J Supercomput (2014) 67:277–303
DOI 10.1007/s11227-013-0999-3

HybridPlan: a capacity planning technique for projecting storage requirements in hybrid storage systems

Youngjae Kim · Aayush Gupta · Bhuvan Urgaonkar · Piotr Berman · Anand Sivasubramaniam

Published online: 9 August 2013

Abstract Economic forces, driven by the desire to introduce flash into the high-end storage market without changing existing software-base, have resulted in the emergence of solid-state drives (SSDs), flash packaged in HDD form factors and capable of working with device drivers and I/O buses designed for HDDs. Unlike the use of DRAM for caching or buffering, however, certain idiosyncrasies of NAND Flash-based solid-state drives (SSDs) make their integration into hard disk drive (HDD)-based storage systems nontrivial. Flash memory suffers from limits on its reliability, is an order of magnitude more expensive than the magnetic hard disk drives (HDDs), and can sometimes be as slow as the HDD (due to excessive garbage collection (GC) induced by high intensity of random writes). Given the complementary properties of HDDs and SSDs in terms of cost, performance, and lifetime, the current consensus among several storage experts is to view SSDs not as a replacement for HDD, but rather as a complementary device within the high-performance storage hierarchy.

Y. Kim (✉)
National Center for Computational Sciences, Oak Ridge National Laboratory, Oak Ridge, TN, USA
e-mail: kimy1@ornl.gov

A. Gupta
IBM Almaden Research, San Jose, CA, USA
e-mail: guptaaa@us.ibm.com

B. Urgaonkar · P. Berman · A. Sivasubramaniam
Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA

B. Urgaonkar
e-mail: bhuvan@cse.psu.edu

P. Berman
e-mail: berman@cse.psu.edu

A. Sivasubramaniam
e-mail: anand@cse.psu.edu

Thus, we design and evaluate such a hybrid storage system with *HybridPlan* that is an improved capacity planning technique to administrators with the overall goal of operating within *cost-budgets*. HybridPlan is able to find the most cost-effective hybrid storage configuration with different types of SSDs and HDDs

Keywords Storage systems · Solid-state drives · Resource provisioning · Mathematical optimization and modeling

1 Introduction

Hard disk drives (HDDs) have been the preferred media for data storage in high-performance and enterprise-scale storage systems for several decades. For example, the HPC storage cluster at the Oak Ridge Leadership Computing Facility (OLCF) provides an aggregate bandwidth of over 240 GB/s with over 10 petabytes of RAID 6 formatted capacity using 13,400 HDDs [21]. The disk storage market totals approximately $34 billion annually and is continually on the rise [18]. Manufacturers of HDDs have been successful in ensuring sustained performance improvements while substantially bringing down the price-per-byte. During the past decade, the maximum internal data rate (IDR) for hard disks has witnessed a 20-fold increase resulting from improvements in rotational speeds (RPM) and storage densities; seek times have improved by a factor of 4 over the same period [13, 22].

However, there are several shortcomings inherent to HDDs that are becoming harder to overcome as we move into faster and denser design regimes. First, designers of HDDs are finding it increasingly difficult to further improve the RPM (and hence the IDR) due to problems of dealing with the resulting increase in power consumption and temperature [4, 12, 20]. Second, any further improvement in storage density—another way to increase the IDR—is increasingly harder to achieve and requires significant technological breakthroughs such as perpendicular recording [34]. Third, and perhaps most serious, despite a variety of techniques employing caching, prefetching, scheduling, write-buffering, and those based on improving parallelism via replication (e.g., RAID), the mechanical movement involved in the operation of HDDs can severely limit the performance that hard disk based systems are able to offer to workloads with significant randomness and/or lack of locality. Specific to our interest in this paper, in a large-scale shared storage system, *consolidation* can result in the multiplexing of unrelated workloads imparting/exaggerating the randomness. Furthermore, such consolidated workloads are likely to exhibit degraded temporal and (more seriously for HDD-based systems) spatial locality, thereby potentially adversely affecting performance [10].

Alongside improvements in HDD technology, significant advances have also been made in various forms of solid-state memory such as NAND flash [32], magnetic RAM (MRAM) [41], STT-RAM [39], phase-change memory (PCM) [16], and Ferroelectric RAM (FRAM) [38]. Solid-state memory offers several advantages over hard disks: lower access latencies for random requests, lower power consumption, lack of noise, and higher robustness to vibrations and temperature. In particular, recent improvements in the design and performance of NAND flash memory (simply

Table 1 Performance, lifetime, cost comparison among different storage media [24]

Media	Access time (μs)	Lifetime	Cost ($/GB)
DRAM	0.9	N/A	125
SSD	<45 (read) , <200 (write)	10 K–1 M erase cycles	25
HDD	<5500	MTTF = 1.2 Mhr	3

flash henceforth) have resulted in its becoming popular in many embedded and consumer devices. Small form-factor HDDs have already been replaced by flash in some consumer devices like music players, PDAs, digital cameras, etc. Flash has, however, only seen limited success in the enterprise-scale storage market [24]. Although (i) the aforementioned advances in flash technology and (ii) its dropping cost-per-byte [6] had led several storage experts to predict the inevitable demise of HDDs [3], flash has so far not been able to make inroads into the enterprise-scale storage market to the extent expected [24].

Table 1 presents a comparison of the performance, lifetime, and cost of representative HDDs, SSDs, and DRAM. There are several important implications of how these properties compare with each other. Flash technology possesses a number of idiosyncrasies that have hindered the SSD from replacing HDD in the general enterprise market. First, it is evident that there exists a huge gap between the Cost/GB of HDDs and SSDs.[1] Second, unlike HDD, SSDs possess an asymmetry between the speeds at which reads and writes may be performed. As a result, the throughput a SSD offers for a write-dominant workload is lower than for a read-dominant workload. Third, flash technology restricts the locations on which writes may be performed—a flash location must be *erased* before it can be written—leading to the need for a garbage collector (GC) for/within an SSD. Certain workload characteristics (in particular, the presence of randomness) increase the fragmentation of data stored in flash memory, i.e., logically consecutive sectors become spread over physically non-consecutive blocks on flash. This exacerbates GC overheads, thereby significantly slowing down the SSD [23]. Furthermore, this slowdown is non-trivial to anticipate. A given set of random writes may themselves experience good throughput, but increase fragmentation, thereby degrading the performance of requests (read or write) arriving much later in future. Finally, to further complicate matters, unlike HDDs, SSDs have a lifetime that is limited by the number of erases performed. Therefore, excessive writing to flash, while potentially useful for the overall performance of a flash-based storage system, limits its lifetime.

From the above description, it should be clear that SSDs are fairly complex devices [1]. Their peculiar properties related to cost, performance, and lifetime make it difficult for a storage system designer to neatly fit them between HDD and DRAM. As has been observed in other recent research, under certain workload conditions,

[1] A similar gap exists between SSD and DRAM. Furthermore, this rules out major changes in the role played by DRAM in future systems that employ SSDs. DRAM will continue to retain both of its important roles related to caching and buffering. Therefore, we will not compare these two devices in the rest of this paper.

an SSD can perform worse than the HDD [23] and in certain SSDs, read throughput can be slower than write throughput for small random workload patterns [5, 30]. Similarly, the SSD's lifetime limit (which is ultimately related, in a complex way, to the intensity of write operations), calls for careful design to gainfully utilize them in conjunction with HDDs in the enterprise. The degrading lifetime with increased write-intensity may result in premature replacement of these devices, adding to deployment, procurement, and administrative costs. Note that we have picked a lifetime of 5 years for a HDD just for illustrative purposes [35]. An excellent study of the useful lifetimes of disks based on data from real enterprise-scale systems appears in a paper by Schroeder and Gibson [35]. Finally, the low throughput offered by SSDs to random write-dominated workloads, which are frequently encountered in enterprise-scale systems [23], necessitates intelligent partitioning of data in such hybrid environments while ensuring that the management costs do not overwhelm the performance improvements.

On this paper, we make the following specific contributions:

- We propose a hybrid storage system containing HDDs and SSDs, called *HybridStore* that exploits the complementary properties of these two media to provide improved performance and service differentiation under a certain cost budget.
- The main component of the HybridStore is a *capacity planner* (*HybridPlan* henceforth) that makes long-term resource provisioning decisions for the expected workload. It is designed to optimize the cost of equipment that needs to be procured to meet desired performance and lifetime needs for the workload.
- We develop models that HybridPlan employs to find the most economical storage configuration given devices and workloads using Mixed Integer Linear Programming (ILP). We expect HybridPlan to provide "rules-of-thumb" to administrators of hybrid storage systems when making provisioning decisions.

As an illustrative result, HybridPlan is able to identify close to minimum SSDs of HybridStore by planning a well-provisioned system needed to meet a specified performance goal for realistic enterprise-scale workloads—MSR Cambridge and Microsoft Exchange Server Traces.

Road-map The rest of this paper is organized as follows. In Sect. 2, we present related works. In Sect. 3, we present the motivation for HybridStore, and provide a bird's eye-view of the overall HybridStore architecture. In Sect. 4, we describe the capacity planner (HybridPlan) in detail. Then we extensively evaluate HybridPlan in Sect. 5. Finally, we present concluding remarks in Sect. 6.

2 Related work

A lot of research has been conducted to improve performance of HDDs using nonvolatile memory. eNVy [43] uses nonvolatile memory for data storage wherein battery-backed SRAM is used to reduce the write overhead. HeRMES [27] uses magnetic RAM to reduce the overhead of frequently and randomly accessing meta-data. MEMS [42] has also been exploited to improve disk performance. Finally, storage

architecture in which flash memory is used as a conventional disk cache has already been explored in [26] Our work goes beyond merely using flash as a cache/write-buffer—rather than treating flash as a *subordinate to the disk*, HybridStore views these as *complementary* storage media.

Samsung and Microsoft [31] have developed and deployed hybrid hard disks for laptops (where NAND flash is located at an upper level in the storage hierarchy as compared to hard disk). Booting time and resuming process from the disk have been improved by overlapping the time for spinning up disk drive with the booting process from flash memory. Bisson et al. [2] have explored the use of a flash-based NVRAM as a write buffer to reduce write latency of hard disks for desktop environments. They employ I/O redirection to reduce seeking overhead from disk by directing requests likely to incur long seeks to the on-disk NVRAM. We view the MixDyn component of our system as conceptually close to Bisson et al.'s work and would be interested in comparing MixDyn with their I/O redirection technique in the future. This work is the closest to the HybridDyn component in [22]. However, their model fails to effectively capture the intricacies of flash, and thus is susceptible to poor performance induced by fragmentation caused by random writes. Our approach to modeling hybrid system, considers the performance variation of flash devices along with varying workload characteristics. A key difference is that our flash model additionally captures the fragmentation within flash (caused by random writes) and incorporates it into its redirection decision-making.

There have been several research efforts to integrate SSDs in HDDs storage systems. In a recent work from Microsoft Research, Narayanan et al. [29] examined the role of SSDs in enterprise storage systems using a number of real data center traces available to them. Their work explores the cost-benefit trade-offs of various SSD and HDD configurations flash and disk capacities/configurations for these real traces. However, there are several key differences between our contributions. First, our work, in particular HybridPlan, is much more general and can be used to target any type of devices including STT-RAM and PCM. In this work, we focus only on flash since it is the only mature technology with concrete and meaningful numbers for cost and performance. Second, we have developed a data classification strategy which can be used to decide partitioning of workloads among the chosen devices. Third, while they admit that flash wear-out needs to be considered while using it as a write buffer, they do not explore any specific ways of doing this. Closest to our work is a recent paper by Guerra et al. [9] and we consider it highly complementary with similar results and insights. There are differences in our performance modeling approaches. Additionally, we consider lifetime constraints and include power costs in our formulation.

3 HybridStore: hybrid storage systems combining SSDs and HDDs

3.1 Motivation for HybridStore

From the above description in Sect. 1, it should be clear that SSDs are fairly complex devices. Their peculiar properties related to cost, performance, and lifetime make it difficult for a storage system designer to neatly fit them between HDD and DRAM.

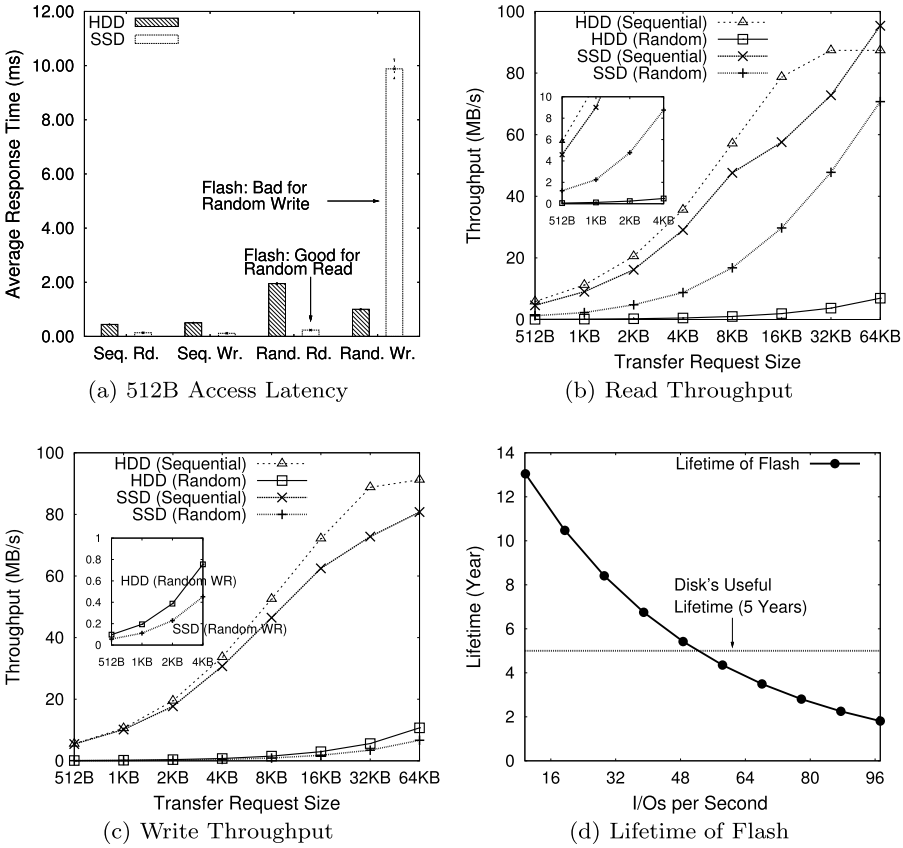


Fig. 1 A comparison of the performance and lifetime characteristics of representative SSD and HDD. Although MTTFs for HDDs tend to be of the order of several decades, recent analysis has established that other factors (such as replacement with next, faster generation) implies a much shorter actual lifetime and hence we assume a nominal lifetime of 5 years in the enterprise. Note that Seq., Rand., Wr., and Rd., respectively, denote Sequential, Random, Write, and Read. I/O request size in (**d**) is a page size (2 KB). *Each bar in* (**a**) *is shown with 99 % confidence interval*

To illustrate the complexity of the relationship between HDD and SSD, we benchmark a 32 GB 2.5 in SLC based Super Talent FSD32GB25M SSD and a 150 GB 3.5 in 10 K RPM Western Digital Raptor X HDD for their performance. Figure 1(a) compares flash and hard drives for 512 Bytes access latency. We ran IO meter which sends raw I/O requests to both devices, SSD and HDD in order to remove caching effect on host system and measure device performance. We first considered ramp-up time for device warm-up. The same experimental setup applied to Fig. 1(c), (d). Figure 1(b) considers 100 GB flash with garbage collection and wear-leveling. We used our HybridStore simulator to calculate the lifetime [22].

As has been observed in other recent research, under certain workload conditions, an SSD can perform worse than the HDD [23] and in certain SSDs, read throughput can be slower than write throughput for small random workload patterns [5, 30]. A look at Figs. 1(a)–(c) provides an illustration of such behavior and calls for care-

ful design to gainfully utilize them in conjunction with HDDs in the enterprise. The degrading lifetime with increased write-intensity, as shown in Fig. 1(d), may result in premature replacement of these devices, adding to deployment, procurement, and administrative costs. Note that we have picked a lifetime of 5 years for a HDD just for illustrative purposes [35]. An excellent study of the useful lifetimes of disks based on data from real enterprise-scale systems appears in a paper by Schroeder and Gibson [35]. Finally, the low throughput offered by SSDs to random write-dominated workloads (Fig. 1(c)), which are frequently encountered in enterprise-scale systems [23], necessitates intelligent partitioning of data in such hybrid environments while ensuring that the management costs do not overwhelm the performance improvements.

As has been shown in Fig. 1, SSDs are fairly complex devices due to their particular properties related to performance, cost, and lifetime. Therefore, storage administrators need to make careful decisions on making their storage systems using both SSDs and HDDs. Otherwise, they would not experience the best benefits offered by both devices.

3.2 HybridPlan: a long-term resource planner

HybridStore is a hybrid storage system combining SSDs and HDDs and exploits the complementary properties of these media to provide improved performance while meeting lifetime and budget (defined as installation and recurring costs) requirements. HybridPlan is a long-term resource provisioner. We envision HybridPlan to be a tool that would enable storage administrators to provision both kinds of devices in cost-effective ways. The decision-making of HybridPlan would occur at coarse time-scales (months to years) corresponding to when procurement and deployment decisions are made. HybridPlan employs a ILP solver engine based on mathematical formulations to make its provisioning decisions. HybridPlan is intended to cost-effectively provision devices to allow HybridStore to (i) adhere to the performance needs of hosted workloads and (ii) meet useful lifetime requirements specified by the administrator, under these workload assumptions.

4 HybridPlan

4.1 Problem formulation

Given the large price gap between flash-based SSDs and HDDs, it is essential to determine appropriate capacities of these devices for the workload the system expects to support. We define this process as *capacity planning*. Our goal is to determine the right number of devices which need to be deployed in a heterogeneous storage environment. as shown in Fig. 2. In this section, we provide a general form of comprehensive methodology using Linear Programming (LP), a well-known technique for optimization problems.

We formulate our capacity planning problem as a means of minimizing the cost of acquiring/installing HybridStore while meeting the workload-specified performance

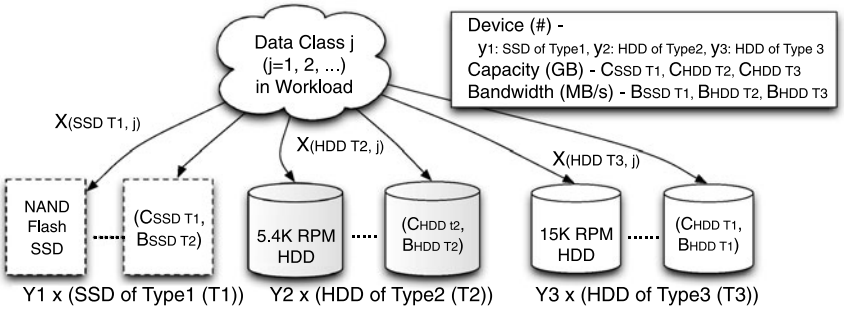


Fig. 2 Example: Storage system model for HybridStore. i, j, k, l: i, j, k, lth data classes, $X_{SSD\ T1,i}$: data class i on $y_1 \times$ SSDs of type1, $X_{SSD\ T2,j}$: data class j on $y_2 \times$ SSDs of type2, $X_{HDD\ T3,k}$: data class k on $y_3 \times$ HDDs of type3, $X_{HDD\ T4,l}$: data class l on $y_4 \times$ HDDs of type4

($Perf_{Budget}$) and useful lifetime budget ($Life_{Budget}$). Our model does not consider other hardware costs such as network switch. $Cost_{Installation}$ indicates the installing cost of devices. Apart from these, costs associated with power consumption, thermal consumption (cooling), other maintenance and management activity form the recurring costs denoted by $Cost_{Recurring}$. However, information in the academic domain about the management/maintenance costs of these devices (HDDs and SSDs) is still sparse and inconclusive. Furthermore, management costs vary with legal contracts and are highly subjective. Hence, we only consider electricity cost of operation due to power consumption as recurring cost in this study. In sum, the total HybridStore cost is the sum of these individual costs ($Cost_{HybridStore} = Cost_{Installation} + Cost_{Recurring}$). Our optimization problems are summarized as follows:

$$\text{Minimize} \quad Cost_{HybridStore}$$

$$\text{Subject to} \quad \begin{cases} Perf_{Hybridstore} \geq Perf_{Budget} \\ Life_{Hybridstore} \geq Life_{Budget} \end{cases}$$

$$\text{where } Cost_{HybridStore} = Cost_{Installation} + Cost_{Recurring}$$

Figure 2 shows an example of a storage system model employing different device types. For the purpose of our study, we try and minimize the deployment and operation cost (in terms of $) subject to a combination of both performance and redeployment constraints (due to lifetime of flash memory). We use IOPS as a metric of HybridStore's performance and term this metric as the system's *Performance Budget*. In addition, we need to consider lifetime issues in the flash because the blocks in SSDs become unreliable beyond 10 K–1 M erase cycles [7]. This poses a significant challenge for a system administrator whose objective is to keep system redeployment frequency and costs under control. We capture these objectives in terms of a *Lifetime Budget* (years) for the system, which is the time between successive capacity planning decisions and equipment procurement/installation.

In order to build cost-effective storage system called HybridStore, we need a framework (we call it HybridPlan henceforth) to satisfy the given workload requirements using known device characteristics. In next section, we describe the data

classification technique for partitioning workloads and then develop a comprehensive methodology for determining the appropriate number of devices using Mixed ILP.

4.2 Data classification

We can extract workload requirements (space or bandwidth requirement) by analyzing their IO traces collected for fairly long time.

In this section, we describe the data classification methodology used to partition a workload into smaller subsets. A workload can be characterized on the basis of certain features such as total size, read/write ratio, request arrival rate, etc. Furthermore, each workload is a collection of subworkloads, which exhibit similar features. Each of these subworkloads are called as *classes*. Classes help in determining commonality within workload streams and are essential for accurately mapping workloads to devices for an effective capacity planning framework.

The entire logical address space of the workload is divided into fixed-size chunks, then mapped to different classes. These fixed-size chunks are called as *records*. We use 1 MB as record size roughly corresponds to the granularity of data prefetching done by HDDs/SSDs. As described above, we need to find the attributes for describing workloads. We capture temporal locality in workloads using the total number of accesses to the records. We use average read volume to describe the read/write ratio of each record. Similarly, we use the median of request sizes to ascribe the request size to each record. This parameter captures the spatial locality in workloads. The reason for using median instead of average request size is because our experimental evaluation showed that median proved to be a better metric as it negated the impact of outliers (very small or very big requests) and helped in distributing records across classes appropriately. Lastly, we use the total number of IOs in a record in the workload as a measure of the number of IO arrivals to the record over the entire life of the workload. Note that there may be other attributes, which can be used for data classification. However, our empirical analysis binds these parameters to be effective in partitioning workloads across classes.

Now that we have defined the parameters for characterizing workloads, we develop a mechanism to segregate records across classes. Figure 3 describes the mechanism with hierarchical data classification on data records. Temporal locality of a class is defined using hot/cold regions. The records which are accessed at least once in the workload are considered *hot* whereas records, which are never accessed are treated as *cold* records. Classes are further divided based on request sizes. Records with request size less than 16 KB are part of highly-random request classes whereas records with request sizes greater than 64 KB are part of highly-sequential data classes. We also have intermediate data classes depending on whether request sizes are greater than 32 KB (partially sequential) or not (partially random). We use the lower(25th), middle(50th) and upper(75th) quartiles of the overall distribution of total IOs across the records to further segregate these records. The readers should note that all the data points for creating classes as described above are based on empirical evaluation as well as qualitative intuition. In this study, we have considered 33 data classes. The number of data classes can be further optimized using merging and reduction techniques and is part of our future work.

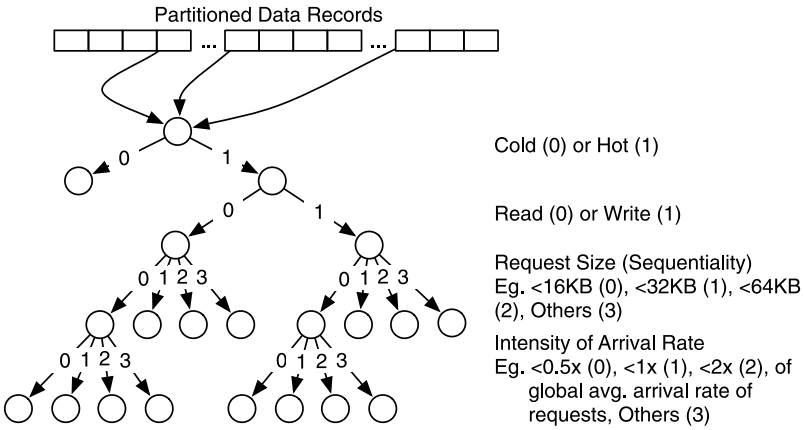


Fig. 3 Hierarchical data classification. The data points for creating classes are based on empirical evaluation. The details about how we select the values for data points, can be found in text

The device characteristics can be obtained not only from their data sheets but also from performance tests.

4.3 Optimization solver

We formulate our provisioning problem as a Mixed Integer Program. We describe a tool which finds the most cost-effective storage configuration using available devices for the provisioned workloads by reducing our optimization problem to a Mixed ILP problem. Table 2 shows declaration of each variable for problem formulation of HybridPlan.

As described earlier, we consider installation cost and electricity cost for the total cost of the storage systems. Given the properties of I different types of devices, the overall installation cost of storage systems is highly dependent on the numbers of each device type $i \in I$, and its individual device cost is $D\$_i$:

$$Cost_{Installation} = \sum_{i=1}^{I} D\$_i \times y_i$$

Given the electricity cost per time ($= K_\$$) and the power consumption of device type i ($= P(i)$), the energy consumption of overall storage system ($= E$) over time followed by the overall electricity cost of operation by the energy consumption can be calculated as

$$E_{Operation} = \sum_{i=1}^{I} y_i \times \int_t P_i \, dt$$

$$Cost_{Recurring} = K_\$ \times E_{Operation}$$

Table 2 Declaration of variables

Variable	Description
General variable	
$K_\$$	Electricity cost ($/KWH)
T	Total trace time
$LIFE$	Lifetime: the threshold (in years) for which provisioning is being done
Device	
i ($i = 1, 2, 3, \ldots, I$)	Device type
C_i	Capacity of device of type i
U_i	Utilization of device of type i
RB_i	Read bandwidth of device of type i
WB_i	Write bandwidth of device of type i
IT_i	Initiation time of device of type i i.e the time for initiating each IO (1/IOPS)
P_i	Average Power consumption of device of type i
L_i	Lifetime of device of type i
$D\$_i$	Cost of device of type i
$E\text{-}UNIT_i$	Block size of a device of type i (only for SSDs)
$W\text{-}UNIT_i$	Size of each write on a device of type i
Data class	
j ($j = 1, 2, 3, \ldots, J$)	Data class
S_j	Volume of data class j in terms of KB of records
IO_j	Total IO count of data class j
R_j	Read volume of data class j (in KB)
W_j	Write volume of data class j (in KB)
Decision variable	
x_{ij}	Data of class j on y_i devices of type i in KB
y_i	The number of devices of type i

Putting these together, we get the dollar cost of installing storage system and its operation. The objective function to minimize is

$$\begin{aligned} Cost_{HybridStore} &= Cost_{Installation} + Cost_{Recurring} \\ &= \sum_{i=1}^{I} D\$_i \times y_i + \left(K_\$ \times \sum_{i=1}^{I} y_i \times \int_t P_i \, dt \right) \end{aligned}$$

The constraints as shown in Table 3, are related to (i) data groups, (ii) device's capacity, (iii) device's bandwidth, and (iv) life-time of the SSD. We describe each equation in Table 3 in detail. Equation (1) is the capacity constraint for the data classes and states that the sum of all the data belonging to class j partitioned across all I devices should be the same as the size of data class. Equation (2) is the capacity constraint for devices and states that the sum of data belonging to J classes on devices of type i should be less than the effective capacity of all the y_i devices. Equation (3) is the performance constraint for devices and states that y_i devices of type i should be

Table 3 Constraints of optimization formulation. Each equation in the above constraints illustrates different constraints: The declaration of variables used in the equations are described in Table 2

$$\sum_i x_{ij} = S_j \quad (\forall j \in J) \tag{1}$$

$$\sum_j x_{ij} \leq (U_i \times C_i) \times y_i \quad (\forall i \in I) \tag{2}$$

$$\sum_j Diff_{ij} \times x_{ij} \leq y_i \quad (\forall i \in I),$$

$$\text{where } Diff_{ij} = \frac{(IT_i \times IO_j + \frac{R_j}{RB_i} + \frac{W_j}{WB_i})}{(S_j \times T)} \tag{3}$$

$$L_i \geq LIFE \quad (\forall i \in I) \tag{4}$$

$$\sum_j (Wear_{ij} \times LIFE \times x_{ij}) \leq L_i \times y_i \quad (\forall i \in SSD_I),$$

$$\text{where } Wear_{ij} = \frac{(W_j/S_j)}{(\alpha \times E\text{-}UNIT_i)/T} \tag{5}$$

capable of handling the performance needs of J data classes placed on these devices. $Diff_{ij}$ refers to a difficulty factor which essentially computes the read bandwidth, write bandwidth and IOPS needs for data class j. Equation (4) is the lifetime constraint for devices and states that each device of type i should last at least the specified LIFE for which provisioning is being undertaken. Generally, storage reprovisioning is carried out every 3–5 years in a typical data center. HDDs are known to have more lifetime than this specified value and hence, this constraint typically degenerates into provisioning for SSDs, which have limited erase cycles. Equation (5) specifies this lifetime constraint for SSDs. $Wear_{ij}$ represents the wear-out factor for SSDs, i.e., the erase rate of blocks on a SSD. It is a function of the rate at which writes are done on a SSD and the amount of free space (pages) available after each erase. Amount of free space reclaimed depends on the amount of fragmentation prevalent on a SSD and α is used to capture this phenomenon. In the worst case, each block erase can result in only 1 free page whereas in the best case, we can reclaim all pages in a block. Thus, the value of α varies from ($W\text{-}UNIT_i/E\text{-}UNIT_i$) to 1.

5 Evaluation

5.1 Experimental setup

We developed the solver of HybridPlan using CPLEX, a well-regarded Integer Linear Programming (ILP) solver written in C [25]. *lp_solve* is a free (GNU licensed) linear programming solver based on simplex method. Also, we have written the trace analyzer for data classification in C. The source codes are less than 500 lines of code. The Solver execution time is extremely short (in seconds), however, the execution time of trace analyser for data classification is dependent on the trace size and can run into minutes for large traces.

Table 4 Description of synthetic workloads. The letter in parentheses denotes intensity of request's arrival rate, Low, Medium, and High

	Index	Read (%)	Size (KB)	Inter-arrival Time (ms)	I/O bandwidth	
					MB/s	IOPs
Sequential read	SR1	80	128	100 (L)	1.25	–
	SR2	80	128	2 (M)	62.5	–
	SR3	80	128	0.1 (H)	1,250	–
Random read	RR1	80	4	100 (L)	–	10
	RR2	80	4	2 (M)	–	500
	RR3	80	4	0.1 (H)	–	10,000
Sequential write	SW1	20	128	100 (L)	1.25	–
	SW2	20	128	2 (M)	62.5	–
	SW3	20	128	0.1 (H)	1,250	–
Random write	RW1	20	4	100 (L)	–	10
	RW2	20	4	2 (M)	–	500
	RW3	20	4	0.1 (H)	–	10,000

Table 5 Description of realistic traces

Workload	Size (TB)	Read (%)	Request size (KB)	IOPS
MSR trace	5.7 TB	68.1	23.32	823
Exchange server	750 GB	38.3	16.54	3,692

The main metrics used in our study include (i) storage installment cost (in $), recurring cost (in $) including operation cost of power consumption and cooling cost, (iii) the number of each type of devices, and (iv) the amount of each partitioned data class. We use a variety of synthetic and real-world enterprise scale storage traces to evaluate the effectiveness of our solver and the data classification process in provisioning storage.

Table 4 describes the characteristics of the synthetic workloads generated using Disksim [8], a well-regarded disk simulator capable of generating workloads based on certain input parameters. The synthetic workloads are divided into 4 categories, Sequential Read (SR), Sequential Write (SW), Random Read (RR), and Random Write (RW) with varying interarrival times. We used exponential distribution for varying the interarrival times and request sizes between subsequent requests. These workloads help in capturing the variations in the overall workload spectrum, which is not possible using a limited number of real-world traces. In order to present the application of our solver in a realistic setting, we use the MSR Cambridge traces [28] and MSR Enterprise Traces [29] to evaluate the effectiveness of our solver and the data classification process in provisioning storage. The details of these workloads are shown in Table 5.

Table 6 shows the characteristics of storage devices that we use in our evaluation. We considered three representative storage devices: high-speed HDD, low-speed

Table 6 Storage device characteristics. SLC and MLC are denoted by Single-Level Cell and Multilevel Cell, respectively

Device	Type	Cap. (GB)	Per-GB ($)	Util.	Read (MB/s)	Write (MB/s)	Lat. (ms)	Erase	Power (W)
Seagate Cheetah [37]	15 K HDD	146	1.80	0.8	171	171	3.6	–	12.92
Seagate Baracuda [36]	7.2 K HDD	750	0.17	0.8	125	125	4.2	–	9.4
Intel X25-E [14]	SLC SSD	32	11.96	0.5	230	200	0.125	100 K	2
Intel X25-M [15]	MLC SSD	80	3.22	0.5	220	80	0.25	10 K	2

HDD, and Flash based SSDs. They are all different in terms of their price, capacity, bandwidth, and power consumption. We use 146 GB 15 K RPM HDDs for high-end disks and 750 GB 7.2 K RPM HDDS for low-end disks. For SSDs, we use Intel SLC SSD and MLC SSD that are different in price, capacity, and performance. The details of devices are described in Table 6.

Device utilization ratio (ratio of amount of actual data stored in the device to its entire storage capacity) needs to be properly set in capacity planning. We set the expected utilization ratio of flash device as 50 % while that of hard disk drive is set as 80 %. This is based on the observation of Kgil et al. that garbage collection overhead in flash dramatically increases if the utilization exceeds 50 % [19]. Also, we have a similar observation in experiment using our flash simulator. The expected disk utilization is set as 50 % to provide sufficient storage space. For hard disk drive, since it is much cheaper in $/GB than flash device and most of cold data (rarely accessed) will be stored in the hard disk drive by HybridPlan, we set it as 80 %. Moreover, we need to consider device use duration in order to consider the recurring cost of the storage system. We used this period as 5 years for our evaluation. Note that 10 cents per kilowatt-hour (kWh) is used to estimate electricity cost in our evaluation.

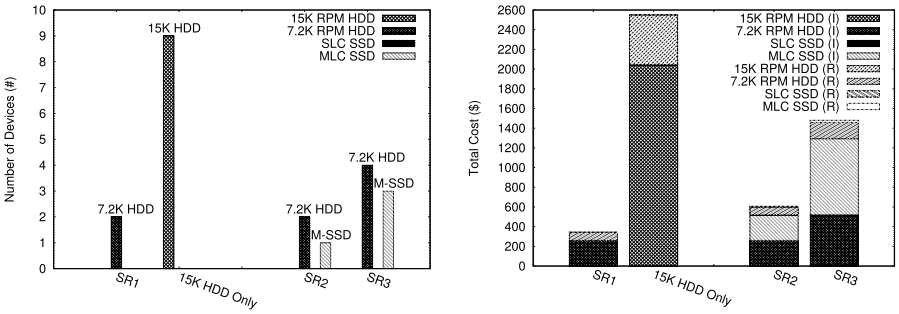
5.2 Evaluation of HybridPlan with synthetic workloads

In this subsection, we study the HybridPlan. In order to explore a wider range of workload characteristics with the HybridPlan, we developed synthetic workloads, the details of which are shown in Table 4.

5.2.1 Impact of I/O intensity

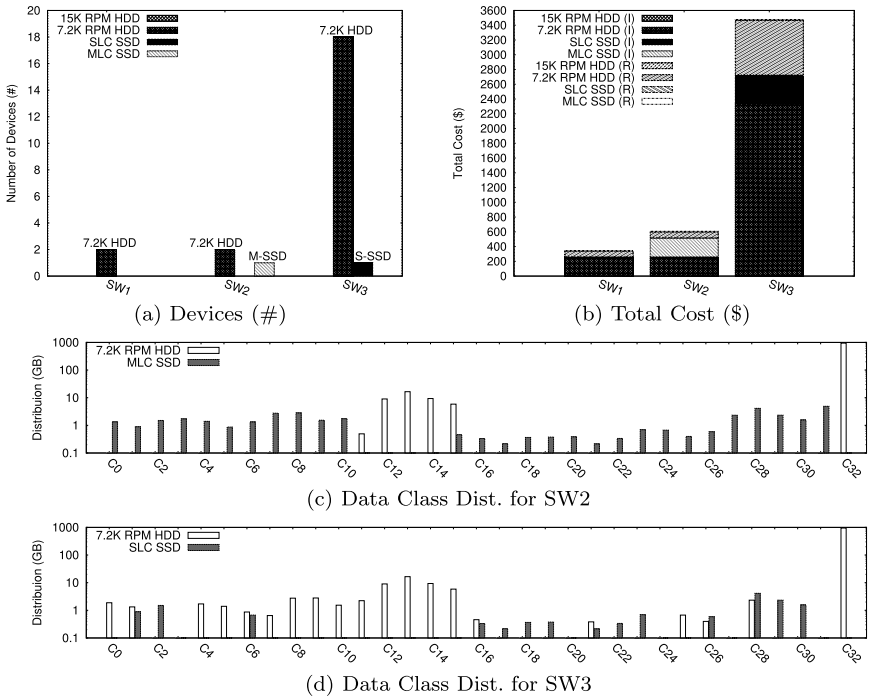
I/O arrival rate is a critical factor to determine how fast storage device needs. Figures 4 and 5 show the results about the impact of change in request arrival rate on the storage configurations provided by the solver. With increased I/O intensity, the number of devices increases as well as the type of devices needed to meet the I/O bandwidth requirements changes.

In Fig. 4(a), SR1 (sequential read only workload with low I/O intensity) only requires 2 slow HDDs. For fast 15 K RPM HDDs, it needs nine HDDs to satisfy the



(a) Devices (#) (b) Total Cost ($)

Fig. 4 Study the impact of I/O intensity in the read dominant workloads. In (**b**), "*I*" and "*R*," respectively, denote Installation Cost and Recurring Cost



(a) Devices (#) (b) Total Cost ($)

(c) Data Class Dist. for SW2

(d) Data Class Dist. for SW3

Fig. 5 Study the impact of I/O intensity in the write dominant workloads. (**c**) and (**d**) show data class distributions in *SW2* and *SW3* synthetic workloads

capacity demand, which requires much higher cost than 7.2 K RPM HDDs only (refer to the cost plot in Fig. 4(b). As we increase the I/O intensity, we observe the need for MLC SSDs to satisfy the bandwidth requirements with increased I/O intensity.

The choice of only slow HDD in SR1 clearly demonstrates that some workloads merely require storage for capacity and IOPS requirements for them are satisfied trivially. The same is corroborated by Narayanan et al. [29]. However, we contend

that even in these situations our solver plays the critical role of determining the right devices to meet the capacity needs. This is demonstrated in Fig. 4(b) (workload SR1) where choosing fast HDDs to meet the storage needs instead of slow ones would have resulted in 10 times increase in cost even though the system would have met the bandwidth requirements and not been over-provisioned. Furthermore, we observe that the recurring costs (in terms of power consumption by the storage devices) over the lifetime of the system are quite small as compared with the procurement costs of the devices. Thus, we observe that at least the direct power consumption by the devices is quite minuscule compared with other costs. The readers should note that we have not taken into account the indirect power consumption costs such as those due to cooling needs and other storage appliances (e.g., RAID controllers, SAN controllers, etc.).

Similar to the read-dominant workloads, we again observe the need for SSDs for write-intensive workloads to service the IOPS needs of the workloads in Fig. 5. However, there are certain subtle differences between the outputs for two workload categories.

For write-dominant SW3, we observe the solver including an SLC SSD instead of the MLC ones for its read-intensive counterpart (SR3). This is because SLC SSDs are about 2.5 times faster than the MLC ones (refer to Table 6), and hence more suitable for write-intensive workloads with high IOPS. Furthermore, we also observe a sharp increase in the number of slow HDDs with increased write intensity (SW3) in contrast to the rising number of MLC SSDs (SR3). This can be attributed to the vast $/GB difference between SLC SSDs and slow HDDs as shown in Table 6. Figure 5(c) and (d) show data class distributions for write dominant workloads (SW2 and SW3)

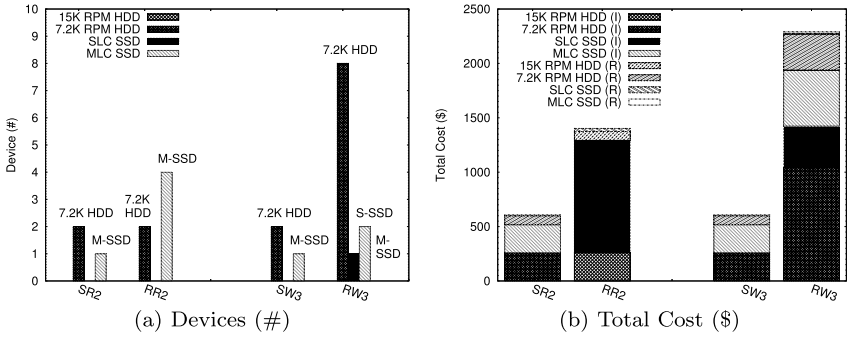
5.2.2 Impact of sequentiality

HDDs are known to perform better for sequential workloads because of reduced seek overhead as compared to the random workloads whereas SSDs are deemed to be primarily random access devices with good performance for both cases (especially for reads as random writes have been shown to have poorer performance comparatively [11, 23]). We explore the role of sequentiality on the decision making process of the solver in HybridPlan.

This is confirmed in Fig. 6(a) where we clearly observe the need for larger number of SSDs with increased randomness in requests even though the arrival rates remain the same. For read-dominant workloads, we see a 3-fold increase in the number of MLC SSDs to meet the IOPS requirements. This directly translates into a large increase in the overall cost of the storage system (Fig. 6(b)). As a consequence, even though the performance constraints for both iso-intensity sequential and random workloads are the same, the cost as well as the type of devices required for provisioning storage are quite different. Hence, as a storage administrator it is highly advisable to increase the sequentiality of incoming workloads.

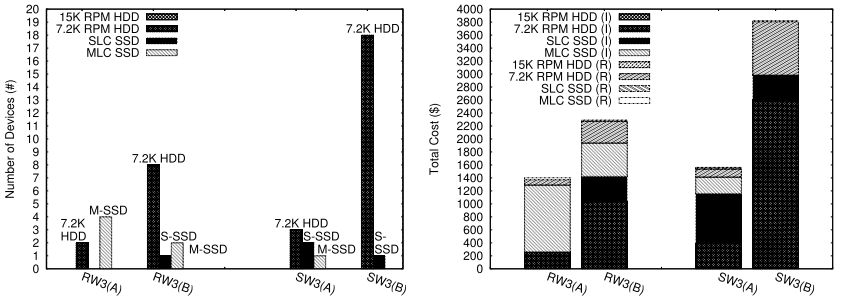
5.2.3 Impact of lifetime constraint

We have already established the importance of the lifetime constraint in capacity provisioning since its a long term decision made by a storage administrator. Figure 7



(a) Devices (#) (b) Total Cost ($)

Fig. 6 Study the impact of sequentiality in the workloads



(a) Devices (#) (b) Total Cost ($)

Fig. 7 Study the impact of lifetime constraints taken into account. "*A*" and "*B*" in the parenthesis denote "without lifetime constraint" and "with lifetime constraint," respectively

shows the difference in decision making with and without the lifetime constraint for both sequential and random write dominant workloads. The readers should note that we have only used write-intensive workloads because the lifetime of SSDs is directly dependent on block erases, which are caused by writes.

Without the lifetime constraint, we see a greater proportion of SSDs being used than with the lifetime constraint enforced. As shown in Fig. 7(a) for SW3, an MLC SSD is used along with 2 SLC SSDs when the lifetime constraint is removed. However, as soon as the constraint is applied, the solver outputs only 1 SLC SSD since MLC SSDs have lower erase count (lower lifetime for same number of writes) than SLC SSDs (Table 6) and would not be suitable in such an environment. Interestingly, the total number of devices as well as the overall costs (Fig. 7(b)) are much lower without the lifetime constraint. This is because a relatively cheaper MLC SSD is able to meet the IOPS needs whereas a large number of slow HDDs are needed to meet the performance guarantees when the lifetime constraint is obeyed. However, the cheaper configuration with MLC SSD may not have the needed longevity and the storage administrator might need to reprovision prematurely, thus increasing the overall costs over the initial estimated provisioning period.

5.2.4 Key lessons learned

From the above results, we note the following important observations across the workloads:

- The arrival rate of I/O requests in workloads is an important metric to evaluate the performance of back-end storage system. Overall, 15 K RPM HDD is surprisingly never recommended in the hybrid systems because 7.2 K RPM HDD is sufficiently cost-efficient. As the I/O arrival rate increases, the HybridPlan suggests to employ SSDs with HDDs and faster SSDs (SLC) in particular for write-dominant workloads for hybrid systems.
- SSD's performance is highly affected by the access pattern (denoted by sequentiality). SSDs are highly recommended for the random workloads. Specifically, SLC SSDs are recommended instead of MLC SSDs for write-intensive and random workloads.
- Lifetime is an important metric in evaluating the SSDs. We observed that when the lifetime of SSDs is of concern, the HybridPlan suggests to employ more HDDs than SSDs.

5.3 Evaluation of HybridPlan with Microsoft I/O traces

We use the MSR Cambridge traces [28] and Microsoft Exchange Server Traces [29] for realistic experiments. The MSR Cambridge traces are composed of several subtraces that have been collected in different directory for 7 days. Since each of these subtraces show very low I/O bandwidth demands, we consolidated the traces for aggregated bandwidth taken into account. Since the traces are too huge to run in the solver, the day of the highest I/O arrival rate—6th day trace has been run by the solver. Figure 8 show the results of time-series analysis of request arrival rates (in IOPS) for both workloads. Table 5 summarizes the characteristics of these real workloads used.

5.3.1 Can SSDs replace HDDs?

We examine if SSDs can actually replace HDDs at current price points and if not, then at what price points does it become viable to use a SSD only storage system. In this experiment, we see that HybridPlan can find the most economic storage composition for a workload, given the available device characteristics and their prices.

From the results in Fig. 9, we see that employing 7.2 K RPM HDDs is more economically efficient than employing 15 K RPM HDDs in the MSR Cambridge traces (Refer to Fig. 9(b)). 7.2K RPM HDD only system requires lesser HDDs than when we consider 15 K RPM HDD only system (refer to Fig. 9(a)). It is because I/O bandwidth requirement of this trace is not much higher than the I/O bandwidth that HDDs can provide. In Fig. 9(c), more than 99 % data are classified into C32; a data class storing data rarely accessed. In case of the SSD only system, it requires several hundreds of SSDs to satisfy the capacity requirement. We see again that a bounding factor for decision-making of HybridPlan is not I/O bandwidth requirement, but a storage capacity requirement. A similar observation can be found in [29] that the SSD only

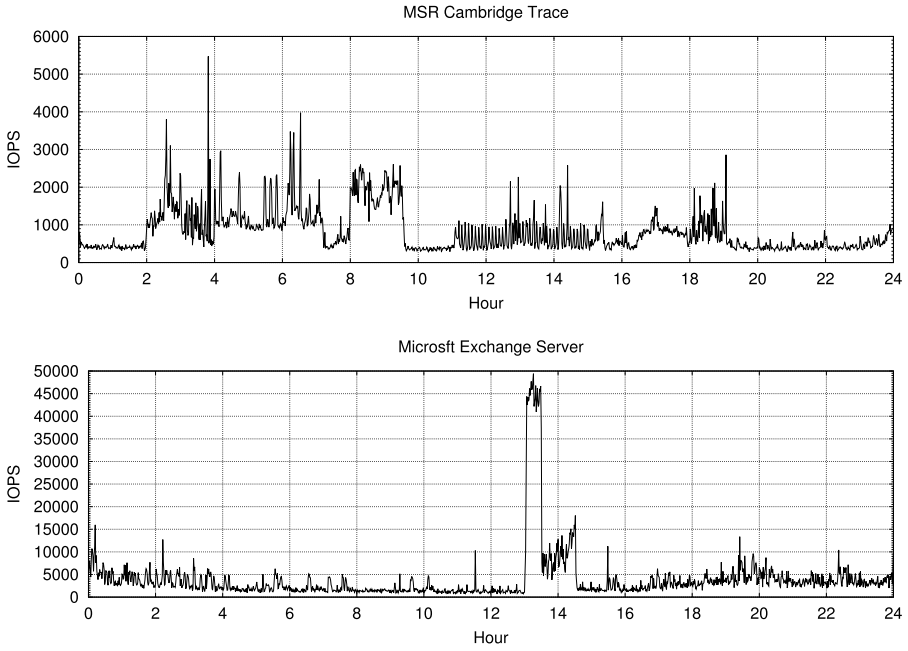


Fig. 8 Timeseries Analysis of MSR Cambridge and Microsoft Exchange Server traces

system is not an economically viable solution under current market prices of devices. We see similar observations in the Microsoft Exchange Server Trace. However, it requires a smaller number of devices in HDD only systems. It is also because of low I/O bandwidth requirement and larger capacity requirement (see Fig. 10(c)).

5.3.2 Efficacy of HybridStore

From Fig. 9(a), we see that HybridPlan can find the most economic storage composition for a workload, given the available device characteristics and their prices. In Fig. 9(a), HybridStore is composed of 10 × 7.2 K RPM HDDs and 1 MLC SSD by HybridPlan. This is a much cheaper configuration than any of the single device only systems. See the total cost of HybridStore with those of other storage configurations in Fig. 9(b). Total cost saving of HybridStore is about 85(%) compared to the high-end HDD only system. Similar to the MSR Cambridge traces, we again observe the efficiency of HybridStore for Microsoft Exchange Sever traces by HybridPlan. HybridStore of this trace is composed of 2 7.2 K RPM HDDs and 1 MLC SSD by HybridPlan. It also saves around 69 % compared to the 15 K RPM HDD only system. We see the data distribution of each data class for MSR Trace in Fig. 9(c). We see the data distribution of each data class for both traces respectively in Fig. 9(c).

Workload characteristics are known to show deviation from their normal behavior and with greater adoption of flash technology. The prices of HDDs and SSDs are also coming down. In this subsection, we examine how it finds the most economical combination of devices, while dealing with variation in device prices. Moreover, we investigate how does the recurring cost of devices affect the decisions by HybridPlan.

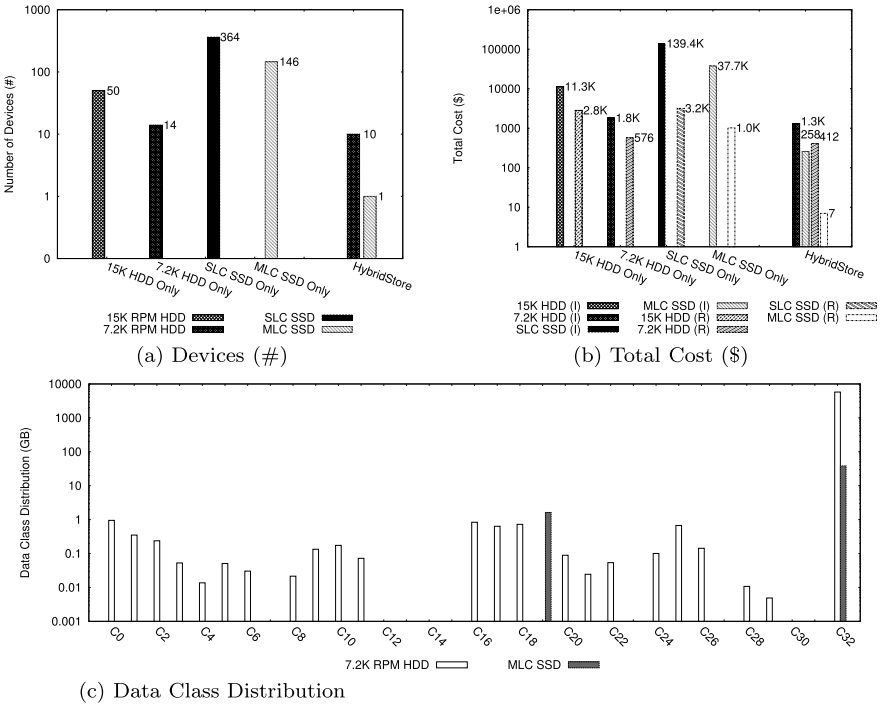


Fig. 9 Results of MSR Cambridge Trace

5.3.3 Impact of device prices

To allow price fluctuation, we varied the price of each device. Under current market price, 15 K RPM HDD and SLC SSD are relatively more expensive than 7.2 K RPM HDD and MLC SSD, respectively, Thus, we conducted hypothetical experiments by reducing the device prices of 15 K RPM HDD and SLC SSD from their prices in Table 6 and see how the HybridPlan operates for the Microsoft Exchange Server trace. We consider the following cases for price variation of devices:

- "Base (Baseline)" is when we use current market prices for devices as shown in Table 6.
- "A" is for when the price of SLC SSD becomes half.
- "B" is for when the price of 15 K RPM HDD becomes 35 % from their current market prices.
- "C" is for when both SLC SSD and 15 K RPM HDD become 50 % and 35 % from the current market prices.

As clearly shown in Table 7, the price variation of each device can impact on the decision of HybridPlan. In case of "A," we see that SLC SSD is employed instead of MLC SSD. In case of "B," the price-down, 50 % of 15 K RPM HDD does not change the result of HybridPlan from the baseline, however, in the case of "C," we see that it completely changes; it employs 1 15 K RPM HDD in addition to 1 7.2 K RPM HDD and 1 MLC SSD.

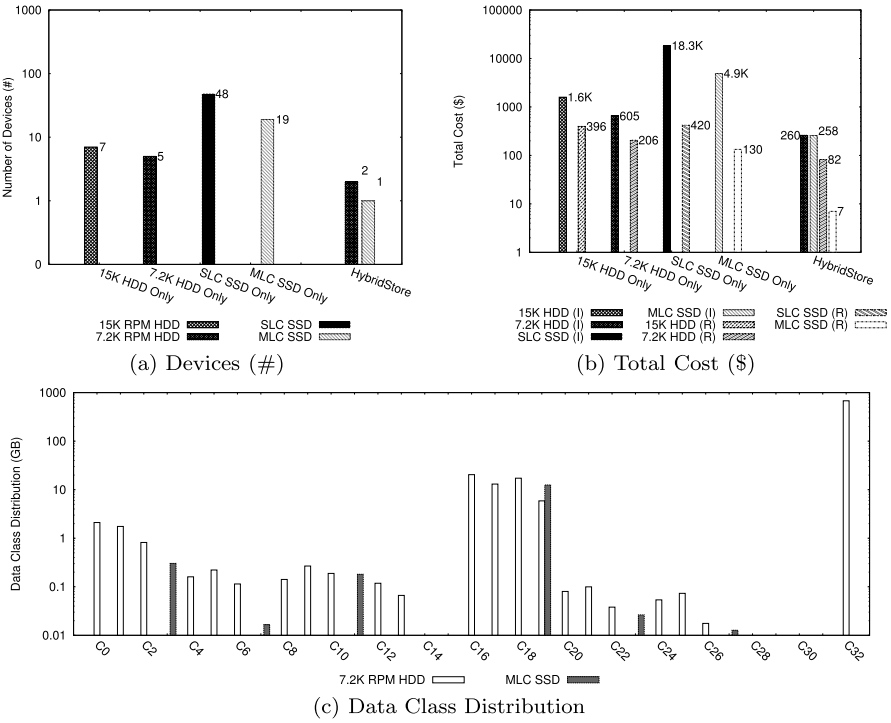


Fig. 10 Results of Microsoft Exchange Server Trace

Table 7 Price fluctuation of device

	HDD		SSD	
	15 K	7.2 K	SLC	MLC
Base	0	2	0	1
A	0	2	1	0
B	0	2	0	1
C	1	1	0	1

5.3.4 Impact of recurring costs

We study the impact on HybridPlan's decision making, when the recurring cost is not considered in HybridPlan and compare the results with Fig. 7 that includes the recurring cost along with the installation cost.

Table 8 which only includes the installation cost clearly demonstrates that recurring cost can play a significant role in the capacity planning process. In cases of "B" and "C," HybridPlan decides to use more 15 K RPM HDDs that those results in Table 7. Except for the baseline considering current market prices of devices, we see that HybridPlan suggests more HDDs than SSDs for all of the other cases, "A," "B," and "C." It is primarily because the recurring cost due to power consumption in HDDs is not taken into account by the decision-making of HybridPlan. Different for

Table 8 Recurring cost not taken into account

	HDD		SSD	
	15 K	7.2 K	SLC	MLC
Base	0	2	0	1
A	0	2	1	0
B	3	1	0	0
C	3	1	0	0

Table 9 Description of enterprise workloads used

Workload	Total volume size (GB)			Bandwidth (IOPS)			Read (%)
	Hot	Cold	Total	Avg.	95th	99th	
Financial [40]*	226 (8 %)	2,537	2,763	366.0	1449.7	1781.4	23
TPC-H [44]*	294 (32 %)	606	900	684.6	1312.6	1504.8	80

Table 10 Storage device characteristics

Type	RPM	Cap. (GB)	Price ($/GB)	IDR (MB/s)		Power (W)		Erase cycles
				Read	Write	Device	Supplement	
SSD [17]	–	80	4.25	220	160	1.0	1.8	10 K–1 M
High-end HDD [37]	15 K	300	1.50	128	128	9.19	17.0	–
Low-end HDD [33]	5.4 K	1 K	0.12	42.66	42.66	5.8	10.7	–

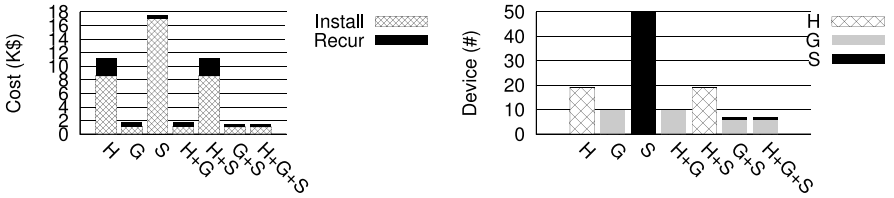
the work by Agrawala et al. in [29], here we show the importance of considering the recurring costs in storage resource provisioning.

5.4 Evaluation of HybridPlan with enterprise-scale workloads

We studied the HybridPlan with Microsoft workloads, however, I/O arrival rates of workloads are quite low, and they are read-dominant. Thus, we evaluated the HybridPlan more with enterprise-scale workloads that are different workloads characteristics than Microsoft workloads. Also, we use different storage devices such that we present the HybridPlan could operate well with various types of devices.

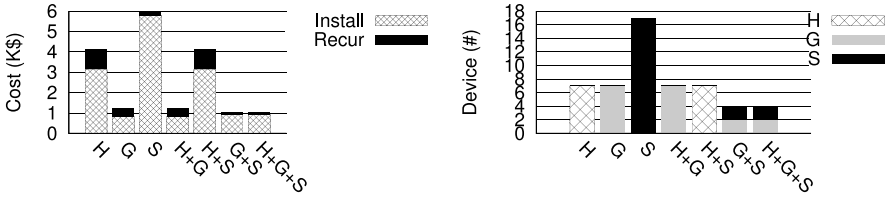
We employ the write-dominant I/O traces of an OLTP application running at a financial institution [40] made available by the Storage Performance Council (SPC), henceforth referred to as the *Financial trace*. TPC-H [44] is an ad hoc, decision-support read dominant benchmark (OLAP workload) examining large volumes of data to execute complex database queries. The summary of these traces used is shown in Table 9.

Table 10 shows the characteristics of storage devices that we use in our evaluation. We considered three representative storage devices: high-speed HDD, low-speed HDD, and Flash based SSD. They are all different in terms of their price, capacity, bandwidth, and power consumption. The low-end disk from Samsung is low-speed,



(a) Total Cost (Financial)

(b) Combination of Devices (Financial)



(c) Total Cost (TPC-H)

(d) Combination of Devices (TPC-H)

	H+(S)	(H)+G	S	(H)+G+S
Financial	7,806 (±2)	1,557 (±1)	258,274 (±64,487)	6,175 (±1,514)
TPC-H	3,696 (±9)	1,422 (±4)	86,678 (±49,704)	13,708 (±5)

(e) Performance results for different storage configurations (Financial and TPC-H)

Fig. 11 Economical comparison of various storage configuration employing multiple choices of devices by HybridPlan. H, G, and S in the above figures denote High-end HDD, Low-end HDD, and SSD, respectively. Note that H+S, H + G, and H + G + S show the same results as H, G, G + S, respectively. The values in parenthesis in (**e**) denote 95 % Confidence Interval (CI)

cheaper, and has higher capacity than the high speed 15 K RPM Seagate Cheetah disk. We use Intel's X25 4-way SSD as a representative SSD in our evaluation. Detailed description of these devices is given in Table 10.

5.4.1 Economical storage configuration

We first study the cost of unitype storage systems. Figure 11(a) and (b) show the results for Financial workloads. We observe that for high-end HDD only system, it needs 19 devices, and for low-end HDD only system, it needs 10 devices in order to meet the performance and capacity requirement of the workloads. The readers can find in Fig. 11(e) that these results of unitype storage systems could meet the performance and capacity requirements. Note that workload requirements to be met by the systems for Financial and TPC-H can be found in Table 9. The results summarize that with the high-end HDDs, the limiting factor in projecting the systems is the total capacity needed to store the data. On the other hand, with low-end HDDs based system, the device performance is the bottleneck (note that green HDDs are 5400 RPM devices) and requires a larger number of devices to service requests in parallel. Because of SSD's much higher price-per-unit ($/GB) than both HDDs, building a storage system with only SSDs is not economically viable. However, if the price difference with respect to High-end HDD falls under 2 times, employing only SSDs becomes an economic selection. Further, if the unit price of SSD falls by even more than 10 times

Table 11 "Static" partitioning versus HybridPlan in hybrid storage systems employing Low-end HDDs (denoted by HDD in the above table) and SSDs. "Static" in the above figures denotes a static data partitioning technique that entire hot data (whose IOPS is greater than 1) are stored in SSD pool and their remaining data (cold data) are stored in Low-end HDD pool. The results show that "Static" partitioning technique is over-provisioned whereas HybridPlan finds an economically optimal solution. The values in parenthesis in this above table denote 95 % Confidence Interval (CI)

	Technique	Total cost		Device		Hot data partition		IOPS
		Instal ($)	Recur ($)	HDD (#)	SSD (#)	HDD pool (%)	SSD pool (%)	
Financial	Static	2,420	990	6	5	–	100.0	25,827 (±6, 449)
	HybridPlan	1,060	891	6	1	95.0	5.0	6,175 (±1,514)
TPC-H	Static	2.280	436	2	6	–	100.0	30,592 (±17, 543)
	HybridPlan	920	338	2	2	83.0	17.0	13,708 (±5)

compared to its current unit price, SSD only system will be a possibility of becoming economic substitutes than using only Low-end HDDs. Next, we study the hybrid systems employing HDDs and SSDs from Fig. 11(a) and (b). Interestingly, we find that at current price points, a hybrid storage system consisting of 6 Low-end HDDs and 1 SSD is as economic as a Low-end HDD only system and SSD is not necessary to meet the Financial workload's requirements. We can see similar observation in TPC-H workload where a hybrid system comprising of 2 SSDs and 2 Low-end HDDs is equally economic as a 7 Low-end HDDs based system. However, as shown in Fig. 11(e), the hybrid system provides better performance than the corresponding Low-end HDD only system because of fast I/O processing from SSD. Thus, even though a complete SSD based storage system may still be a distance away in the future, partial replacement of HDDs is not only a viable alternative but provides higher performance at similar cost levels.

5.4.2 Static partitioning vs. HybridPlan

Until now, our experiments have shown that a hybrid system comprising Low-end HDDs and SSDs is the most economical selection in Financial and TPC-H workloads. In this subsection, we build on these results and show that HybridPlan is able to make economically efficient data partitioning and device combination decisions in the hybrid systems employing Low-end HDDs and SSDs. We compare HybridPlan with a static data partitioning technique in order to show the superiority of HybridPlan.

The static data partitioning technique places hot data on SSDs and the remaining data (cold) is stored on Low-end HDDs.

In Table 11, we see that HybridPlan can reduce the whole expense of the system by about 50 % or more in both Financial and TPC-H workloads. As seen from the results of device combinations for Financial trace in Table 11, a static data partitioning technique needs 6 HDDs and 5 SSDs whereas Mixplan provisions 6 HDDs and

1 SSD only. This combination results in significant reduction in installation cost of the system. HybridPlan makes this possible by moving inefficiently stored data on the SSDs by static partitioning into HDDs, thus reducing the total expense by only storing less than 10 % of the hot data on SSDs and moving the rest to HDDs. We see similar results for TPC-H where HybridPlan reduces the number of SSDs from 6 to 2. The last column in Table 11 shows how well HybridPlan satisfies the IO bandwidth requirements of workloads in comparison with static partitioning technique. As shown in Table 11, static partitioning causes a lot of over-provisioning, thus wasting most of the bandwidth. On the contrary, HybridPlan can minimize total expense by provisioning a smaller number of SSDs and still satisfying the bandwidth requirement of the workloads. that even HybridPlan seems to exceed the overall bandwidth requirement of the devices, but still causes much less wastage as compared with a static partitioning technique. Furthermore, some degree of bandwidth is wasted because a unit size of SSD used in our evaluation is 80 GB and the total capacity of the SSDs increases in units of 80 GB.

6 Conclusion and future work

This research was based on the emerging consensus among several storage experts that in the foreseeable future, with the exception of certain specialized domains, SSDs should be used as complementary devices to HDDs in problems in such a hybrid system employing HDDs and SSDs. We provide a general form of comprehensive methodology using a well-known technique for optimization problems, Linear Programming (LP). Based on this technique, we developed an capacity planner, called HybridPlan that finds the most economically efficient storage configuration while meeting the performance and lifetime requirements of SSDs and HDDs. As illustrative results, we showed that HybridPlan is able to find the most economical storage configuration: two 7.2 K RPM HDDs and one MLC SSDS for Microsoft Exchange Server trace and ten 7.2 K RPM HDDs and one MLC SSD for the consolidated Microsoft Cambridge trace. Moreover, we have not only studied that whether to consider recurring cost in HybridPlan can significantly impact on the decision-making of HybridPlan, but also we see that lifetime constraints are critical to protect the data of SSDs during useful lifetime of SSDs.

Workloads are known to exhibit variation from their predicted behavior. In such circumstances, capacity planning alone is not sufficient to meet the lifetime and performance budgets. With higher intensity of writes, the garbage collector is invoked more often; thus degrading the system's performance. Moreover, it results in higher number of block erases in flash, reducing the flash lifetime. Thus, we require additional sophisticated data partitioning mechanisms, which can dynamically adapt to these changing workload environments. We will explore techniques to meet the various budgets and work in HybridPlan for the future work.

Acknowledgements We would like to thank the anonymous reviewers for their detailed comments, which helped us improve the quality of this paper. This research was funded in part by NSF grant CCF-0811670. It was also supported in part by, and used the resources of, the Oak Ridge Leadership Computing Facility, located in the National Center for Computational Sciences at ORNL, which is managed by UT Battelle, LLC for the U.S. DOE (under the contract No. DE-AC05-00OR22725).

References

1. Agrawal N, Prabhakaran V, Wobber T, Davis JD, Manasse M, Panigrahy R (2008) Design tradeoffs for SSD performance. In: USENIX 2008 annual technical conference on annual technical conference, Berkeley, CA, USA, pp 57–70. USENIX Association
2. Bisson T, Brandt SA (2007) Reducing hybrid disk write latency with flash-backed I/O requests. In: Proceedings of the 2007 15th international symposium on modeling, analysis, and simulation of computer and telecommunication systems (MASCOTS). IEEE Computer Society, Washington, pp 402–409
3. Can flash memory become the foundation for a new tier in the storage hierarchy? http://www.acmqueue.com/modules.php?name=Content&pa=showpage&pid=547. Sept 2008
4. Charrap SH, Lu PL, He Y (1997) Thermal stability of recorded information at high densities. IEEE Trans Magn 33(1):978–983
5. Chen F, Koufaty DA, Zhang X (2009) Understanding intrinsic characteristics and system implications of flash memory based solid state drives. In: Proceedings of the eleventh international joint conference on measurement and modeling of computer systems, SIGMETRICS'09. ACM, New York, pp 181–192
6. Flash Price Drop Spurs Innovation. http://www.washingtonpost.com/wp-dyn/content/article/2008/02/01/AR2008020101313.html. Feb 2008
7. Gal E, Toledo S (2005) Algorithms and data structures for flash memories. ACM Comput Surv 37(2):138–163
8. Ganger GR, BucJonh JS, Bucyy S (January 2003) The DiskSim simulation environment version 3.0 reference manual
9. Guerra J, Pucha H, Glider J, Belluomini W, Rangaswami R (2011) Cost effective storage using extent based dynamic tiering. In: Proceedings of the annual conference on file and storage technology (FAST)
10. Gulati A, Merchant A, Varman PJ (2007) pClock: an arrival curve based approach for QoS guarantees in shared storage systems. In: Proceedings of the 2007 ACM SIGMETRICS international conference on measurement and modeling of computer systems, SIGMETRICS'07. ACM, New York, pp 13–24
11. Gupta A, Kim Y, Urgaonkar B (2009) DFTL: a flash translation layer employing demand-based selective caching of page-level address mappings. In: Proceeding of the 14th international conference on architectural support for programming languages and operating systems, ASPLOS'09. ACM, New York, pp 229–240
12. Gurumurthi S, Sivasubramaniam A, Natarajan VK (2005) Disk drive roadmap from the thermal perspective: a case for dynamic thermal management. In: Proceedings of the 32nd annual international symposium on computer architecture, ISCA'05. IEEE Computer Society, Washington, pp 38–49
13. HDD Technology Trends. http://www.storagenewsletter.com/news/disk/hdd-technology-trends-ibm. 2011
14. Intel. Intel X25-E extreme SATA solid-state drive. http://www.intel.com/design/flash/nand/extreme/index.htm
15. Intel. Intel X25-M and X18-M mainstream SATA solid-state drives. http://www.intel.com/design/flash/nand/mainstream/index.htmk
16. Intel. STMicroelectronics deliver industry's first phase change memory prototypes. http://www.intel.com/pressroom/archive/releases/20080206corp.htm
17. Intel X25-M SATA II, SSD, 80 GB. http://www.intel.com/design/flash/nand/mainstream/index.htm
18. International Data Group (2008) Datacenter SSDs: solid footing for growth
19. Kgil T, Roberts D, Mudge T (2008) Improving NAND flash based disk caches. In: Proceedings of the 35th annual international symposium on computer architecture, ISCA'08. IEEE Computer Society, Washington, pp 327–338
20. Kim Y, Gurumurthi S, Sivasubramaniam A (2006) Understanding the performance-temperature interactions in disk I/O of server workloads. In: Proceedings of the international symposium on high-performance computer architecture (HPCA), February 2006, pp 179–189
21. Kim Y, Gunasekaran R, Shipman GM, Dillow D, Zhang Z, Settlemyer BW (2010) Workload characterization of a leadership class storage. In: Proceedings of the 5th petascale data storage workshop supercomputing'10 (PDSW'10) held in conjunction with SC10 and sponsored by the DOE SciDAC petascale data storage ins, November 2010
22. Kim Y, Gupta A, Urgaonkar B, Berman P, Sivasubramaniam A (2011) Hybridstore: a cost-efficient, high-performance storage system combining ssds and hdds. In: Proceedings of the 2011 IEEE 19th

annual international symposium on modeling, analysis, and simulation of computer and telecommunication systems, MASCOTS'11, pp 227–236
23. Lee S, Moon B (2007) Design of flash-based DBMS: an in-page logging approach. In: Proceedings of the international conference on management of data (SIGMOD), August 2007, pp 55–66
24. Leventhal A (2008) Flash storage memory. Commun ACM 51(7):47–51
25. LP SOLVE: Linear Programming Code. http://lpsolve.sourceforge.net/5.5/
26. Marsh B, Douglis F, Krishnan P (1994) Flash memory file caching for mobile computers. In: Proceedings of the 27th Hawaii conference on systems science
27. Miller EL, Brandt SA, Long DDE (2001) HeRMES: high-performance reliable MRAM-enabled storage. In: HotOS, pp 95–99
28. Narayanan D, Donnelly A, Rowstron A (2008) Write off-loading: practical power management for enterprise storage. Transf Storage 4(3):1–23
29. Narayanan D, Thereska E, Donnelly A, Elnikety S, Rowstron A (2009) Migrating server storage to SSDs: analysis of tradeoffs. In: Proceedings of the 4th ACM European conference on computer systems, EuroSys'09. ACM, New York, pp 145–158
30. Rajimwale A, Prabhakaran V, Davis JD (2009) Block management in solid-state devices. In: Proceedings of the USENIX annual technical conference
31. Samsung. Samsung hybrid hard drive. http://www.samsung.com/Products/Semiconductor/Support/ebrochure/hddodd/hybrid_hard_drive_datasheet_200606.pdf
32. Samsung. Samsung 256 GB flash SSD with high-speed interface. http://www.i4u.com/article17560.html. May 2008
33. Samsung "Green" 1 TB serial ATA 3 Gbps internal hard drive. http://www.iconocast.com/EB0000000000000030/R4/News1.htm
34. Schirle N, Lieu DF (1996) History and trends in the development of motorized spindles for hard disk drives. IEEE Trans Magn 32(3):1703–1708
35. Schroeder B, Gibson GA (2007) Understanding disk failure rates: what does an MTTF of 1,000,000 hours mean to you? In: Proceedings of the annual conference on file and storage technology (FAST), February 2007
36. Seagate. Seagate barracuda 7.2 K. http://www.seagate.com/www/en-us/products/desktops/barracuda_hard_drives/
37. Seagate. Seagate cheetah 15K.5. http://www.seagate.com/www/en-us/products/servers/cheetah/
38. Shimada Y (2008) FeRAM: next generation challenges and future directions. In: Electronics weekly, May 2008
39. Smullen CW, Mohan V, Nigam A, Gurumurthi S, Stan MR (2011) Relaxing non-volatility for fast and energy-efficient STT-RAM caches. In: Proceedings of the international symposium on high performance computer architecture (HPCA)
40. Storage-Performance-Council. OLTP trace from UMass trace repository. http://traces.cs.umass.edu/index.php/Storage/Storage
41. Tehrani S, Slaughter JM, Chen E, Durlam M, Shi J, DeHerren M (1999) Progress and outlook for MRAM technology. IEEE Trans Magn 35(5):2814–2819
42. Uysal M, Merchant A, Alvarez GA (2003) Using MEMS-based storage in disk arrays. In: Proceedings of the USENIX conference on file and storage technologies (FAST), Berkeley, CA, USA, pp 89–101. USENIX Association
43. Wu M, Zwaenepoel W (1994) eNVy: a non-volatile main memory storage system. In: Proceedings of the international conference on architectural support for programming languages and operating systems (ASPLOS), pp 86–97
44. Zhang J, Sivasubramaniam A, Franke H, Gautam N, Zhang Y, Nagar S (2004) Synthesizing representative I/O workloads for TPC-H. In: Proceedings of the international symposium on high performance computer architecture (HPCA), pp 142–151