 10–14, March 2010.

[18] P. H. Hargrove and J. C. Duell, "Berkeley lab checkpoint/restart (blcr) for linux clusters," *Journal of Physics: Conf. Series*, vol. 46, no. 1, p. 494.

[19] "Watts up? pro product details," https://www.wattsupmeters.com/.