On the Importance of Evaluating Storage Systems' $Costs

Zhichao Li, Amanpreet Mukker, and Erez Zadok — Stony Brook University

Abstract

Modern storage systems are becoming more complex, combining different storage technologies with different behaviors. Performance alone is not enough to characterize storage systems: energy efficiency, durability, and more are becoming equally important. We posit that one must evaluate storage systems from a monetary cost perspective as well as performance. We believe that cost should consider the workloads used over the storage systems' expected lifetime. We designed and developed a versatile hybrid storage system under Linux that combines HDD and SSD. The SSD can be used as cache or as primary storage for hot data. Our system includes tunable parameters to enable trading off performance, energy use, and durability. We built a cost model and evaluated our system under a variety of workloads and parameters, to illustrate the importance of cost evaluations of storage systems.

1 Introduction

Storage systems are getting more complex with solid-state technologies rapidly taking hold, shingled devices available, and hybrids thereof being proposed and commercialized [15]. As the amount of digital data grows rapidly, virtualization and cloud technologies highlight the need to consolidate storage and save on the longer-term data storage costs. Complex workloads play a key role in how storage systems behave. The interplay of hardware, software, and workloads has a significant impact on throughput, energy consumption [8], and device durability. We propose to evaluate modern storage systems from a monetary cost perspective that includes many dimensions including performance [3]. We assume that server-class storage systems should be utilized at high yields, due to consolidation and virtualization. We propose that monetary costs should be evaluated over the expected lifetime of the storage system, typically years, and consider device wear-out and replacement [13].

Several studies integrate SSDs into storage systems, and some consider the original purchase cost or short-term energy use, but neglect to consider the long term impact on device wear-out [4–6, 9, 12, 14]. Some simulated the results instead of conducting empirical studies [6, 12]. Few explore the pros and cons of tiering vs. caching approaches to hybrid storage systems [1, 5].

To facilitate this study, we developed a device-mapper target for the Linux kernel that combines HDD and SSD. Our target can use the SSD as either (1) a cache with asynchronous write-back of dirty data to the HDD, or (2) a primary store for hot data. Our target include versatile policies for management of hot/cold data between the SSD and HDD. We parameterized many aspects of our system and added counters and instrumentation to measure its behavior under various configurations. We conducted extensive experiments using many workloads and configurations—including single-drives and hybrids. We present a subset of these results here.

Next, we built a cost model that also includes lifetime cost of ownership: energy and power costs, replacement cost, and more. We populated the model with realistic figures from industry and our own empirical experiments. We observed that for some workloads, an SSD-only solution incurs the highest overall costs in the short term but much lower costs in the long run. We also observed that for some workloads, using the SSD as a cache had lower costs than when the SSD was used as primary hot-data storage; but other workloads completely reversed this trend. That is why we believe that future storage systems must be evaluated across dimensions of lifetime cost, performance, as well as workloads.

2 Cost Model

A cost metric is important to justify storage systems' expenditures [4, 10]. Generally, monetary costs include upfront purchase and the Total Cost of Ownership (TCO) [3]. We use a time factor to estimate longer-term costs. We summarize our model below.

$$1 \leq i \leq n \, (n: the\, number\, of\, devices) \quad (1)$$

$$1 \leq \alpha \, (time\, factor,\, default = 1) \quad (2)$$

$$Cost = Purchase + TCO \quad (3)$$

$$Purchase = \sum_{i=1}^{n} C_{dev_i} \quad (4)$$

$$C_{dev_i} = Normalized\, Price_{dev_i} \times Capacity_{dev_i} \quad (5)$$

$$TCO = \alpha \times (C_{energy} + C_{power} + C_{endu}) + C_{ser} \quad (6)$$

$$C_{energy} = Lookup_{LIPA}(Amount_{energy}) \quad (7)$$

$$C_{power} = Lookup_{LIPA}(Amount_{power}) \quad (8)$$

$$C_{endu} = \sum_{i=1}^{n} C_{endu_i} \quad (9)$$

$$C_{endu_i} = C_{dev_i} \times \frac{dev_i\, wearout}{Limit_i} \quad (10)$$

$$dev_i\, wearout = \begin{cases} writes \text{ if } dev_i = SSD \\ \#startstop \text{ if } dev_i = HDD \end{cases} \quad (11)$$

$$Limit_i = \begin{cases} Limit_{writes} \text{ if } dev_i = SSD \\ Limit_{cycles} \text{ if } dev_i = HDD \end{cases} \quad (12)$$

$$C_{ser} = fixed\, estimation \quad (13)$$

Equation 1 names a variable i for each device. Equation 2 specifies α as the time factor to project future cost estimates (i.e., run the same amount of workload multiple times). Equation 3 shows that the total cost ($Cost$) depends on the upfront purchase cost ($Purchase$) and the TCO. Equations 4 and 5 show that the upfront purchase cost depends on a normalized price of each device ($Normalized\,Price_{dev_i}$) and the capacity of each device ($Capacity_{dev_i}$). Note that the normalized price of each device can change over time. In our paper we present results based on prices the the Intel SSD and Seagate HDD we purchased in 2012.

Prices	$<=$ **7KW**		$<=$ **145KW**		$>$ **145KW**	
	Egy	**Pow**	**Egy**	**Pow**	**Egy**	**Pow**
offpeak	0.0863	0	0.0191	0	0.0218	0
peak	0.1052	0	0.0340	48.78	0.0446	28.76
intermediate	0.0863	0	0.0317	5.94	0.0356	8.13

Table 1: LIPA energy and power prices for commercial use as of May 2013, based on per KWh and per KW. "Egy" is Energy; "Pow" is Power.

Equation 6 shows that the TCO depends on the energy cost (C_{energy}), the power cost (C_{power}), the endurance cost (C_{endu}), and the service cost (C_{ser}). We also use α as the time factor to predict future costs associated with the energy, power, and endurance (or replacement) in the longer run (i.e., assuming we run the same workload multiple times). Equations 7 and 8 show that we can get the energy and power cost by looking up the price table ($Lookup_{LIPA}$) provided by the local electricity authority (Long Island Power Authority), as shown in Table 1. We assume that: (1) the energy we collected is distributed by $3/8$, $1/4$, and $3/8$ in accordance with off-peak, peak, and intermediate; (2) the power we collected in off-peak, peak, and intermediate is the average power. The energy and power measurement is based on the whole system. We used a simplified method to estimate the energy and power cost. Equation 9 shows that we can get the total endurance cost by summarizing each device's endurance cost (C_{endu_i}). Equation 10 shows that we can get each device's endurance cost by multiplying the wear out degree ($\frac{dev_i\,wearout}{Limit_i}$) of each device type by the device's cost (C_{dev_i}). Note that the wear-out degree and the endurance limit of each device may be different.

Equations 11 and 12 show that the Flash-based SSD endurance depends more on the writes ($writes$). Note that reads also affect SSD's endurance: we convert the effect of reads to writes based on a parameterized ratio (e.g., writes caused by reads is calculated as $reads/10$). We also show that for HDD, the number of start-stop cycles ($\#startstop$) is a major factor. Other factors include vibration, sector errors, and more [11]. We use the number of HDD start-stop cycles for simplicity. Based on manufacturers' specifications, our SSD can sustain a total of 36.5TB writes, and our HDD can handle at most 300,000 spin up/down cycles. Equation 13 shows that we use fixed estimation as the service cost ($C_{service}$) for the hardware setup. Service costs may include manpower and air-conditioning costs.

3 Systems

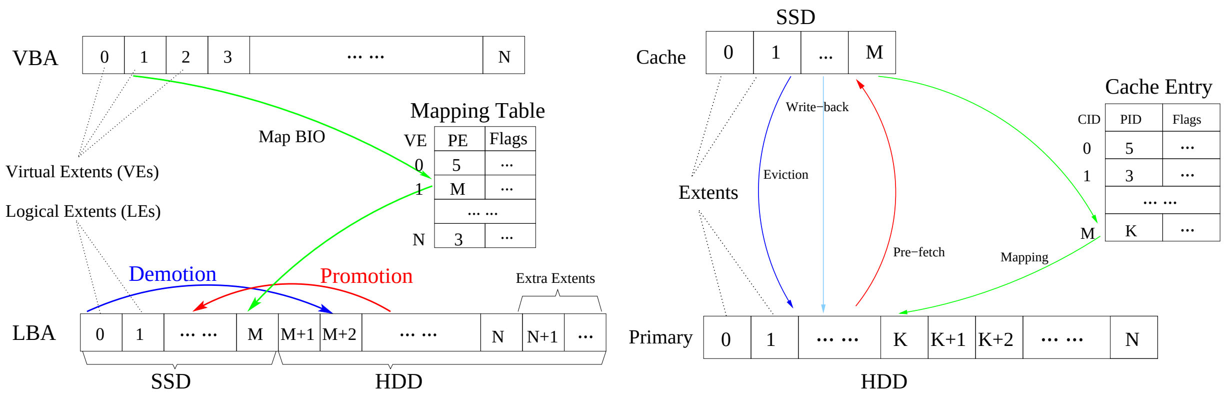
We implemented both tiering and caching hybrid systems in the Linux Device Mapper framework. We wrote around 4,000 LoC of kernel code in twelve months. Both systems are scalable: they can be easily configured to use multiple drives with minor code change. However, to better analyze the behavior of our system, we used a two-drive setup in this paper: one SSD and one HDD. We present the data management of the two system in Figure 1. The two systems are fairly similar in terms of design and implementation: frequently accessed data goes to the faster device and less frequently accessed data goes to slower device. The two systems are versatile to enable adaptation to different workloads. We support several configurable system parameters: (1) Extent Size (ES); (2) Promotion/Pre-fetching Threshold (PT)—access counts before being promoted/fetched; and (3) Maximum Concurrent Migration Limit (MCML). We summarize the key differences between the two systems below.

Capacity. In the caching system, since the SSD is not counted toward the total capacity, the HDD capacity needs to be expanded to yield the same amount of total capacity as the tiering system has. When the SSD capacity is not largely different from the total capacity, a tiering system can have better purchase cost per GB than the caching system does.

Management Unit. The caching system uses a cache entry table and the tiering system uses a mapping table. The cache entry table maintains mapping information only from the cache device to the lower-level device, and contains not only the four fields in the mapping table of the tiering system (i.e., extent ID, state, usage counter, and time-stamp of the latest access), but also a dirty flag to indicate whether a cached extent is updated or not.

Data Movement. The two systems use the same method to move data around. We name the hot data moving process *promotion* and *pre-fetch* in the tiering system and caching system, respectively. We name the cold data moving process *demotion* and *eviction*, respectively. The caching system does not need to reserve extra extents in the HDD for eviction to succeed, as it is guaranteed to map an extent from the SSD to the HDD.

Read/Write Policy. In a tiering system, since the SSD is used as primary storage, reads and writes access the data from the current location either on the SSD or HDD according to the mapping table. Cold data migrates to the HDD and hot data eventually migrates to the SSD using



(a) **Tiering System**

(b) **Caching System**

*Figure 1: **Data Management in Two Modes of Our System**.*

Workload	**Drive Size**	**Reads**		**Writes**	
		Total	**Avg Sz**	**Total**	**Avg Sz**
Web-search	32GB	1,055,236	16KB	212	8KB
FIU online	8GB	655,526	8KB	4,211,786	4KB

*Table 2: **Trace Workloads Summary***

kernel threads. In a caching system, reads and writes access data from the SSD if the data is still there, else from the HDD. If it is an SSD write hit, the system stores information of the pending write-back I/O in a queue, and an asynchronous write-back kernel thread wakes up to flush dirty writes from the SSD to the HDD. I/O access can be slow during write-back activity.

4 Evaluation

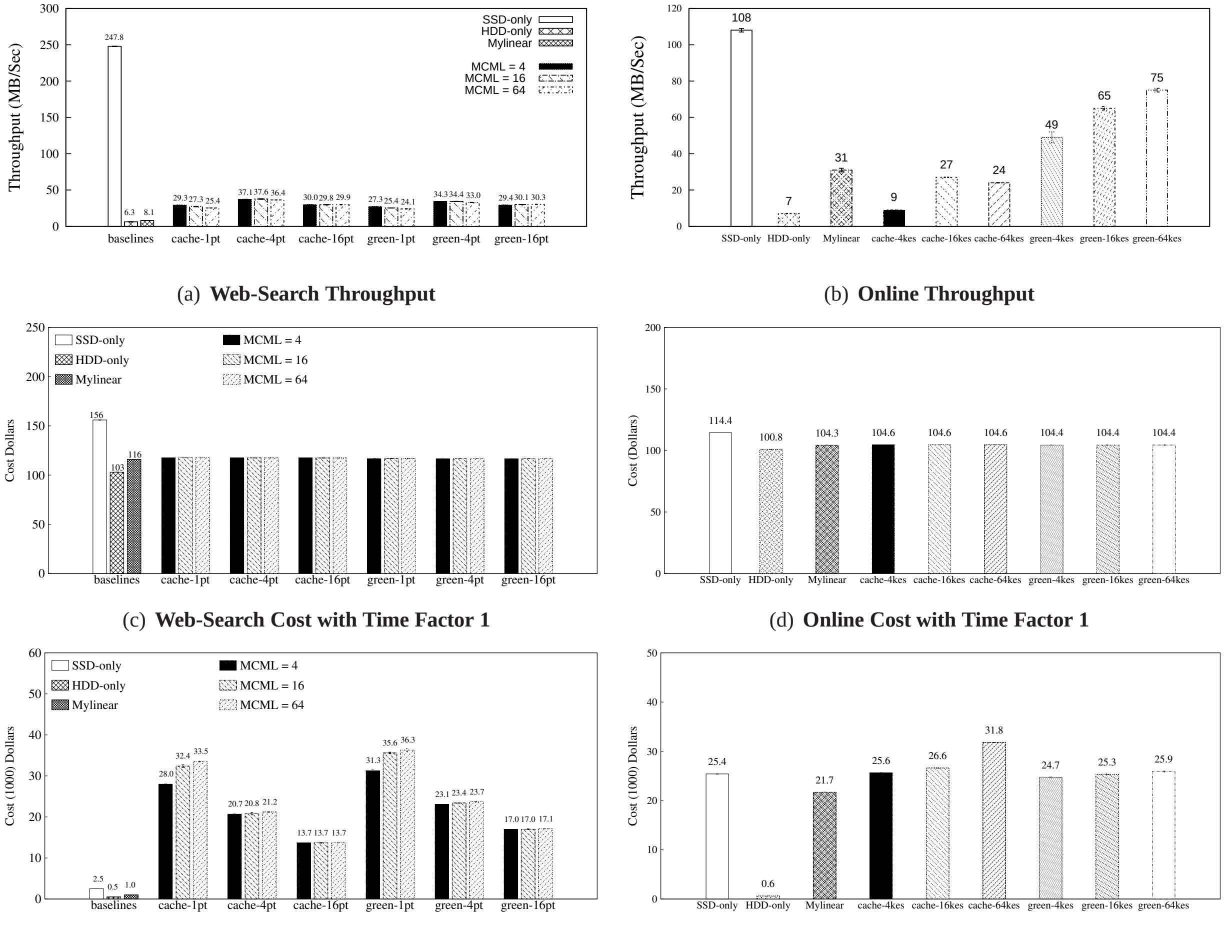
Experimental setup. We experimented on two identical Lenovo®ThinkCenter computers. Each has 4GB RAM and one Intel®Core-2$^{\text{TM}}$Quad 2.66GHz CPU. For storage, we used parts of an Intel SSDSA2CW300G3 300GB SSD and Seagate ST32000641AS 2TB HDD. A Linux 3.5.0 kernel ran on a separate SATA drive. We connected each computer to a WattsUP Pro ES in-line power meter to measure energy and power use.

Benchmarks. We evaluated with two trace workloads: Web-search trace from the UMass Trace Repository and the FIU's online trace. Trace details are shown in Table 2. We set up 32GB and 8GB storage capacities for the Web-search and online traces, respectively. The "green" is our tiering hybrid drive and the "cache" is our caching hybrid drive. "Mylinear" is another tiering hybrid drive based on the Linux DM "linear" target that linearly maps from the virtual block address to the logical block address without any additional data management. For the two tiering hybrids, we chose $1/4$ as an example ratio for the SSD over total capacity. To show comparable results, we used the same SSD and total capacities for the caching system. We ran all tests three times. We computed the standard deviations and presented as error bars in figures.

Results. We show the results in Figure 2. We also have results for Filebench's file-server workload but omit them for brevity because we observed similar trends. For Web-search, the caching system achieves slightly higher throughput (4–9%) than the tiering system does when the Pre-fetching Threshold (PT) is 4 and 16. It achieves similar throughput as the tiering system does when PT is 64 (Figure 2(a)). The SSD hit ratio ranges from 81–98% for the caching system, and ranges from 85–98% for the tiering system. Both the SSD hit ratio and the data movement affect the throughput for hybrids. Mylinear achieves an SSD hit ratio of only 8%. This workload has many more reads than writes (see Table 2). Thus the overhead of the write-back is not significant as there are only a few writes. Moreover, as the SSD in the tiering system contains either cold or hot data beforehand, it can add some overhead to the overall throughput. However, the cache device in the caching system only contains hot data. It suggests that overall throughput of the caching system could be higher than the tiering system if the tiering primary storage (SSD) initially contains cold data.

The caching system has lower total cost (8–20%) in the long run than the tiering system does (Figure 2(e)). For Web-search, when the time factor is 100,000, it translates to an average of 2.1 years (ranging from 0.2–7.7 years) for all types of benchmarks, a reasonable time-frame for the expected lifetime of storage systems. The reasons are: (1) there are no additional primary I/Os to the SSD in the caching system, but the tiering system does since its SSD is used as primary storage; and (2) the SSD endurance reduction counts more toward the total cost of ownership in the long run. When the time factor is 1 (Figure 2(c)), the caching system incurs little additional dollar cost compared to the tiering system because the caching system only has to pay for the expanded HDD capacity.

For online, the caching system achieves lower throughput (58–82%) than the tiering system as the ES varies (Figure 2(b)). The SSD hit ratio ranges from 92–99% for the caching system, and from 98–99% for the tiering system. Both the SSD hit ratio and data movement affect system throughput. Mylinear achieves an SSD hit ratio of 84%. When the ES is 4K, the throughput of the



(a) **Web-Search Throughput**

(b) **Online Throughput**

(c) **Web-Search Cost with Time Factor 1**

(d) **Online Cost with Time Factor 1**

(e) **Web-Search Cost with Time Factor 100,000**

(f) **Online Cost with Time Factor 100,000**

*Figure 2: **Two Traces Replay Results**. For Web-Search, we set ES to 1MB. For Online, we set MCML to 16 and PT to 1.*

caching system is 82% lower because it has even more write-back I/Os. This workload has lots of writes (Table 2). It suggests that the overhead of the write-back can be a throughput bottleneck.

The caching system has a higher total cost (5–23%) in the long run than the tiering system (Figures 2(f)). For FIU online, when the time factor is 100,000, it translates to an average of 3.3 years (ranging from 0.7–9.8 years) for all types of benchmarks. There are no additional primary I/Os to the SSD for the caching system, but the caching system has many more write-back I/Os. Therefore, the caching system reduces the SSD endurance faster, and incurs more long-term cost than the tiering system. When the time factor is 1 (Figure 2(d)), the caching system incurs just a little bit more cost than the tiering system due to the same reason.

Overall, we observed six trends. (1) For read-intensive workloads, a larger PT value reduces long-term costs; for write-intensive workloads, a smaller ES value reduces long-term costs. (2) The HDD-only system has the least initial capital investment and lowest long-term dollar cost, but it has the lowest performance. (3) The SSD-only system has the highest initial capital investment, and can incur low and high long-term costs for read-intensive and write-intensive workloads, respectively; but it has the highest performance. (4) Tiering and caching systems have the benefits of incurring only medium initial capital investments, and can incur some long-term costs to a different degree depending on the workloads; both systems have medium performance. (5) The tiering and caching systems incur more long-term cost than Mylinear does due to data movement; but both systems achieve better performance than Mylinear does. (6) Different tiering and caching system configurations lead to variations in cost, which increases as the time factor increases.

5 Related Work

Few have investigated the long-term costs of storage systems with SSDs. Some use simulation [6, 12], instead of empirical experiments. Some do not consider the SSD replacement cost in their total cost calculation [4, 9, 10]. Industry also discusses this, but detailed cost models that include TCOs are not publicly available [2, 14].

Several have compared caching and tiering systems. MAID [1] briefly discusses the pros and cons of caching and migration based policies for massive storage systems. With the advent of Phase Change Memories (PCMs), Kim et al. [5] evaluate PCMs for enterprise storage systems using case studies of caching and tiering approaches. However, there is no direct comparison study performed for the caching and tiering approaches from the perspective of total cost of ownership.

Our work is different in five aspects. (1) We collect real energy and power numbers from experiments. (2) We consider the SSD's endurance cost. (3) We scale the experiments to observe long-term effects. (4) We developed and discussed a cost model containing the total cost of ownership. (5) We built two realistic systems (i.e., tiering and caching) with similar strategies and environment to evaluate fairly the pros and cons of the caching and tiering based hybrid storage systems.

6 Limitations and Future Work

Modeling storage systems' monetary costs is challenging. Our model has several limitations. We do not fully consider the following three aspects yet: computer hardware cost, air-conditioning cost, and labor cost. We also do not yet consider equipment financing cost with different interest rates. We simplify several conditions to facilitate easier understanding: (1) the hardware setup in a real data center may be more complex than ours; (2) the service cost may vary accordingly; and (3) the workloads in a real data center may be more complex than ours. It is our hope that this work helps others build more elaborate cost models in the future.

Caching and tiering systems share several design traits. Our caching system is fairly similar to the tiering one. Although both systems estimate the endurance metric by counting SSD reads and writes and the HDD start-stop cycles, the endurance metric can be improved. Detailed access to the SSD internals (e.g., erasure cycle counts, FTL behavior) could improve the SSD's endurance model. The size ratios of the SSD vs. HDD in both systems affects throughput, energy and power, device endurance, and dollar cost. We are currently investigating that [7], especially where in large scale storage servers, a cache is much smaller than total capacity.

7 Conclusion

We developed a device-mapper target for the Linux kernel that combines HDD and SSD together. Our system can use the SSD as either a cache or a primary storage for hot data. We built a cost model that also considers the lifetime cost of ownership: energy and power costs, replacement cost, and more. Our extensive evaluation results show that for some workloads, an SSD-only solution incurs the highest overall costs in the short term but much lower costs in the long run. We also observed that for some workloads, using the SSD as a cache had lower costs than when the SSD was used as primary hot-data storage; but other workloads completely reversed this trend. It is therefore important that future storage systems be evaluated across dimensions of lifetime cost, performance, as well as workloads. It is our hope that this work would encourage new research into more realistic long term cost models of storage systems.

References

[1] D. Colarelli and D. Grunwald. Massive Arrays of Idle Disks for Storage Archives. In *Proceedings of the 2002 ACM/IEEE conference on Supercomputing*, pages 1–11, 2002.

[2] David Floyer. Enterprise Flash Drive Cost and Technology Projections, 2009. *http://wikibon.org/wiki/v/Enterprise_Flash_Drive_Cost_and_Technology_Projections*.

[3] Gartner, Inc. Server Storage and RAID Worldwide. Technical report, Gartner Group/Dataquest, 1999. *www.gartner.com*.

[4] J. Guerra, H. Pucha, J. Glider, W. Belluomini, and R. Rangaswami. Cost Effective Storage Using Extent Based Dynamic Tiering. In *USENIX FAST*, 2011.

[5] H. Kim and S. Seshadri and C. L. Dickey and L. Chiu. Evaluating Phase Change Memory for Enterprise Storage Systems: A Study of Caching and Tiering Approaches. In *Proceedings of the 12th USENIX Conference on File and Storage Technologies*, pages 33–45, Berkeley, CA, 2014. USENIX.

[6] Y. Kim, A. Gupta, B. Urgaonkar, P. Berman, and A. Sivasubramaniam. HybridStore: A Cost-Efficient, High-Performance Storage System Combining SSDs and HDDs. In *IEEE MASCOTS*, 2011.

[7] Z. Li. *GreenDM: A Versatile Tiering Hybrid Drive for the Trade-Off Evaluation of Performance, Energy, and Endurance*. PhD thesis, Computer Science Department, Stony Brook University, May 2014.

[8] Z. Li, K. M. Greenan, A. W. Leung, and E. Zadok. Power Consumption in Enterprise-Scale Backup Storage Systems. In *Proceedings of the Tenth USENIX Conference on File and Storage Technologies (FAST '12)*, San Jose, CA, February 2012. USENIX Association.

[9] D. Narayanan, E. Thereska, A. Donnelly, S. Elnikety, and A. Rowstron. Migrating server storage to ssds: analysis of tradeoffs. In *EuroSys '09: Proceedings of the 4th ACM European conference on Computer systems*, pages 145–158, New York, NY, USA, 2009. ACM.

[10] P. O'Neil, E. Cheng, D. Gawlick, and E. O'Neil. The log-structured merge-tree (LSM-tree). *Acta Inf.*, 33(4):351–385, 1996.

[11] E. Pinheiro, W. Weber, and L. A. Barroso. Failure trends in a large disk drive population. In *Proceedings of the Fifth USENIX Conference on File and Storage Technologies (FAST '07)*, pages 17–28, San Jose, CA, February 2007. USENIX Association.

[12] T. Pritchett and M. Thottethodi. SieveStore: A Highly-Selective, Ensemble-level Disk Cache for Cost-Performance. In *Proceedings of the 37th Annual International Symposium on Computer Architecture*, ISCA '10, 2010.

[13] David S. H. Rosenthal, Daniel C. Rosenthal, Ethan L. Miller, Ian F. Adams, Mark W. Storer, and Erez Zadok. The economics of long-term digital storage. In *The Memory of the World in the Digital age: Digitization and Preservation*. United Nations Educational, Scientific and Cultural Organization (UNESCO), September 2012.

[14] J. D. Strunk. Hybrid Aggregates: Combining SSDs and HDDs in a Single Storage Pool. *SIGOPS Oper. Syst. Rev.*, pages 50–56, 2012.

[15] Tintri VMStore. *www.tintri.com/resources/videos/introduction-to-tintri/*.