Understanding Performance Anomalies of SSDs and Their Impact in Enterprise Application Environment

Jian Hu
University of Nebraska-Lincoln
Lincoln, Nebraska, USA
jhu@cse.unl.edu

Hong Jiang
University of Nebraska-Lincoln
Lincoln, Nebraska, USA
jiang@cse.unl.edu

Prakash Manden
VeloBit Inc.
Boxboro, Massachusetts, USA
prakash@velobit.com

ABSTRACT

SSD is known to have the *erase-before-write* and *out-of-place* update properties. When the number of invalidated pages is more than a given threshold, a process referred to as *garbage collection* (GC) is triggered to erase blocks after valid pages in these blocks are copied somewhere else. GC degrades both the performance and lifetime of SSD significantly because of the read-write-erase operation sequence.

In this paper, we conduct intensive experiments on a 120GB Intel 320 SATA SSD and a 320GB Fusion IO ioDrive PCI-E SSD to show and analyze the following important performance issues and anomalies.

(1) The commonly accepted knowledge that the performance drops sharply as more data is being written is not always true. This is because GC efficiency, a more important factor affecting SSD performance, has not been carefully considered. It is defined as the percentage of invalid pages of a GC erased block. It is possible to avoid the performance degradation by managing the addressable LBA range.

(2) Estimating the residual lifetime of an SSD is a very challenging problem because it involves several interdependent and mutually interacting factors such as FTL, GC, wear leveling, workload characteristics, etc. We develop an analytical model to estimate the residual lifetime of a given SSD.

(3) The high random-read performance is widely accepted as one of the advantages of SSD. We will show that this is not true if the GC efficiency is low.

Categories and Subject Descriptors

B.3.2 [**Design Styles**]: Mass storage

General Terms

Measurement, Performance

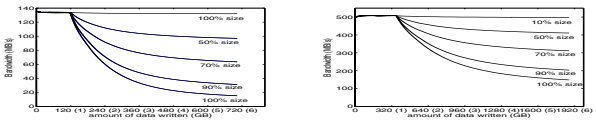
Keywords

SSD, anomaly

1. EVALUATIONS AND ANALYSIS

The unique physical characteristics and dynamic behaviors of SSD require different evaluation approaches than those for HDD. We follow the Solid Stage Storage Test Specification devised by SNIA to securely erase the SSD before all the tests, in light of the statefulness of SSD in that the evaluation results of one experiment run is affected by the completion status of the previous one.

SIGMETRICS'12, June 11–15, 2012, London, England, UK.
ACM 978-1-4503-1097-0/12/06.



(a) 120GB Intel 320 SSD (b) 320GB Fusion IO ioDrive

Figure 1: Performance impact of GC efficiency in SSDs

1.1 Performance impact of GC efficiency

It is widely accepted that the write bandwidth of SSD drops as more data is being written because of GC. However, we will show that GC efficiency is a more important factor than GC itself. By improving GC efficiency, it is possible to avoid or control the performance degradation.

Figure 1 shows the bandwidth as a function of the amount of data written to the SSDs. We customize the workloads by issuing 64KB random write requests with their LPNs confined to 0-10%, 0-50%, 0-70%, 0-90% and 0-100% respectively of the maximal LPN range of the SSDs. The amount of data written is set to be 6 times the full capacity of the SSD. The numbers in the parentheses of the x axis indicate the amount of data written divided by the full capacity of the SSD. By customizing the workloads this way, we can control the amount of invalidated pages generated when the SSD is full.

As shown in Figure 1, before the amount of data written reaches the capacity of the SSD, the bandwidth remains unchanged and is independent of the percentage of reserved LPN range. This is because before an SSD is full, it can process the write requests immediately by using its available free pages.

However, after the SSD is full, the bandwidth drops at very different rates depending on the GC efficiency. The fastest drop rate occurs when the LPNs of write requests are within 0-100% of the LPN range, where the bandwidth drops to as low as 12% and 30% of their peak bandwidth for the Intel 320 SSD and the Fusion IO ioDrive. The reason is that LPNs of the write requests scatter all over the address space of the SSD, making it less likely for the same LPN to be updated to generate enough invalidated pages. Therefore, GC has to copy many valid pages but only collects a limited number of invalid pages when victim blocks are erased, leading to a low GC efficiency.

At the other extreme, when the LPNs of write requests are within 0-10% of the LPN range, the write bandwidth of the SSDs hardly decreases as more data is being written. This is because the LPNs of the write requests concentrate within a small percentage of the LPN range. In this case, after all the free pages are consumed, most pages are likely to have been updated multiple times. Therefore, GC can find

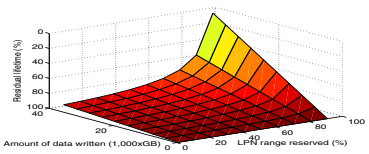


Figure 2: Analytical model of residual lifetime

victim blocks with a high percentage of invalidated pages, decreasing the number of valid pages copied and increasing the GC efficiency. The performances of the 0-50%, 0-70%, 0-90% cases fall in between the two extremes.

1.2 Residual lifetime estimation of SSD

Based on the addressable LBA range that can be controlled, we develop an analytical model to predict the residual lifetime as a function of the SSD capacity, the reserved LPN range, the amount of data written, and the maximal erase cycles of the flash memory, as shown in Equation 1.

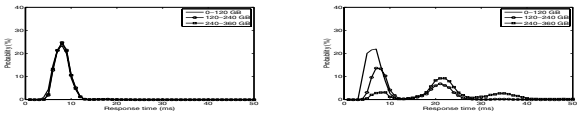
$$ResidualLifetime = 1 - \frac{A}{S \cdot [P + (1-P) \cdot M]} \quad (1)$$

In this equation, S denotes the SSD capacity, P_{size} the page size, $N_{page} = \frac{S}{P_{size}}$ the total number of physical pages in SSD, P the percentage of the reserved LPN range of the SSD (e.g., 0.9 if 0-90% of the full LPN range is used ($0 < P < 1$)), A the amount of data written to SSD in GB, and M maximal erase cycles.

We make the following justifiable assumptions. First, the SSD is assumed to be new to begin with and the only background operation is assumed to be GC because GC arguably dominates the background/internal operations in SSD. Second, the write requests are assumed to be random within the LPN range reserved for use, i.e., 0 to P of the full LPN range because of the high access randomness of enterprise-level applications. Therefore, the invalid pages are uniformly distributed among all blocks and the average number of invalid pages in each block is the same. We further assume that, for simplicity, when a victim block is erased, the valid pages in it are first copied somewhere else (e.g., RAM) and, after the block is erased, copied back to the same block that now contains no invalid pages.

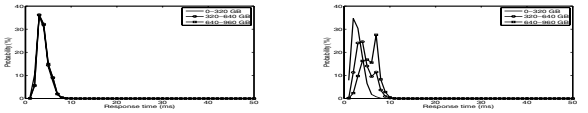
Write requests first propagate from the 0-to-P portion of the full PPN range of the SSD and, once this portion is full, then continue to propagate the remaining $1-P$ portion of the SSD PPN space. At this time, all the PPNs are used and the percentage of invalid pages in each block is $(1-P) \cdot 100$ because the latter requests are all updates. Then GC starts to collect these $(1-P) \cdot N_{page}$ invalid pages and the blocks containing these invalid pages are erased once as a result. After this point, every time $(1-P) \cdot N_{page}$ pages are written, the same number of invalid pages are generated, causing GC to be triggered to collect these invalid pages by erasing the corresponding blocks once. This process repeats until all the blocks are erased M times. Therefore, the maximal amount of data that can be written to the SSD is $N_{page} \cdot P_{size} \times P + N_{page} \cdot P_{size} \times (1-P) \times M = S \cdot P + S \cdot (1-P) \times M$. The residual lifetime of SSD is then calculated based on how much data is written to the SSD, as shown in Equation 1, in terms of the percentage of lifespan.

Figure 2 illustrates the predicted residual lifetime of a 100GB MLC SSD as a function of the reserved LPN range and the amount of data written based on the analytical model. We can see that when the reserved LPN range is small, say, less than 60%, the residual lifetime does not decrease significantly when more data is being written. However, when the reserved LPN range is more than 80%, the residual lifetime is reduced more quickly as more data is being written. Therefore, the reserved LPN range is a key factor in keeping the SSD running longer.



(a) 0-10% LPN is reserved (b) 0-100% LPN is reserved

Figure 3: Read latency of the Intel 320 SSD



(a) 0-10% LPN is reserved (b) 0-100% LPN is reserved

Figure 4: Read latency of the Fusion IO ioDrive

1.3 Unpredictability of random read

Most of the data sheets of SSDs only provide a single average value on response time of read requests. This value, however, turns out to depend highly on the GC efficiency, as revealed by our experimental evaluation.

In order to evaluate the read response time at different stages of GC, we mix the read and write requests and set the read and write requests ratio to 1:1. Both the read and write requests are random and their request size is 64KB. Further, in order to evaluate the read response time at different GC efficiencies, we customize the workloads by issuing requests with their LPNs confined to the ranges of 0-10% and 0-100% respectively of the maximal LPN range of the SSDs, as defined in Section 1.1. We write 3 times the capacity of the SSDs.

Figure 3 through Figure 4 show the response time of read requests as a function of the amount of data written to the SSDs and the percentage of the reserved LPN range of the requests. For all the SSDs, when the GC efficiency is high, as when the LPNs of requests are within 0-10% of the maximal LPN ranges of the SSDs, the response time of read requests falls into a normal distribution and is almost completely independent of the amount of data written, as shown in Figure 3(a) and Figure 4(a), which indicates that as long as GC efficiency is high, the response time of read requests is more predictable and can be kept steady no matter how much data is written to the SSDs. On the other hand, when the GC efficiency is low, as when the LPNs of requests are within 0-100% of the capacity of the SSDs, as shown in Figure Figure 3(b) and Figure 4(b), the response time of read requests increases sharply for all the three SSDs. For all the three SSDs, except for the period when the amount of data written is less than the capacity of the SSDs, the height of the curves decreases and the curves shift to the right, which indicates that the response time of read requests not only increases, but also fluctuates dramatically as the amount data written grows beyond the capacity of the SSDs.

2. ACKNOWLEDGEMENTS

This work is supported by US NSF under grants CNS-1116606, CNS-1016609, IIS-0916859 and CCF-0937993. This work is a joint work with VeloBit, Inc. The authors are also grateful to anonymous reviewers for their valuable input.