Power, Energy and Thermal Considerations in SSD-Based I/O Acceleration

Jie Zhang, Mustafa Shihab and Myoungsoo Jung
Department of Electrical Engineering, The University of Texas at Dallas
{jie.zhang6, mustafa.shihab, jung}@utdallas.edu

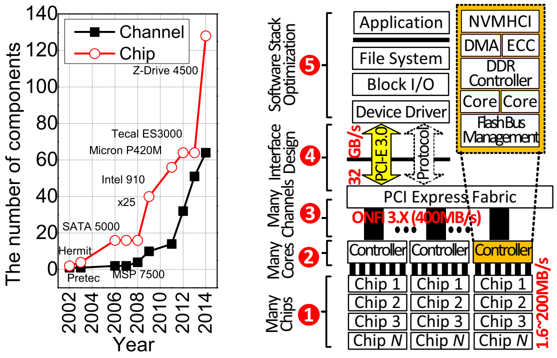
Abstract

Solid State Disks (SSDs) have risen to prominence as an I/O accelerator with low power consumption and high energy efficiency. In this paper, we question some common assumptions regarding SSDs' operating temperature, dynamic power, and energy consumption through extensive empirical analysis. We examine three different real high-end SSDs that respectively employ multiple channels, cores, and flash chips. Our evaluations reveal that dynamic power consumption of many-resource SSD is, on average, 5x and 4x worse than an enterprise-scale SSD and HDD, respectively. This work also addresses SSD overheating problem and power throttling issues, which result in significant performance degradation. Lastly, we offer an evidence that HW/SW optimization studies are needed to improve energy efficiency in future SSDs.

1 Introduction

Over the past few years, data storage systems have undergone significant architectural changes and hardware/software optimizations to accelerate their I/O services by taking advantage of the performance superiority of multiple flash memories. Modern Solid State Disks (SSDs) employ high speed bus such as PCI Express (PCIe) in place of conventional storage interfaces (e.g., SATA, SCSI), and equip more flash chips and internal bus channels. Figure 1(a) portrays the trend of how many internal resources are employed in them. From 2002, SSDs increased the number of channels and flash chips by 64 times, which in turn can improve their performance by 27x at most. In parallel, I/O interfaces exposed the channels and physical layouts of the underlying SSD to host-side kernel, and the corresponding software stack has been reconstructed [12, 14]. This in turn allows the systems to achieve higher performance with much better parallelism and utilization of SSD-related resources, being aware of system information. Thanks to these efforts, SSD-accelerated data-center applications reduce performance bottlenecks by 5x [3]. Similarly, high performance computing (HPC) applications improve their system performance by 2x∼10x [4], and big-data analytic can accelerate execution times by 7x∼10x through employing modern SSDs [7].

While all these prior works mainly focus on system performance improvements by better utilizing SSD-related resources, we believe that power, energy and thermal considerations demand high priority in numerous computing domains ranging from HPC to data-center to mobile computing systems. [11] and [2] anylized power consumption trend in SSDs for different configurations and combinations of workloads, and [13] proposed a new metric which can be used to dynamically control the degree of parallelism in SSDs. Even then, the resource utilization and internal parallelism affect dynamic power, energy and temperature, unfortunately these factors have received little attention in the study of many-resources SSDs. In this paper, we quantitatively analyze operating temperature, dynamic power, and the energy consumed by three different real high-end SSDs that respectively employ multiple channels, cores/controllers, and flash chips. Contrasting a common presumption that, SSDs can offer high performance with less power and energy consumption and at low operating temperature, our empirical evaluations reveal that dynamic power consumptions of a many-resource SSD is, on average, 5x and 4x worse than a conventional SSD and HDD, respectively. To the best of our knowledge, this is the first paper that quantitatively analyze dynamic power, energy and operating temperature exhibited by different types of modern SSDs. Our **contribution** can be summarized as follows:



(a) (b)

Figure 1: Resource employment trend over time for high performance SSD (a) and a research spectrum for SSD-based I/O acceleration (b).

- *Analyzing operating temperature on many-resource SSDs.* We measured operating temperature of each SSD we tested based on 44 different I/O access patterns with two different device conditions (pristine vs. aged) and offer extensive empirical analyses. We observed that many-resource SSDs generate 150%∼210% higher temperature than the conventional SSD. In cases where many internal flash chips are enabled to maximize the benefits of parallelism, the many-resource SSDs we tested, exhibit operating temperature of 160°F∼182°F, whereas the conventional SSD's heat-output is around 80°F. Considering the recommended datacenter temperature range (68°F∼77°F [6]), we believe that this operating temperature of the many-resource SSDs might not be acceptable for many market segments.
- *Revealing dynamic power and energy consumption behaviors.* For an accurate power evaluation, we built an in-house analyser, which can capture dynamic power values of diverse types of SSDs in a real-time fashion. We observed that, even though many-resource based SSD offer 7x better bandwidth and 6x shorter latency, they require 2x∼7x more dynamic power to extract their maximum

bandwidth, which can in turn make SSDs a power hungry device. Specifically, the conventional multi-channel SSD requires 4 watts at most, whereas the power consumption of the many-resource SSD, employing more flash chips, cores and channels, consumes 18 watts. Since many-resource SSDs exhibit shorter latency than the conventional SSD by 75% on average, it is expected to be an energy efficient device, which is another common presumption. We however observed that the energy requirement of the many-resource SSD is 282% and 109% of the conventional SSD, for reads and writes, respectively.

• *Addressing SSD overheating and power throttling issues.* We observed that high temperature values exhibited by the many-resource SSD can also have significant impact on its performance degradation. When the heat-output of the many-resource SSD reaches 180°F, it begins to throttle power consumption by reducing the number of active internal resources in an attempt to cool down the device. However, this power adjustment in turn degrades the perform around 16%, which is as significant as the write cliff [9] that dramatically drops the device performance to internally perform garbage collections.

2 Background and Related Work

2.1 Hardware Architecture

Many chips. A flash memory chip (die), by itself, offers data rate of only 1.6 MB/sec to 200MB/sec (even under the assumption that all the data movements can be completely overlapped with memory operations), which is lagging far behind the maximum bandwidth of modern PCIe interfaces. One promising way to bridge this performance disparity between the flash die and high-speed interface is to take advantage of parallelism among multiple flash chips as shown in ❶ of Figure 1(b). State-of-the-art SSDs in practice have 64 ∼ 128 chips, so that the aggregate performance of all these flash chips can follow up the high-speed interface bandwidth. Because of this, prior studies propose diverse hardware approaches [5, 4] and queue optimizations [10] to take advantage of chip-level parallelism.

Many channels. Many flash chips can be connected to a flash channel, which is an internal data path directly connected to PCIe interface as shown in ❸ of Figure 1(b). Since each channel can be enabled with few constraints, exploiting channel-level parallelism is another key to improving modern SSD performance. As shown in Figure 1(a), the number of channels has increased by 64 times over the past decade, and to take advantage of these many channel resources, diverse bus/channel topologies and queuing algorithms have been proposed [5, 4].

Many cores/controllers. In addition to these many channels and flash chips, modern SSDs have various features needed to operate in parallel - which a single computational unit may be unable to cope with. For example, a phase tag technique has been introduced [1], which can compose multiple queues (64K); each of them can simultaneously submit requests and collect completion data from the underlying storage mediums. For taking full benefit of these queues, the number of parallel computations has increased too. As internal DRAM size increases, separate DMA/DRAM controllers are required. Further, the flash chips are needed to employ powerful error correction code as flash feature size shrunk. All these require more computation power, which can introduce multiple cores and controllers into an SSD (shown in ❷ of Figure 1(b)).

	SP-SSD	**MC-SSD**	**MR-SSD**
Feature	Multi-channel	Many-core	Many-resource
Interface	SATA 6Gpbs	PCIe 2.0 x4	PCIe 2.0 x4
Cores	3	16	16
# of channels	8	8	32
# of flash chips	64	64	128
DRAM size	256MB	2GB	2.25GB
Storage cap.	512GB	400GB	512GB

Table 1: Important characteristics of the tested SSDs.

2.2 Software Architecture

Although the many-resource SSDs approach may improve performance, their resource utilization and degree of parallelism are limited by I/O access patterns and sizes determined by host kernel modules [12, 10]. Therefore, there exist diverse software and I/O stack research performed in an attempt to fully utilize the underlying resources.

I/O stack reconstruction. [12] minimizes enforcement necessities imposed by the host-side software stack, by migrating them to hardware and through better utilization of internal resources. In contrast [8] migrates garbage collector and channel manager from the underlying SSD to host-side I/O stack, so that systems can make better decisions to achieve higher performance and parallelism. All these studies target to optimize ❺ of Figure 1(b), by fully utilizing the underlying hardware architecture, which can lead to high performance and parallelism.

Scheduler. [5] employs an internal scoreboard and queuing algorithm that keeps track of all in-flight requests and improves performance with a high degree of parallelism. Similarly, [10] proposes a scheduling mechanism to serve I/O requests in an out-of-order fashion, which can lead to high parallelism with less dependency on the host-side software. These approaches require more computation units and internal DRAM buffers, which has some impact on research - as shown in ❷ of Figure 1(b).

Hardware and software codesign. [12] proposes codesign approach to fully exploit the raw SSD resource performance. [14] proposes an way to expose some of the physical addresses and internal information to host-side kernel, so that the host-side software can be aware of available internal resources. These codesign and interface optimizations (❸ and ❹ of Figure 1(b)) also contribute to better utilization of the underlying hardware in SSD based I/O acceleration.

Note that, all these hardware and software research exclusively focus on parallelism and resource utilization for performance enhancements, which can adversely affect dynamic power, energy and temperature, *but this factor is getting no attention.*

3 Evaluation Setup

3.1 Testbeds and Toolkits

SSD testbeds. We evaluate three different real SSD testbeds; i) SP-SSD, ii) MC-SSD, and iii) MR-SSD. SP-SSD (Simple-processor SSD) employs three cores, eight independent channels, and each of them have eight flash chips. The configuration of MC-SSD (Many-core SSD) is similar to SP-SSD, but it has more cores (up to 16) and bigger internal DRAM buffer (2GB). In addition to 16 cores, MR-SSD employs 128 flash chips and 32 channels.

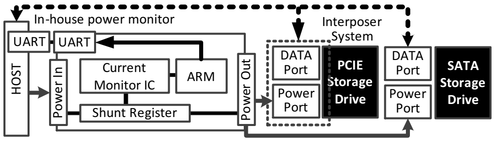


Figure 2: In-house power monitor.

While SP-SSD uses 6Gbps SATA, MC-SSD and MR-SSD both use high-speed PCIe bus. All important characteristics of SSDs we tested are shown in Table 1.

System environment. Our experimental system is equipped with an Intel Quad Core i7 Sandy Bridge 2600 3.4 GHz processor and 8GB DDR3-1333Mhz memory. We executed all our tests with our measurement tools and power analyzer (which we will explain shortly). We then stored logs and output results into separate block devices in a full asynchronous fashion; neither a system partition nor a file system is created on our SSD testbeds, which allows each SSD testbed to be completely separated from the evaluation scenarios and tools.

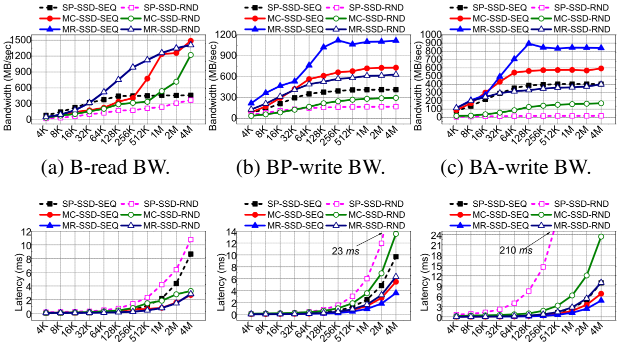
Dynamic power measurement. In order to capture dynamic power values under diverse workloads in a real-time fashion, we developed an in-house power evaluation platform. The overall architecture of our power analyzer is shown in Figure 2. The power analyzer employs an ARM SAM D20 core, I/O power ports, a current sense monitor and a microcontroller. The host evaluation system and two different SSDs are connected to the input power and output power ports of our power analyzer on Northbridge and Southbidge, respectively. Through a shunt register (0.1∼0.01 ohm), the current monitor controller senses current values used by the target SSDs, and converts analogue input values to digital values. The converted voltage values are transformed to power by the ARM processor.

Temperature measurement. We also developed an application, which can capture very specific raw-level information such as the number of read/write error, block-erase count, error correct count and power cycles through SMART command. One concern we had in using this tool is that it might introduce a certain overhead for evaluation workload executions, if we measure all the data too frequently. To address this concern, we measured the operating temperature with a 1 minute interval period.

3.2 Workloads and Preliminary Evaluation

Device condition and workload. We generate full random access and sequential access read and write, with varying I/O request sizes ranging from 4KB to 4MB. It is established that SSDs can exhibit different characteristics as I/O service progressed, so we made two different device statuses: pristine and aged. While evaluations on pristine SSDs has factory default status, we made the aged SSD fragmented by writing 4KB∼128KB with fully aligned data over entire logical block address range provided by the underlying device. This aged SSD can mimic the case of an actually aged SSD. To make description concise, we added *SEQ* and *RND* at the end of each device codename for sequential access tests and random access tests, respectively. In our evaluation, *B-read*, *BP-write* and *BA-write* indicate read, write on the pristine device, and write on the aged device benchmarks, respectively.

Overall performance. To better understand the relationship among performance, temperature, power and energy, we provide preliminary performance evaluation with the same test condition used for Section 4; the evaluation results in terms of bandwidth and latency are shown in Figure 3. On average, MC-SSD and MR-SSD can respectively offer around 2x and 3x better bandwidth, and around 1.6x and 2.75x shorter latency than the SP-SSD. While the performance on sequential access is better than random access in most cases, MR-SSD-RND exhibits better read performance compared to MR-SSD-SEQ in terms of both bandwidth and latency. We will shortly discuss these performance issues, considering other factors like energy, temperature, and power consumption behaviors.



(a) B-read BW. (b) BP-write BW. (c) BA-write BW.

(d) B-read latency. (e) BP-write latency. (f) BA-write latency.

Figure 3: Overall performance.

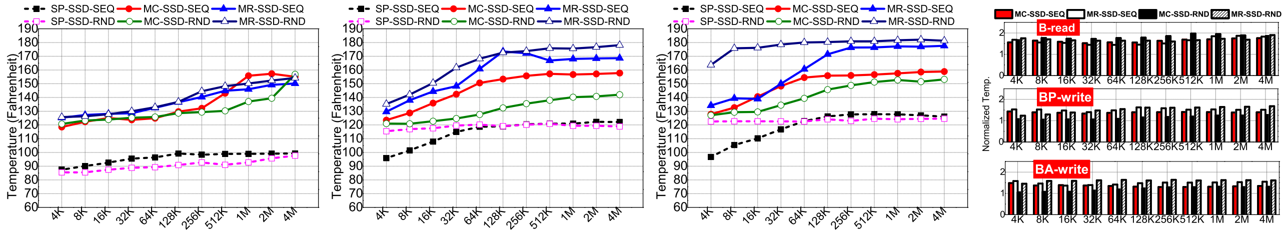
4 Results

4.1 Operating Temperature Analysis

Figures 4(a), 4(b) and 4(c) show diverse operating temperature values for *B-read*, *BP-write* and *BA-write*, respectively. To better understand the temperature trend, we also normalized each value to that of SP-SSD, and they are shown in Figure 4(d). One can see from the figures that, SP-SSD is in a temperature range between 85°F and 128°F, which is similar to datacenter recommended thermal consideration [6]. In contrast, the average heat-output of all many-resource SSDs we tested is in a range between 119°F and 182°F, which exceeds the recommended temperature by around 70%, as shown in Figure 4a.

Many-core. While SP-SSD has little or no impact on its operating temperature for varying request sizes, the many-core and many-resource SSDs increase temperature by upto 33% as the request size increases. We believe that, this is because many flash chips are activated by the multiple channels and cores, the same feature that enables them to achieve higher performance and parallelism. Even though MC-SSD has the same number of channels and chips as SP-SSD, it generates 42% higher temperatures. Further, in cases where MC-SSD has read bandwidth similar to MR-SSD with more flash chips and channels (1MB∼4MB in Figure 4a), the heat-outputs reaches around 158°F, which is 6% higher than MR-SSD.

Many-resource. In most cases, the heat-output of MR-SSD is higher than any other devices we tested. MR-SSD generates 71%, 49% and 54% higher temperature than SP-SSD for *B-read*, *BP-write* and *BA-write*, respectively. Even compared to MC-SSD, it exhibits 13% higher thermal values, on average. Unlike SP-SSD and MC-SSD, the temperature values of MR-SSD-RND are higher than that of MR-SSD-SEQ by 2%, 4% and 12% *B-read*, *BP-write* and *BA-write*, respectively. We believe that, this is one of the reasons why MR-SSD's performance on random



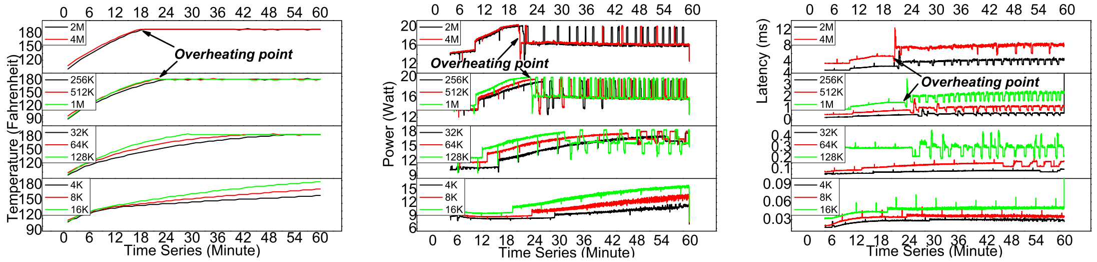
(a) B-read.

(b) BP-write.

(c) BA-write.

(d) Normalized to SP-SSD.

Figure 4: Operating temperature.



(a) Temperature.

(b) Dynamic power.

(c) Latency.

Figure 5: Time series analysis of MR-SSD for the autonomic power adjustment regarding the overheating problems.

accesses are much better then sequential accesses under *B-read*, in fact better than all other devices under sequential accesses. Interestingly, the temperature of MR-SSD-RND on the aged SSD (Figure 4c) presents the highest heat-output in most cases and reach 180°F for incoming write requests greater than 64KB. We believe that, this is because the aged-device can introduce extra internal I/O operations, such as garbage collection and wear-levelling, more than a pristine device.

4.2 Overheating Issues

In contrast to the fact that SP-SSD has no overheating problem, we observed that MC-SSD exhibits a performance degradation of more than 10x - when the temperature of MC-SSD exceeds 170°F. We conjecture that this unreasonable performance-drop is caused by a malfunction of MC-SSD internal hardware due to high temperature. Even though we are not able to demonstrate time series analyses regarding MC-SSD due to space limit, we observed that MC-SSD has no autonomic power adjustment to protect itself against the overheating problem. Therefore, we will focus on explaining the overheating issues with our observations of MR-SSD in this work.

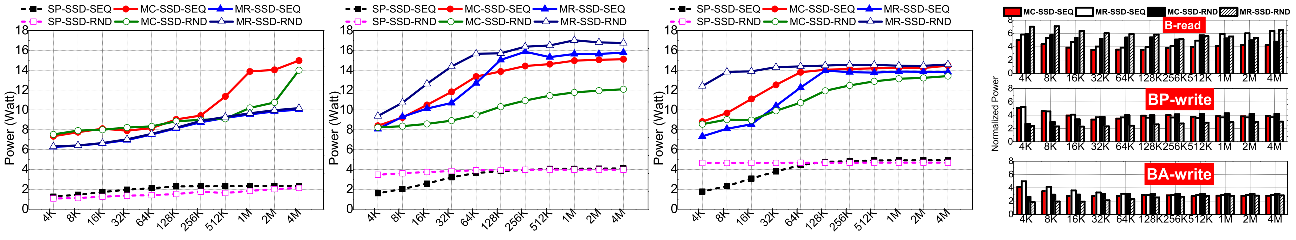
We observed MR-SSD has a device-level protection mechanism, which dynamically adjusts its power level based on temperature. Specifically, Figure 5a shows the trend of MR-SSD's operating temperature over time for random writes on a pristine device. In addition, Figures 5b and 5c show the dynamic power trend and latency trend of MR-SSD associated to Figure 5a, respectively. For each figure, we marked overheating points where MR-SSD reaches the threshold temperature that begin to throttle the performance and consume much less power so that MR-SSD can cool down on their own – in our empirical evaluation, we observed that the threshold is 188°F. As discussed in the previous section, MR-SSD enables many internal resources as the request size increases. One can observe from these figures that MR-SSD automatically sets power 15%, 15% and 16% lower than a normal case of 512KB, 2MB, and 4MB, respectively, when it reaches the overheating point (around 18∼24 minutes). This dynamic power adjustment can lead to performance drops by 17%, 12% and 18% for 512KB, 2MB, and 4MB request sizes, respectively.

4.3 Dynamic Power Analysis

The SP-SSD lives up to the common presumption of SSDs being power efficient devices by consuming only 2W and 4W for read and write, respectively. However, we observed that the high-performance SSDs consume 4.5x and 3.5x more power than SP-SSD. Specifically, as shown in Figure 6, MC-SSD and MR-SSD, on average, consume 11W and 12W dynamic power - which might not be acceptable in many low power applications.

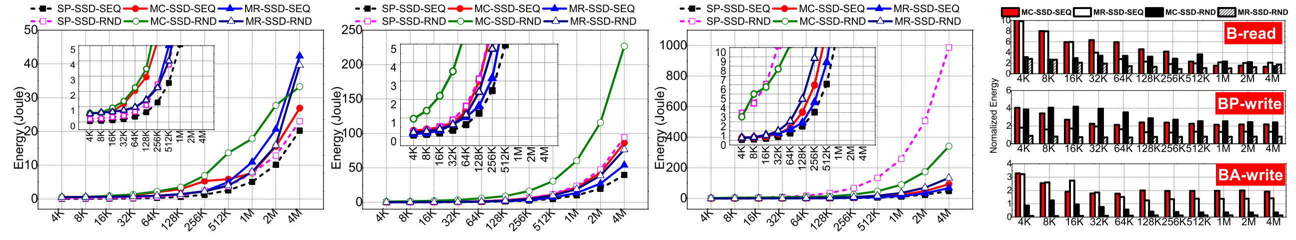
Many-core. While SP-SSD exhibits similar power consumption behaviors irrespective of which access pattern has been executed, MC-SSD consumes more power for sequential accesses than for random accesses. Specifically, it requires 6%, 25% and 14% more power for *B-read*, *BP-write* and *BA-write* workloads, respectively. When the power values exhibited by MC-SSD reach around 15 watts, we observe two accompanying phenomenons: i) the temperature settles down at around 160°F, and ii) the performance saturates and there is no further benefit in taking advantage of SSD's internal parallelism. When it consumes more than 16 watts, the temperature goes over the overheating point, which can in turn introduce the significant performance degradation. We believe this is one of the reasons why modern SSDs need to take more attention to the power management.

Many-resource. As mentioned in the previous analysis, MR-SSD exhibits better bandwidth and shorter latency on random access rather than sequential access. The dynamic power consumption behavior is one of the good evidences that support the superiority of random access performance for the many-resource SSD. Specifically, MR-SSD is expected to enable more flash chips and better utilize the internal DRAM, even under random access pattern. With



(a) B-read. (b) BP-write. (c) BA-write. (d) Normalized to SP-SSD.

Figure 6: Dynamic power analysis.



(a) B-read. (b) BP-write. (c) BA-write. (d) Normalized to SP-SSD.

Figure 7: Energy analysis.

4x more channels and 2x more flash chips than MC-SSD, it should encounter less resource contention in taking the benefits of SSD's internal parallelism. As shown in Figure 6, the MR-SSD require more dynamic power to feed its internal components under random access patterns than that of sequential access patterns - it requires 15% and 28% more power for *BP-write* and *BA-write* workloads, respectively. Interestingly, while all other SSD devices require more power under *BA-write* because of the extra I/O operations imposed by the flash firmware (such as garbage collection), the aged MR-SSD consumes 1% less power under random access pattern, compared to what it consumed when in pristine condition (shown in Figure 6c). We can find the reason behind this by correlating the overheating problem. The temperature of the aged MR-SSD hit the overheat point (168°F) for writes with random access patterns, which make the autonomic adjustment module diminishes its dynamic power.

4.4 Dynamic Energy Analysis

Figure 7 illustrates the energy analysis for each benchmark. The top left corner of the figures shows the energy values in cases where the request sizes less than 512 KB.

Overall energy efficiency. We believe that MC-SSD and MR-SSD are unfortunately not an energy efficient solutions. Specifically, MC-SSD and MR-SSD consume,on average, 120%, 110% and 15% higher energy values than SP-SSD for *B-read*, *BP-write* and *BA-write* workloads, respectively. One of the reasons why they show poor energy efficiency is that, even though many-core and many-resource SSDs are successful for improving performance by enabling their many internal resources, they require much more power to feed all their internal resources.

Energy efficiency on aged devices. Even though many-core and many-resource SSDs are not energy efficient in general, they can significantly reduce energy values in the aged device for random accesses by utilizing abundant internal resources. For example, while MC-SSD-SEQ and MR-SSD-SEQ shows worse energy efficiency than SP-SSD-SEQ by 63% and 85% respectively, they consume, on average, only 16% and 18% of the energy values exhibited by SP-SSD from random accesses.

Considering that the dynamic power consumption of the MC-SSD-SEQ and MR-SSD-SEQ are so much higher than SP-SSD-SEQ, we speculate that many-core and many-resource SSDs are well optimized to hide overheads of the extra I/O operations by activating many internal resources for such computation units (for better scheduling), flash chips and channels (for higher parallelism).

5 Conclusions

In this paper, our empirical analysis reveal that dynamic power consumption of many-resource SSDs are respectively 5x and 4x worse than conventional SSD and HDD. Many-resource SSDs generate 58% higher operating temperature, which can introduce SSD overheating problem and power throttling issues. Based on our analysis, HW/SW optimization studies are required to improve energy efficiency of modern SSDs in many user scenarios.

References

[1] Nvm express specification. 2013.

[2] Matias Bjorling et al. uflip: Understanding the energy consumption of flash devices. In *IEEE Data Engineering Bulletin*, 2010.

[3] Christian Black and Darrin Chen. Data center ssd - accelerating data center workloads. In *Intel*, 2012.

[4] M. Caulfield, A et al. Gordon: An improved architecture for data-intensive applications. In *MICRO*, 2010.

[5] M. Caulfield, A et al. Moneta: A high-performance storage array architecture for next-generation, non-volatile memories. In *MICRO*, 2010.

[6] Ken Darrow and Bruce Hedman. Opportunities for combined heat and power in data centers. In *Oak Ridge National Laboratory*, 2009.

[7] Arup De et al. Minerva: Accelerating data analysis in next-generation ssds. In *FCCM*, 2013.

[8] O. Jian et al. Sdf: Software-defined flash for web-scale internet storage systems. In *ASPLOS* , 2014.

[9] M. Jung and M. Kandemir. Revisiting widely held ssd expectations and rethinking system-level implications. In *SIGMETRICS*, 2013.

[10] M. Jung and M Kandemir. Sprinkler: Maximizing resource utilization in many-chip solid state disks. In *HPCA*, 2014.

[11] Euiseong Seo et al. Empirical analysis on energy efficiency of flash-based ssds. In *HotPower*, 2008.

[12] S. Swanson and A.M. Caulfield. Refactor, reduce, recycle: Restructuring the i/o stack for the future of storage. *Computer*, 2013.

[13] Balgeun Won et al. Ssd characterization: From energy consumptions perspectives. In *HotStorage*, 2011.

[14] Yiying Zhang et al. De-indirection for flash-based ssds with nameless writes. In *FAST*, 2012.