Performance Modeling and Analysis of Flash-based Storage Devices

H. Howie Huang and Shan Li

Department of Electrical and Computer Engineering
George Washington University
{howie, shallia}@gwu.edu

Alex Szalay and Andreas Terzis

Department of Computer Science
Johns Hopkins University
{szalay, terzis}@jhu.edu

***Abstract*—Flash-based solid-state drives (SSDs) will become key components in future storage systems. An accurate performance model will not only help understand the state-of-the-art of SSDs, but also provide the research tools for exploring the design space of such storage systems. Although over the years many performance models were developed for hard drives, the architectural differences between two device families prevent these models from being effective for SSDs. The hard drive performance models cannot account for several unique characteristics of SSDs, e.g., low latency, slow update, and expensive block-level erase. In this paper, we utilize the black-box modeling approach to analyze and evaluate SSD performance, including latency, bandwidth, and throughput, as it requires minimal *a priori* information about the storage devices. We construct the black-box models, using both synthetic workloads and real-world traces, on three SSDs, as well as an SSD RAID. We find that, while the black-box approach may produce less desirable performance predictions for hard disks, a black-box SSD model with a comprehensive set of workload characteristics can produce accurate predictions for latency, bandwidth, and throughput with small errors.**

I. Introduction

Today, flash-based solid-state drives (SSDs) have appeared in a wide spectrum of computer systems, from mobile computers where SSDs provide low power consumption and resist rough handling, to enterprise class server and storage where SSDs promise high data transfer rate and low access latency. Because SSDs emulate the hard disk interfaces (SCSI and SATA), they are treated as block-level storage devices by host computers. In theory, one can simply replace every disk drive with an SSD. However, this approach of hard disk drive (HDD) replacement does not automatically provide improved performance because traditional operating systems and applications were optimized for spinning magnetic platters [1] [2]. For SSDs, time-sensitive and I/O-intensive applications are often considered as good candidates. For example, Online Transaction Processing (OLTP) systems can benefit greatly from caching a small portion of the databases on SSDs to achieve high IOPS (I/O per second) to the "hot data" (most frequently accessed). As SSDs become key components in future storage systems, we believe that an accurate performance model will not only help understand the state-of-the-art of SSDs, but also provide the research tools for exploring the design space of such storage systems.

In this paper, we utilize the black-box modeling approach to analyze and evaluate SSD performance, including latency, bandwidth, and throughput, extending our prior work [3]. This approach is attractive because it requires limited *a priori* information about a storage device. For SSDs, this is especially beneficial, because SSD vendors are reluctant to reveal the design details in order to protect intellectual properties. Prior research showed that the black-box modeling can give a reasonable performance prediction for hard disks [4] [5]. To build a good performance model for SSDs, one needs to recognize that vast architectural differences exist between hard drives and solid-state drives, e.g., page-level reads and writes, out-of-place updates, and block-level erase operations that can lead to slow random writes in a solid-state drive [6] [7]. In this work, we first build a basic black-box model with traditional workload characteristics, e.g., read/write ratio and request size. As we will demonstrate in this paper, although a basic model works better for SSDs than hard drives, the prediction accuracy remains unsatisfactory. To address this problem, we investigate several additional aspects of the workloads in a systematic manner. Specifically, we add the write stride (for the effect of request alignments), split the request size into read and write sizes (because of SSD asymmetric read/write performance), and change the randomness into read and write randomness (that have different impacts on the SSD performance). In addition, we study eight workload-specific models for corner cases (e.g., read only, random only, etc.). By doing this, we believe that the models can further reveal the importance of each individual workload characteristic in the context of the SSD models.

To construct the black-box models, we collect a large number of the training data on workload parameters and device performance, where the value of the latter is predicated as a function of the former. Our approach applies statistical machine learning algorithms for model fitting. We evaluate the models with one hard disk and threes SSDs using the microbenchmarks, as well as four real-world I/O traces. The results are encouraging - the mean relative errors of an SSD model are as low as 9% for the latency prediction, 6% for bandwidth and throughput, and less than 1% for the workload-specific models.

Our main contributions in this work are two-fold:

- We analyze a number of different workload characteristics for SSDs and demonstrate that the traditional models designed for hard drives are ineffective in this case. We

978-1-4577-0428-4/11/$26.00

propose an extended model to properly correlate the SSD performance and I/O requests, and further investigate the models for each specific data access pattern.

- Although the black-box performance models are not new for hard drives, they are known unsatisfactory. In this paper, we show that this approach can work very well on a variety of SSDs, that is, the model produces accurate predictions under a collection of different workloads. To the best of our knowledge, this work is the first attempt in developing the black-box performance models that match the real-world SSDs, which we believe will help design and utilize flash based storage systems.

The remainder of this paper is organized as follows. Section II presents the background on flash memory and SSDs. Section III describes our approach for constructing the black-box performance models for SSDs, and Section IV evaluates the models through the microbenchmarks and real-world traces. We present the related work in Section V and conclude in Section VI.

II. Background

A. Flash Memory

NAND flash memory is an increasingly popular choice for the secondary storage, due to its low cost and high density. Each NAND flash package contains a small number of dies where digital logic gates (memory cells) are grouped into blocks (e.g., 256KB). A block is further divided into a set of uniform pages (e.g., 2KB and 4KB). NAND flash memory supports three kinds of operations: read, write (program), and erase. Data reads and writes are performed at the page granularity, but erases are done at the block granularity. Page writes can only be performed to an erased block, that is, a page becomes available for writes only after the entire block is erased. In a typical flash [8], a read operation completes in 25 microseconds and a write operation in 200 microseconds. In contrast, erase operations take considerably longer time in 1.5 milliseconds.

Because the nature of NAND flash prevents in-place updates, flash memory utilizes out-of-place writes, that is, an update to an existing page is written to a new location and the old page is marked invalid. If every page in a block becomes invalid, then the block can be erased. This process is called garbage collection. When needed, valid pages in a block can be copied to new locations to make the block invalid and ready for garbage collection. Flash wear-leveling that ages memory cells evenly is crucial, because unlike hard disks, flash memory wears out - each block can only be erased for a finite number of cycles (e.g., 100,000 to 1 million). Wear-leveling can be achieved by carefully selecting an obsolete block, copying valid pages, updating map structure, and erasing the block.

B. Solid-State Drives

Fig. 1 depicts a typical SSD architecture. From the outside, a solid-state drive resembles the form factor (2.5 or 3.5 inches)

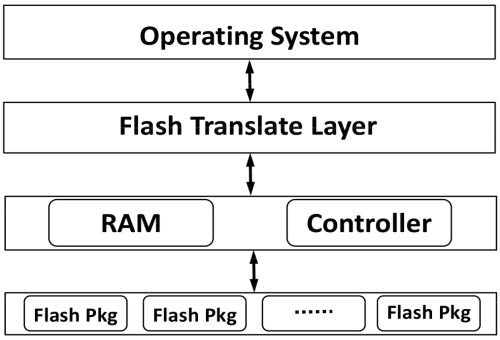


Fig. 1. SSD Architecture

and has the same interface as a hard drive. Internally, an SSD is very different from a hard drive that consists of rotating platters, associated disk heads, and arms. Simply put, an SSD includes a recording mechanism that consists of several NAND flash packages and a controller that implements the control logic. Although SSDs eliminate moving mechanical parts, the limited number of writes that can be done per cell casts doubt on their life cycles. The read/write/erase behaviors of SSDs require careful flash management and data placement in order to provide high throughput and good lifetime. Compared with hard drives whose performance is approximate to rotational speeds, SSDs deliver asymmetric read/write performance that is highly dependent on vendors and workloads [13] [14]. How to model this behavior is one of questions that we aim to answer in this paper.

Internally, the SSD controller contains a processor, a RAM (cache buffer), and the host interface logic that becomes the flash translation layer (FTL). The FTL mimics a hard disk and manages the mappings from logical block addresses (LBA) to physical flash locations. The FTL is essential to SSDs since the nature of NAND flash prevents in-place updates. Basically, the block map can be a direct map that contains the pointers from a logical block to a physical page, an indirect map, or a combination of both. For example, N. Agrawal et al. [6] assumed that the indirect map is stored in the flash while the direct map is reconstructed during the initialization and kept in volatile memory during run time. As a widely used technique designed for NAND flash memories, NFTL (NAND FTL) [15] [16] utilizes map blocks for storing direct map within the flash.

In this paper, we analyze three SSDs, including Intel X-25M (SSD_I) [10], OCZ Apex (SSD_A) [11] and Samsung (SSD_S) [12]. In comparison, we use a 5,400RPM hard disk, Samsung Spinpoint M7 (HDD_S) [9]. Table I lists the specification numbers for all the devices. The hard disk specification [9] does not give the specific read and write bandwidths, but the numbers are expected to be lower than those of SSDs. Note that SSD_S is one of the early generation flash devices, which may contribute to its lower performance.

III. Black-Box Performance Modeling for SSDs

The goal of our performance modeling for SSDs is to build a black-box model that can be used to predict the performance of a given SSD through its workload characteristics.

TABLE I
SPECIFICATION COMPARISON FOR SSDS AND HARD DRIVE

	HDD_S [9]	**SSD_I** [10]	**SSD_A** [11]	**SSD_S** [12]
Capacity	500GB	80GB	120GB	32GB
Buffer Size	8MB	Unknown	64MB	Unknown
Read Bandwidth	-	250MB/s (seq)	250MB/s	100MB/s (seq)
Write Bandwidth	-	70MB/s (seq)	100MB/s (sustained)	80MB/s (seq)
Latency	5.6ms (avg)	85μs (Read) 115μs (Write)	$< 100\mu$s	-

In our approach shown in Fig. 2, a black-box model can be constructed in two steps: 1) benchmark an SSD and collect the training data that consist of the model inputs (workloads characteristics) and outputs (performance metrics); and 2) utilize the statistical methods to quantify the correlations between the inputs and outputs, construct and validate the model. The rational is that the performance tends to be highly correlated with the workload characteristics. For example, SSD latency and throughput fluctuate when the percentage of write requests, the number of random requests, and the outstanding I/O requests vary.

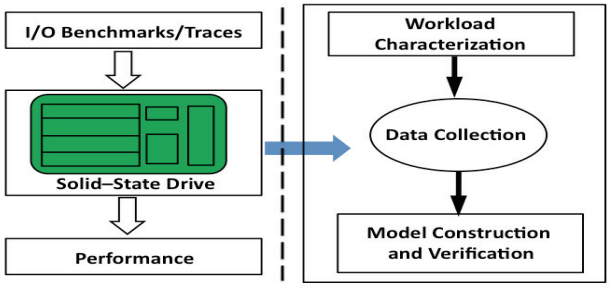


Fig. 2. Black-box performance modeling for SSDs

In this section, we begin with a basic black-box model that has been used for hard drives and build our extended models to include additional workload characteristics that have large influences on the SSD performance. Furthermore, we introduce eight workload-specific models to help predict SSD performance under the pre-defined scenarios.

A. Basic Model

A typical black-box model predicts the performance as a function (F) of workload characteristics. This model takes the workload characteristics (wc) as input parameters and outputs the predicted performance metric (p), which can be formally written as in equation 1:

$$p = F(wc). \tag{1}$$

The workload is defined as a stream of I/O requests. Typically when modeling a hard drive, the workload can be characterized by read and write ratio (*rw_ratio*) that is defined as the percentage of writes in the request, request size (*req_size*) that represents the number of bytes transferred to/from the storage device, queue depth (*q_dep*) that represents the number of outstanding I/Os, and request randomness (*rand*) that is defined as the percentage of random accesses in the I/O request stream. Thus, a single workload wc can be represented as a vector of workload characteristics as shown in equation 2:

$$wc =< wr\_ratio, q\_dep, req\_size, rand > . \tag{2}$$

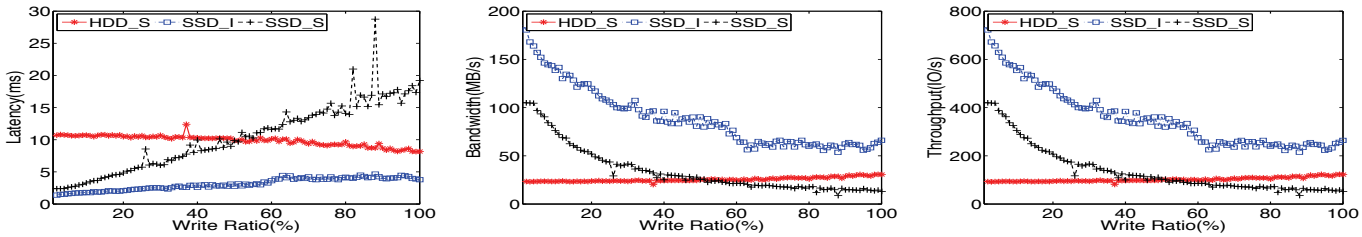
In this paper, we focus on three performance metrics: latency (*lat*), bandwidth (*bw*), and throughput in IOs per second (*iops*). Thus, the performance p can be represented as either of three metrics shown in equation 3:

$$p = lat|bw|iops. \tag{3}$$

The function F in equation 1 will be different for each metric and is expected to vary for different SSDs. Also, F can be represented in various forms. In this paper, we utilize a regression tree which will be introduced shortly.

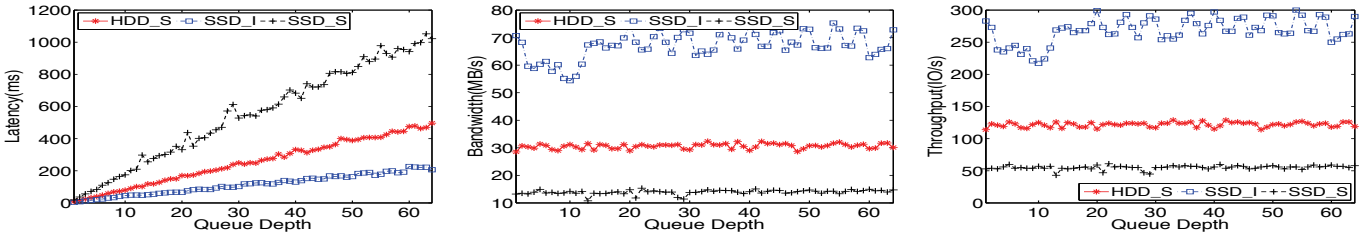
First, we want to study whether the four workload parameters in the basic model are effective for predicting the SSD performance. To this end, we conduct a number of experiments to study the effectiveness of this model (the experiment setup is described in detail in Section IV). We start with the impacts of the write ratio and queue depth on the latency, bandwidth, and throughput of the solid-state drives. Fig. 3 and 4 plot the results for three drives, SSD_I, SSD_S, and HDD_S. In the tests, we change one parameter each time while using the default values for the others. When a parameter is not being analyzed, the default values for write ratio, queue depth, read size and write size are 0% for read (or 100% for write), 1, 256KB, and 256KB, respectively. All tests use 100% randomness for both writes and reads, which attributes to large latencies in the figures.

As shown in Fig. 3, it is not surprising to find that two solid-state drives outperform the hard drive, especially when dealing with a lot of read requests. For example, when 90% of requests are reads, the latency on SSDs are about 2 milliseconds and the bandwidth is 100 - 150MBs. In this case, the latency on the hard drive is five times as much and the bandwidth is around 20MB/s. From the figure, one can clearly see that the write ratio has a large influence on all three performance metrics for SSD_I and SSD_S. As there are more writes in the workload, the latency increases and both the bandwidth and throughput decrease. As we mentioned before, SSDs have asymmetric performance for reads and writes, that is, fast read and slow write. In contrast, the hard drive HDD_S shows small changes on three performance metrics. For all three categories, SSD_I outperforms the other two devices, i.e., achieves low latency,



(a) Write Ratio vs. Latency (b) Write Ratio vs. Bandwidth (c) Write Ratio vs. Throughput

Fig. 3. Impacts of write ratio on latency, bandwidth, and throughput



(a) Queue Depth vs. Latency (b) Queue Depth vs. Bandwidth (c) Queue Depth vs. Throughput

Fig. 4. Impacts of queue depth on latency, bandwidth, and throughput

high bandwidth and throughput in most cases. One can suspect that the higher performance of SSD_I comes from a good design of control logic and possibly a larger internal buffer. Note that an SSD is not necessarily better than a hard drive. When there are more writes, HDD_S actually outperforms SSD_S, indicating an inferior design of this early generation SSD. In Fig. 4, we can see that the queue depth significantly affects the latency for the three devices, with SSD_I being the best and SSD_S the worst. The impact of the queue depth on the bandwidth and throughput are minimal, once both reach the saturation state.

B. Extended Model

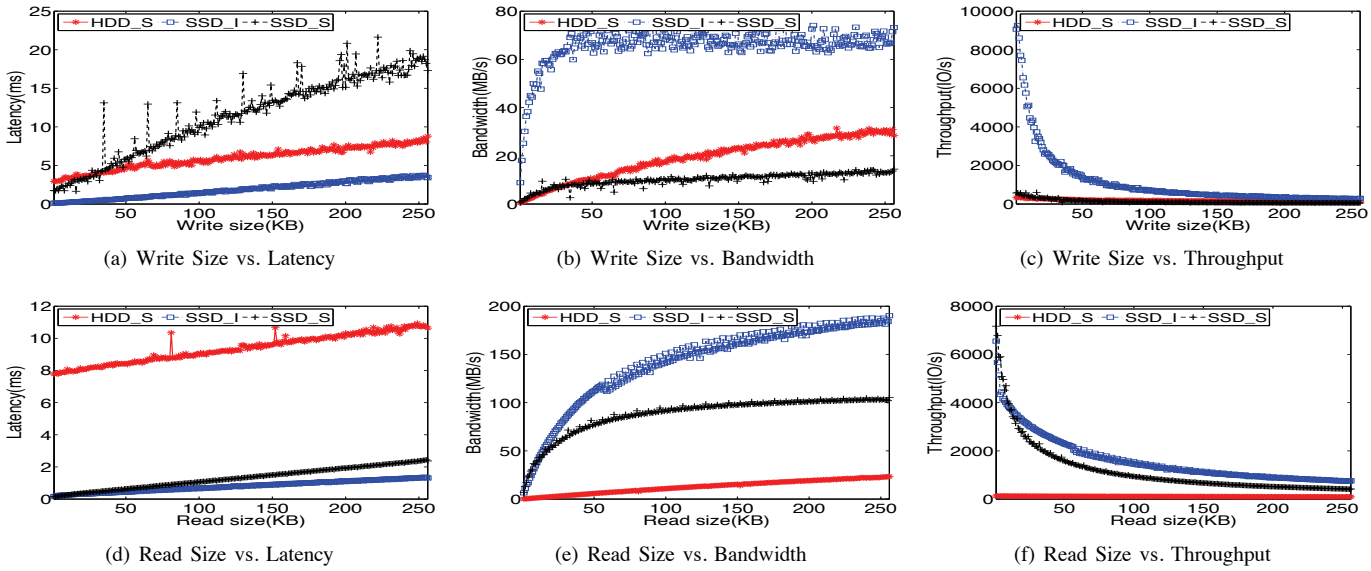
Clearly, the four parameters in the basic model, *wr_ratio*, *q_dep*, *req_size*, and *rand*, remain critical in capturing the correlation between the workloads and SSD performance. Now given the architectural differences between hard drives and solid-state drives, we want to further examine the following questions:

1) Are the four basic workload characteristics sufficient to characterize the I/O workloads in a statistically significant matter? Is there a need for additional workload characteristics?
2) Is the relationship between workload characteristics and SSD performance predictable? How accurate will the predictions be?

To answer these questions, we take into consideration several new parameters: read and write stride for the effect of request alignments, read and write size because of SSD asymmetric read/write performance, and read and write randomness that can also have varied impacts on the SSD performance.

In Fig. 5, we divide the request size into read size (*rd_size*) and write size (*wr_size*), since reads and writes are asymmetric in SSDs. As the write size varies from 1K to 256K, one can see that the latency increases for all three devices, where one can clearly observe a linear trend. It is worthy to note that SSD_S has slower writes than HDD_S, which again suggests that a better FTL is needed to improve the write performance for this device. When the read size changes from 1K to 256K, the latency and bandwidth also increase linearly, albeit at a slower rate, and the throughput decreases. Both SSDs have clear performance advantages for reads. The changes from read size are device-dependent - two SSDs read in a similar speed but differ greatly on write speed. Note that while for bandwidth and throughput, two SSDs (especially SSD_I) present some degrees of nonlinearity, the evaluation results will show that they can be reasonably captured by the extended models.

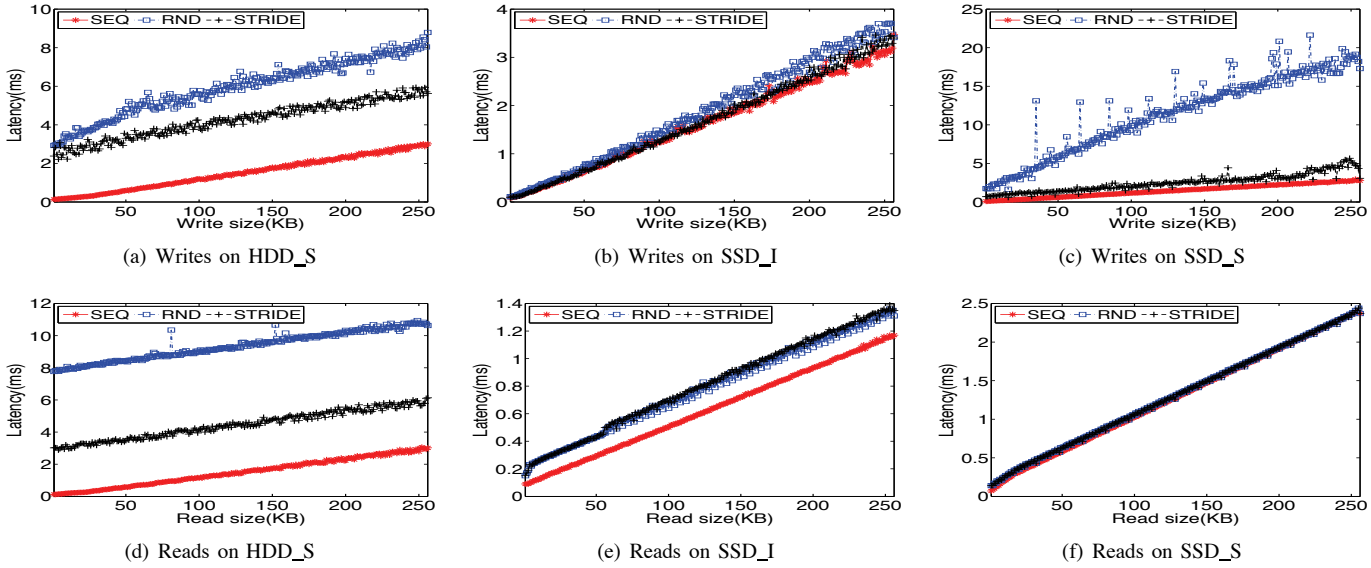
It is well known that hard disks have much better sequential performance than random access. In contrast, SSDs are generally considered having comparable performance, including latency, for sequential and random access. To examine how different access patterns influence the SSD performance, here we evaluate three different access patterns, including sequential, random, and stride (write stride, *wr_stride*, and read stride, *rd_stride*) that defines the number of bytes between two consecutive reads and writes, respectively. We say that an I/O request is a stride access when there exists a common distance between the end and start of successive accesses. We examine all three devices and the experiment results are collected when write or read size changes from 1KB to 256KB under each access pattern. Fig. 6 shows the impacts of the access patterns



(a) Write Size vs. Latency (b) Write Size vs. Bandwidth (c) Write Size vs. Throughput

(d) Read Size vs. Latency (e) Read Size vs. Bandwidth (f) Read Size vs. Throughput

Fig. 5. Impacts of request size on latency, bandwidth, and throughput



(a) Writes on HDD_S (b) Writes on SSD_I (c) Writes on SSD_S

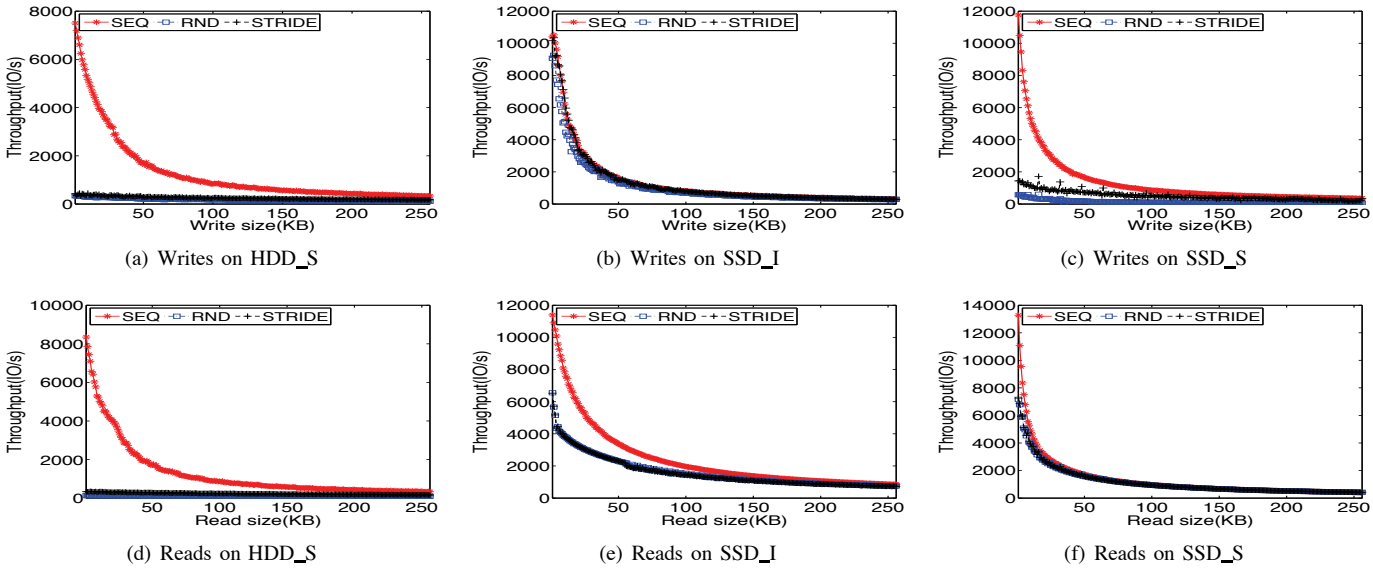
(d) Reads on HDD_S (e) Reads on SSD_I (f) Reads on SSD_S

Fig. 6. Impacts of access patterns on latency

on the latency. All three devices present different performance under sequential and random access, even for two SSDs that are believed to have a similar performance. In particular, when the write size changes under random access pattern, both SSDs especially SSD_S suffers performance degradation, which can be caused by expensive erases and out-of-place writes. At the same time, stride access shows a unique performance for all three devices. For writes, stride access on both SSD_I and SSD_S performs close to sequential writes, and for reads the difference between stride and random reads is small. For the hard drive, the performance of stride access is clearly between sequential and random access. The impacts of various access patterns on throughput are shown in Fig. 7. While we expect that sequential access on the hard drive is order of magnitude better, it is very interesting to see that even on SSDs sequential access tends to outperform random access to some extent.

These observations inspire us to compose a model with an extended set of workload characteristics. Now, a workload *wc* can be formally written as a vector of workload characteristics shown in equation 4:

$$wc =< wr\_ratio, q\_dep, wr\_size, rd\_size, \\ wr\_rand, rd\_rand, wr\_stride, rd\_stride > . \quad (4)$$



(a) Writes on HDD_S (b) Writes on SSD_I (c) Writes on SSD_S

(d) Reads on HDD_S (e) Reads on SSD_I (f) Reads on SSD_S

Fig. 7. Impacts of access patterns on throughput

C. Workload Specific Model

In the above tests, the SSDs have showed the similarity in I/O performance, e.g., read throughput that is orders of magnitude better than HDDs, as well as several interesting differences, e.g., slower writes on SSD_S. To understand the problem further, we feel that it is important to study each SSD under a specific workload. In this work, we explore the SSD performance models for eight special cases: read only, write only, random access, sequential access, random read, random write, sequential read, and sequential write. Here we continue to use the extended black-box model. The only difference is that the workload reflects only one type of access pattern each time. These models can help study SSD performance for each unique request type, as well as the overall performance when given a mix of various requests. For example, if one wants to upgrade a storage system for a new application, one shall first profile which kind of workloads are most popular, and examine whether a flash-based storage system will likely provide performance benefits. Prior work [4] [5] [17] did not consider these types of special cases.

D. Regression Tree

To construct a black-box model, we need to collect the training data that consist of workloads characteristics and the corresponding performance of a storage device. The training data shall be representative (to span a wide spectrum) and sufficient (to have an adequate number of the tests). The next challenge for our black-box approach is to use statistical machine learning algorithms to capture the mapping between workload characteristics (*independent variables*) and performance metrics (*dependent variables*). In our approach, given the training data as the input, the regression algorithm is applied to calculate a predictive function that maps the input to the desired output. The linear regression represents the relationship between *independent variables* and *dependent variables* with a linear model. After using the least-squares approach to fit the performance model, the linear regression can make a performance prediction based on a given set of workload characteristics.

Specifically, we construct a regression tree from the regression function, which is generated by recursively splitting the input *independent variables* into leaf nodes using a binary sequence. The leaf nodes of the tree provide the predicted values for *dependent variables* as a constant function of *independent variables*. We follow three steps to build the regression tree: 1) select an algorithm to split an intermediate node; 2) determine when we should terminate a tree node; and 3) generate a value for each leaf node. The node-splitting process is applied iteratively. The best split for a linear regression tree is to minimize the mean square error among all training data at the leaf nodes. That is, the split results in the smallest difference between each data point and the mean of all training data that are represented by the leaf nodes. Two splitting strategies can produce two different regression trees for the same data. An example of a regression tree is shown in Fig. 8. The tree splits at *wr_rand* at the first level, and *q_dep* at the second level, and so forth, till it terminates to make a prediction of the bandwidth as a function of the workload characteristics that are listed on the path. In this research, we employ the least-squares multilinear regression [18] to build our performance models. The models are trained and tested using both synthetic workloads and real-world traces, and all the observations from the training data are used for fitting and validation. In addition to the least-squares approach, we also test the quantile regression technique [19] for model fitting and get similar performance for our models.

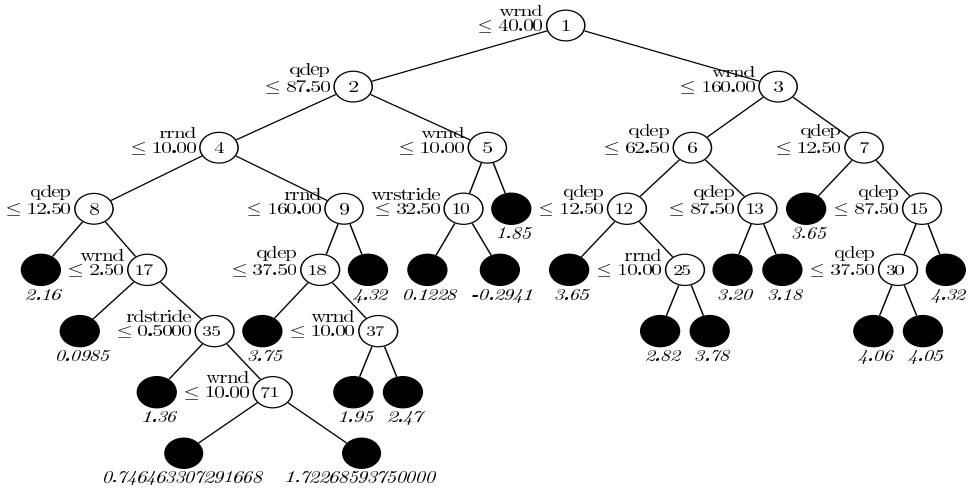


Fig. 8. An example of regression tree for bandwidth prediction. Note that here a negative prediction at a leaf node shall be considered as invalid.

IV. Evaluation

A. Experiment Methodology

The experiments are run on the desktop machines with Intel Core 2 Duo 2.93 GHz, 4GB memory, and Linux kernel 2.6. Our training data are collected from five storage devices shown in Table I, three SSDs and one HDD. For each device, three types of black-box models are trained and tested: basic models, extended models, and workload specific models. We also test with the Amdahl cluster [20] at the George Washington University, where each node has dual-core 1.6GHz Intel Atom processor and 2GB memory. Our modeling technique performs closely on two platforms. We will use the numbers from the desktop machines by default. In addition, we train and test these three models on a RAID-0 that consists of two SSD_I devices (namely Array_I).

The training data is generated by a synthetic I/O workload generator [21], which takes the workload characteristics as input parameters and generates a series of I/O requests that are sent to the storage devices. We create the I/O workloads under three types of access patterns, sequential, random, and stride. In our experiments, for the basic and extended models, the workloads use one access pattern at each time. The values for workload characteristics are selected as follows. The write ratio is from 0% to 25%, 50%, 75%, and 100%, where 0% means read only and 100% write only. The read and write size is selected from 1KB to 256KB (times of 4), queue depth from 1 to 64 (times of 4), read and write randomness in the value of 0%, 50%, and 100%, and stride size in the range of 1KB, 64KB, 128KB to 256KB. The workloads for the workload-specific models are collected in the same way as the basic and extended models, but they have different values for the workload characteristics. Specifically, the write ratio is read only or write only, queue depth of 1 means no outstanding I/Os, read/write size increases from 1KB to 256KB, read and write randomness from 0% to 100%, and stride size for write and read in the range of 1KB to 256KB.

Each I/O request is run for one minute by the workload generator. To remove the cache effects, we use direct I/Os, clear OS cache between the tests, and start each test at a random offset. Three performance metrics, latency in millisecond, bandwidth in MB/s, and throughput in IO/s are measured. For each I/O request, we collect a pair of the input and output, where each input consists of the values of workload characteristics and each output those of performance metrics. For each device, we run 12,000 one-minute workloads that in total take about 200 hours (about 8 days) to complete.

To construct a model, we feed the data points into an open source statistical software [22] and construct a linear regression tree. Given the training data, we apply a logarithmic transform to the performance values before the model building, which is a commonly used technique to squeeze in data values with a large spread. In our case, this will decrease the variance of performance values, especially for the latency that tends to have larger values with a small probability. Once a model is built, we evaluate its accuracy with two types of benchmarks, I/O requests from the synthetic generator, and four real-world I/O traces from OLTP (Online Transaction Processing) applications and a web search engine [23].

In this research, we measure the accuracy of the model with three statistical metrics:

- Mean Absolute Error (MAE) is defined as $|p - \hat{p}|$, and equivalent to the difference between the observed and predicated performance;
- Mean Relative Error (MRE) is defined as $|\frac{p-\hat{p}}{p}|$, and equivalent to the ratio between the absolute error and the observed performance;
- R^2 is defined as $1 - \frac{SSE}{SST}$ which determines how well the performance is likely to be predicted by the model, where error sum of squares $SSE = \sum(p_i - \hat{p}_i)^2$ and total sum of squares $SST = \sum(p_i - \bar{p})^2 = \sum(p_i^2) - (\sum(p_i))^2/n$.

TABLE II
PREDICTION ACCURACY OF BASIC MODELS

(a) Latency

Device	R^2	MAE(Mean)	MRE
HDD_S	0.808	28.94(94.61)	90%
SSD_I	0.627	6.90 (15.97)	63%
SSD_A	0.926	5.61 (36.31)	23%
SSD_S	0.693	14.21 (34.90)	55%

(b) Bandwidth

Device	R^2	MAE(Mean)	MRE
HDD_S	0.281	7.29(14.63)	110%
SSD_I	0.515	21.87(68.61)	40%
SSD_A	0.570	15.72(38.17)	86%
SSD_S	0.548	13.66(36.33)	63%

(c) Throughput

Device	R^2	MAE(Mean)	MRE
HDD_S	0.080	467(664)	50%
SSD_I	0.500	1,547(3,967)	53%
SSD_A	0.765	246(1,054)	19%
SSD_S	0.459	749(1,702)	48%

TABLE III
PREDICTION ACCURACY OF EXTENDED MODELS

(a) Latency

Device	R^2	MAE(Mean)	MRE
HDD_S	0.866	17.96(94.61)	26%
SSD_I	0.986	1.42 (15.97)	12%
SSD_A	0.976	3.16(36.31)	9%
SSD_S	0.911	6.22(34.90)	20%

(b) Bandwidth

Device	R^2	MAE(Mean)	MRE
HDD_S	0.768	3.67(14.63)	35%
SSD_I	0.981	3.91(68.61)	6%
SSD_A	0.882	6.29(38.17)	18%
SSD_S	0.917	5.21(36.33)	19%

(c) Throughput

Device	R^2	MAE(Mean)	MRE
HDD_S	0.870	152(664)	18%
SSD_I	0.970	254(3,967)	6%
SSD_A	0.971	74(1,054)	8%
SSD_S	0.951	212(1,702)	15%

The smaller the MAE and the MRE are, the better the model is. R^2 is a statistical measure of how well the regression line approximates the real data points and varies from 0 to 1. A R^2 value of 1 indicates that the regression line perfectly fits the observed data. In this paper, we evaluate the models by computing the means of the R^2, MAE, and MRE values.

B. Microbenchmarks

Basic Models: Table II lists the average values of R^2, MAE, and MRE of the latency, bandwidth, and throughput predictions for the basic model from all devices. This model does not work very well for the hard drive - the MRE for latency, bandwidth, and throughput is 90%, 110%, and 50%. In particular, the throughput prediction for the hard drive has low accuracy with the R^2 value of 0.08, while the bandwidth prediction with R^2 of 0.28. For the SSDs, one can see that all three metrics, latency, bandwidth and throughput, remain difficult to model, with R^2 values as low as 0.459 and MRE as high as 86%. The reason for high MRE is that four workload characteristics in the basic black-box model can not cover sufficient details to produce accurate predictions for both the hard drive and SSDs. For example, for SSD_I, the performance predictions have more than 50% MRE for throughput, even worse than HDD_S. Clearly, the basic model with a limited number of workload characteristics does not fit well for SSDs.

Extended Models: Using the workload characteristics defined in equation 4, we construct the extended models for all four devices. As shown in Table III, all the MRE values are greatly improved, i.e., 9% to 26% for latency, 6% to 35% for bandwidth, and 6% to 18% for throughput, with SSDs on the lower side. The improvements on R^2 are significant too - for SSDs, R^2 improves by more than 50% in some cases, and most of R^2 values are around 0.95. While the basic models have the mixed performance for SSDs, the extended models for all three SSDs have significantly better accuracy when compared to the hard drive model, that is, the SSD models have larger R^2 and smaller MRE values. In particular, the model for SSD_I has the best performance, 6% MRE for throughput, 6% for bandwidth, and 12% for latency.

For the extended models, latency remains most difficult to predict, consistent with the results from prior research [5] [4] [17]. The latency tends to increase dramatically when dealing with random writes, regardless of the type of the storage device. In our experiments, it helps that we apply the log transformation to our training data, which decreases the variance of data points and leads to the improved accuracy for the latency model.

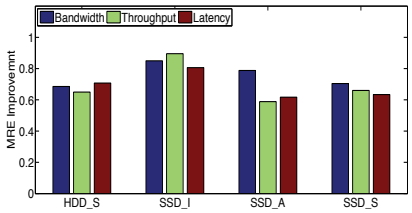


Fig. 9. MRE improvements between basic and extended models

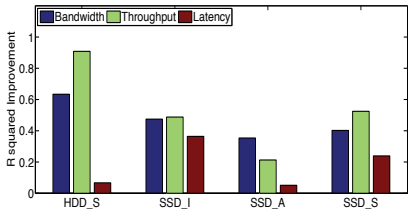


Fig. 10. R^2 improvements between basic and extended models

Fig. 9 and 10 show the improvements (in percentage) of MRE and R^2 of the extended models when compared with the basic models. For MRE, all the devices have close to or higher than 60% improvements for three performance models; for SSD_I there is 80% improvement. The most noticeable improvements for the latency models are the dramatic decreases

in MRE and large increases in R^2 values for SSD_I and SSD_S. While SSD_A has a small improvement on R^2, the MRE is greatly reduced. In summary, using a comprehensive set of workload characteristics, our black-box models are able to provide accurate predictions for various SSDs.

Workload-specific Models: In this test, we construct eight workload-specific models for HDD_S, SSD_I, and SSD_A: read-only (*rd_only*), write-only (*wr_only*), random-only (*rand_only*), sequential-only (*seq_only*), random-read (*rand_rd*), random-write (*rand_wr*), sequential-read (*seq_read*), and sequential-write (*seq_wr*). Table IV lists the R^2 and MRE values of three performance models HDD_S, SSD_I, and SSD_A. Clearly, the workloads with one access pattern will train our model better. For two SSDs, these models show similar patterns in the prediction accuracy for latency, bandwidth, and throughput. For example, the MRE values of two SSD models are between 0.4% and 7% for *rd_only* and *wr_only* model. For the other six workloads, most MRE values are less than 10%, which represents a good improvement compared to the extended models. The hard drive models are less accurate with much larger MRE.

TABLE IV
PREDICTION ACCURACY OF WORKLOAD-SPECIFIC MODELS

(a) Latency

	R^2			MRE		
Workloads	HDD_S	SSD_I	SSD_A	HDD_S	SSD_I	SSD_A
rd_only	0.953	0.999	0.999	18%	4%	1%
wr_only	0.942	0.996	0.990	17%	4%	7%
rand_only	0.910	0.975	0.991	29%	15%	5%
seq_only	0.933	0.954	0.989	16%	12%	6%
rand_rd	0.965	0.999	0.968	46%	2%	8%
rand_wr	0.984	0.993	0.993	9%	9%	5%
seq_read	0.964	0.999	0.999	8%	2%	6%
seq_wr	0.961	0.999	0.998	7%	7%	3%

(b) Bandwidth

	R^2			MRE		
Workloads	HDD_S	SSD_I	SSD_A	HDD_S	SSD_I	SSD_A
rd_only	0.923	0.994	1	15%	4%	0.4%
wr_only	0.911	0.978	0.990	15%	4%	4%
rand_only	0.879	0.980	0.960	15%	7%	12%
seq_only	0.915	0.996	0.977	16%	3%	11%
rand_rd	0.891	0.999	1	11%	1%	0.3%
rand_wr	0.963	0.946	0.993	9%	7%	4%
seq_read	0.920	0.999	1	9%	2%	0.4%
seq_wr	0.909	0.995	0.998	9%	2%	1%

(c) Throughput

	R^2			MRE		
Workloads	HDD_S	SSD_I	SSD_A	HDD_S	SSD_I	SSD_A
rd_only	0.970	0.979	1	14%	4%	0.4%
wr_only	0.975	0.996	0.816	16%	4%	7%
rand_only	0.049	0.987	0.999	11%	6%	4%
seq_only	0.987	0.996	0.998	13%	4%	3%
rand_rd	0	0.998	1	9%	3%	0.3%
rand_wr	0.946	0.989	0.255	9%	6%	5%
seq_read	0.990	0.996	1	10%	1%	0.4%
seq_wr	0.993	0.996	0.999	9%	2%	2%

C. SSD Array

In this case, we apply both the basic and extended black-box models on an SSD array (Array_I) on the Amdahl cluster, which provides the aggregate bandwidth and throughput by accessing two SSDs in parallel. For the basic model as shown in Table V, the disk array model achieves similar prediction accuracy to a single SSD_I on all three performance metrics, while for the extended model in Table VI, the model performance is somewhat worse where the extended models have the MRE value of 23% for latency, 25% for bandwidth, and 17% for throughput. Overall, the improvements for three performance metrics are about 30% compared to the basic models.

TABLE V
PREDICTION ACCURACY OF THE BASIC MODEL FOR SSD ARRAY

	R^2	**MAE(Mean)**	**MRE**
Latency	0.763	3.21(8.78)	59%
Bandwidth	0.446	41.87(98.82)	60%
Throughput	0.399	1,980(4,346)	52%

TABLE VI
PREDICTION ACCURACY OF THE EXTENDED MODEL FOR SSD ARRAY

	R^2	**MAE(Mean)**	**MRE**
Latency	0.939	1.43(8.15)	23%
Bandwidth	0.885	18.45(101.70)	25%
Throughput	0.860	934(4,875)	17%

D. Traces

In this section, we evaluate our extended black-box models using real world I/O workloads. We replay four block-level I/O traces [23] on the cluster with four devices (including HDD_S, SSD_I, SSD_S and Array_I). There are four traces, *Financial1* and *Financial2* from OLTP, and *WebSearch1* and *WebSearch2* from a web search engine. Eight workload characteristics and three performance metrics are measured at one-minute interval during trace replay. For each trace, our extended black-box models are trained with the first two hours of the trace. Then the models make the performance predictions for the third hour.

Table VII shows the MRE values of the models. Overall, our black-box models are able to achieve high accuracy for four traces. Two search engine traces produce much better accuracy than the financial traces for all devices - the MRE values of two web search engines traces are in the range of 1% to 5% on latency, bandwidth, and throughput predictions, while for two financial applications, the MRE values lie between a large range, from 7% to 38%. This can be attributed to the fact that two web search engine traces are read intensive with a very high read to write ratio, while two financial traces are write heavy. These prediction performances are consistent with the earlier results of the workload specific models - a read workload is easier to predict than write.

TABLE VII
PREDICTION PERFORMANCE OF I/O TRACES BY EXTENDED MODELS

(a) Latency (s)

Device	*Financial1*	*Financial2*	*WebSearch1*	*WebSearch2*
HDD_S	16%	12%	1%	3%
SSD_I	7%	19%	1%	1%
SSD_S	18%	15%	2%	1%
Array_I	19%	11%	1%	1%

(b) Bandwidth (MB/s)

Device	*Financial1*	*Financial2*	*WebSearch1*	*WebSearch2*
HDD_S	18%	30%	5%	5%
SSD_I	11%	38%	1%	1%
SSD_S	17%	14%	2%	1%
Array_I	26%	25%	2%	2%

(c) Throughput (IO/s)

Device	*Financial1*	*Financial2*	*WebSearch1*	*WebSearch2*
HDD_S	15%	26%	5%	5%
SSD_I	9%	37%	1%	1%
SSD_S	15%	14%	2%	1%
Array_I	25%	12%	1%	1%

V. RELATED WORK

Storage system performance modeling is of interest for many reasons, including architectural design, analysis and evaluation [24]–[26], power efficient storage [27], automatic resource control [28], and database management [29]. Prior research [30], [31] demonstrate large performance improvements from SSDs in database applications. Considering costs, however, [32] finds that replacing hard disks with SSDs in enterprise data centers is not economical based on workload trace analysis. We believe that accurate performance modeling of SSDs will facilitate the design, development, and evaluation of high performance flash-based storage systems, which is the motivation of this work.

Common performance modeling studies have targeted hard drives using analytical modeling [33]–[35], simulation [25], [36], benchmarking [37], [38], and black-box approach [4], [5], [39], [40]. Many analytical models have been developed for studying different disk characteristics, e.g., write caching [41], [42], cache hit and miss ratio [34], scheduling (FCFS and SSTF [43]), and LOOK and SCAN [44]). These analytic models are constructed in a white-box manner by developing an understanding of the internal organization of hard disks. Prior research studied performance modeling and simulation of SSDs [6], [45], [46], and all these SSD simulators rely on a deep understanding of the SSD internal architectures and control algorithms. However, as Gal and Toledo [47] point out, the internal design employed by SSDs are often trade secrets and regarded as closely-held intellectual property. In this paper, we extend our previous work [3] and conduct a comprehensive study of the SSD performance models. Prior research [4], [5] has constructed hard drive models in a black-box manner that is closely related to our work. Although the hard disk models are helpful when we study the SSD performance, we can not simply apply them directly to SSDs. Our work is different from the above work because we focus on workload characteristics that are most correlated to the SSD performance. To this end, we design an extended black-box model and investigate several workload-specific models.

VI. CONCLUSIONS

Flash-based solid-state drives will play an important role in future storage systems. An accurate performance model will provide important research tools for exploring the design spaces for such systems. In this paper, we study the black-box modeling for the analysis and evaluation of SSD performance. We construct the black-box models, using both synthetic workloads and real-world traces, on three SSDs, as well as an SSD RAID. We find that, while the black-box approach may produce less desirable performance predictions for hard disks, a black-box SSD model with a comprehensive set of workload characteristics can produce accurate predictions for latency, bandwidth, and throughput with small errors. In the future we plan to explore two directions: 1) evaluate our models against existing simulators, e.g., SSDSim [6] and FlashSim [46]; and 2) apply our black-box models, preferably in an autonomic manner, to help design and configure heterogeneous storage systems that consist of a large number of hard drives and flash-based devices.

VII. ACKNOWLEDGMENTS

We thank the anonymous reviewers for their helpful suggestions. This work was in part supported by the NSF grants OCI-0937875 and OCI-0937947.

REFERENCES

[1] M. CREEGER, "CTO Storage Roundtable," *Communications of the ACM*, vol. 51, no. 8, 2008.

[2] J. Gray and B. Fitzgerald, "Flash disk opportunity for server applications," *Queue*, vol. 6, no. 4, pp. 18–23, 2008.

[3] S. Li and H. Huang, "Black-Box Performance Modeling for Solid-State Drives," in *The 18th Annual Meeting of the IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), short paper*, 2010.

[4] M. Wang, K. Au, A. Ailamaki, A. Brockwell, C. Faloutsos, and G. Ganger, "Storage device performance prediction with CART models," in *Proceedings of the joint international conference on Measurement and modeling of computer systems*. ACM New York, NY, USA, 2004, pp. 412–413.

[5] L. Yin, S. Uttamchandani, and R. Katz, "An empirical exploration of black-box performance models for storage systems," in *14th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2006. MASCOTS 2006*, 2006, pp. 433–440.

[6] N. Agrawal, V. Prabhakaran, T. Wobber, J. Davis, M. Manasse, and R. Panigrahy, "Design tradeoffs for SSD performance," in *USENIX Annual Technical Conference*, 2008, pp. 57–70.

[7] F. Chen, D. Koufaty, and X. Zhang, "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*. ACM New York, NY, USA, 2009, pp. 181–192.

[8] Samsung, "K9XXG08UXM Flash Memory Specification," http://www.samsung.com/global/system/business/semiconductor/product/2007/6/11/NANDFlash/SLC_LargeBlock/8Gbit/K9F8G08U0M/ds_k9f8g08x0m_rev10.pdf, 2007.

[9] ——, "Spinpoint m7 hard disk specification," http://www.samsung.com/global/system/business/hdd/prdmodel/2009/1/13/728799m7_sheet_0.5.pdf, 2009.

[10] Intel, "Intel X-25M SSD Specification," http://download.intel.com/design/flash/nand/mainstream/mainstream-sata-ssd-datasheet.pdf, 2009.

[11] OCZ, "OCZ Apex SSD Specification," http://www.ocztechnology.com/products/memory/ocz_apex_series_sata_ii_2_5-ssd-eol, 2009.

[12] Samsung, "Samsung SSD Specification," http://www.samsung.com/global/system/business/semiconductor/product/2008/10/29/21970225_SATA_30Gbps_SLC.pdf, 2009.

[13] D. Ajwani, I. Malinger, U. Meyer, and S. Toledo, "Characterizing the performance of flash memory storage devices and its impact on algorithm design," MPI-I-2008-1-001, Tech. Rep., 2008.

[14] M. Moshayedi and P. Wilkison, "Enterprise ssds," *Queue*, vol. 6, no. 4, pp. 32–39, 2008.

[15] "Intel Corporation. Understanding the Flash Translation Layer (FTL) Specification," 1998.

[16] A. Ban, "Flash file system optimized for page-mode flash technologies," Aug. 10 1999, uS Patent 5,937,425.

[17] M. Mesnier, M. Wachs, R. Sambasivan, A. Zheng, and G. Ganger, "Modeling the relative fitness of storage," *ACM SIGMETRICS Performance Evaluation Review*, vol. 35, no. 1, p. 48, 2007.

[18] W. Loh, "Regression trees with unbiased variable selection and interaction detection," *Statistica Sinica*, vol. 12, no. 2, pp. 361–386, 2002.

[19] R. Koenker, *Quantile regression*. Cambridge Univ Pr, 2005.

[20] A. Szalay, G. Bell, H. Huang, A. Terzis, and A. White, "Low-Power Amdahl-Balanced Blades for Data Intensive Computing," in *USENIX HotPower*, 2009.

[21] Intel, "Intel-ISCSI open storage toolkit."

[22] W.-Y. Loh, "Guide regression tree version 7.9," 2009.

[23] UMass, "Umass trace repository," http://traces.cs.umass.edu/index.php/Storage/Storage, 2007.

[24] C. Wilkes and C. Ruemmler, "An introduction to disk drive modeling," *IEEE Computer*, vol. 27, no. 3, pp. 17–29, 1994.

[25] J. Bucy, J. Schindler, S. Schlosser, and G. Ganger, "The DiskSim simulation environment version 4.0 reference manual," Technical Report CMU-PDL-08-101, Carnegie Mellon University, Tech. Rep., 2008.

[26] G. Haring, C. Lindemann, and M. Reiser, *Performance evaluation: Origins and directions*. Springer, 2000.

[27] G. Da Costa, J. Gelas, Y. Georgiou, L. Lefevre, A. Orgerie, J. Pierson, O. Richard, and K. Sharma, "The GREEN-NET framework: Energy efficiency in large scale distributed systems," 2009.

[28] G. Alvarez, E. Borowsky, S. Go, T. Romer, R. Becker-Szendy, R. Golding, A. Merchant, M. Spasojevic, A. Veitch, and J. Wilkes, "Minerva: An automated resource provisioning tool for large-scale storage systems," *ACM Transactions on Computer Systems (TOCS)*, vol. 19, no. 4, pp. 483–518, 2001.

[29] O. Ozmen, K. Salem, M. Uysal, and M. Attar, "Storage Workload Estimation for Database Management Systems," in *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*. ACM, 2007, p. 388.

[30] S. Lee, B. Moon, C. Park, J. Kim, and S. Kim, "A case for flash memory SSD in enterprise database applications," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. ACM, 2008, pp. 1075–1086.

[31] S. Lee, B. Moon, and C. Park, "Advances in flash memory ssd technology for enterprise database applications," in *Proceedings of the 35th SIGMOD international conference on Management of data*. ACM, 2009, pp. 863–870.

[32] D. Narayanan, E. Thereska, A. Donnelly, S. Elnikety, and A. Rowstron, "Migrating server storage to SSDs: analysis of tradeoffs," in *Proceedings of the 4th ACM European conference on Computer systems*. ACM, 2009, pp. 145–158.

[33] C. Ruemmler and J. Wilkes, "An introduction to disk drive modeling," *Computer*, vol. 27, no. 3, pp. 17–28, 1994.

[34] E. Shriver, A. Merchant, and J. Wilkes, "An analytic behavior model for disk drives with readahead caches and request reordering," in *Proceedings of the 1998 ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*. ACM New York, NY, USA, 1998, pp. 182–191.

[35] M. Uysal, G. Alvarez, and A. Merchant, "A modular, analytical throughput model for modern disk arrays," in *9th International Symposium on Modeling, Analysis and Simulation on Computer and Telecommunications Systems (MASCOTS 2001)*.

[36] J. Wilkes, "The Pantheon storage-system simulator," *HP Laboratories technical report HPL–SSP–95–14 (rev. 1, May 1996), available from http://www. hpl. hp. com/SSP/papers*.

[37] A. Traeger, E. Zadok, N. Joukov, and C. Wright, "A nine year study of file system and storage benchmarking," *ACM Transactions on Storage (TOS)*, vol. 4, no. 2, p. 5, 2008.

[38] N. Agrawal, A. C. Arpaci-Dusseau, and R. H. Arpaci-Dusseau, "Towards realistic file-system benchmarks with codemri," *SIGMETRICS Perform. Eval. Rev.*, vol. 36, no. 2, pp. 52–57, 2008.

[39] E. Anderson, "Simple table-based modeling of storage devices. Technical note," HPL-SSP-2001-4, HP Labs, July 2001. http://www.hpl.hp.com/research/itc/scl/ssp/papers, Tech. Rep.

[40] T. Kelly, I. Cohen, M. Goldszmidt, and K. Keeton, "Inducing models of black-box storage arrays," Citeseer, Tech. Rep.

[41] S. Carson and S. Setia, "Analysis of the periodic update write policy for disk cache," *IEEE Transactions on Software Engineering*, vol. 18, no. 1, pp. 44–54, 1992.

[42] J. Solworth and C. Orji, "Write-only disk caches," *ACM SIGMOD Record*, vol. 19, no. 2, pp. 123–132, 1990.

[43] M. Hofri, "Disk scheduling: Fcfs vs.sstf revisited," *Commun. ACM*, vol. 23, no. 11, pp. 645–653, 1980.

[44] E. Coffman Jr and M. Hofri, "On the expected performance of scanning disks," *SIAM Journal on Computing*, vol. 11, p. 60, 1982.

[45] J. Lee, E. Byun, H. Park, J. Choi, D. Lee, and S. H. Noh, "Cps-sim: configurable and accurate clock precision solid state drive simulator," in *SAC '09: Proceedings of the 2009 ACM symposium on Applied Computing*. New York, NY, USA: ACM, 2009, pp. 318–325.

[46] Y. Kim, B. Tauras, and B. Gupta, A.and Urgaonkar, "FlashSim: A Simulator for NAND Flashbased Solid-State Drives," CSE 09-008, Penn State University, 2009., Tech. Rep.

[47] E. Gal and S. Toledo, "Algorithms and data structures for flash memories," *ACM Computing Surveys (CSUR)*, vol. 37, no. 2, p. 163, 2005.