

# Ultra-Low Latency SSDs’ Impact on Overall Energy Efficiency

Bryan Harris and Nihat Altiparmak

*Computer Science and Engineering Department, University of Louisville*  
*{bryan.harris.1,nihat.altiparmak}@louisville.edu*

## Abstract

Recent technological advancements have enabled a generation of Ultra-Low Latency (ULL) SSDs that blurs the performance gap between primary and secondary storage devices. However, their power consumption characteristics are largely unknown. In addition, ULL performance in a block device is expected to put extra pressure on operating system components, significantly affecting energy efficiency of the entire system. In this work, we empirically study overall energy efficiency using a real ULL storage device, *Optane SSD*, a power meter, and a wide range of IO workload behaviors. We present a comparative analysis by laying out several critical observations related to idle vs. active behavior, read vs. write behavior, energy proportionality, impact on system software, as well as impact on overall energy efficiency. To the best of our knowledge, this is the first published study of a ULL SSD’s impact on the system’s overall power consumption, which can hopefully lead to future energy-efficient designs.

## 1 Introduction

Various innovations have taken place in the storage subsystem within the past few years, including wide adoption of the NVMe interface for new generation storage devices [1], the corresponding multi-queue request submission/completion capability (*blk-mq*) implemented in the Linux block IO layer [2,3], and new IO scheduling algorithms specifically designed for *blk-mq* [4–6]. We have also seen the emergence of new storage technologies such as Intel’s Optane SSDs based on their 3D XPoint technology [7], Samsung’s Z-SSD with their SLC based 3D NAND technology [8], and Toshiba’s XL-Flash design using a similar technology as Z-SSDs [9,10]. All these innovations enable a new generation of Ultra-Low Latency (ULL) SSDs that are broadly defined as providing sub-10 $\mu$ s data access latency [11]. Table 1 compares ULL SSD performance with existing storage technologies and shows how ULL SSDs help close the performance gap between primary and secondary storage devices.

Depending on the NVM technology that they use, ULL SSDs may come with completely different internal char-

Device	Latency (4 KB random read)	Performance Gap (relative to DRAM)
HDD	$\sim 10$ ms	100,000 $\times$
Traditional SSD	$\sim 100$ $\mu$ s	1000 $\times$
ULL SSD	$\sim 10$ $\mu$ s	100 $\times$
DRAM	$\sim 100$ ns	1 $\times$

Table 1: Performance gap in storage hierarchy

acteristics than HDDs and traditional SSDs. For example, 3D XPoint developers state that traditional techniques used in HDDs and Flash SSDs, such as IO merging in IO schedulers, prefetching, use of log structures instead of in-place updates, and using an intermediate buffer cache are not only unnecessary but get in the way of Optane SSDs [12]. Recently, Wu et al. released their work towards an unwritten contract for Optane SSDs and further outlined performance characteristics of the 3D XPoint technology, including the lack of garbage collection, equal treatment of reads and writes, and comparable performance of sequential and random requests [13]. However, the power consumption characteristics of Optane SSDs and their 3D XPoint technology are largely unknown. Furthermore, no previous work has investigated the impact of ULL IO performance on overall energy efficiency, especially considering the pressure it puts on system software.

In this paper, first we characterize the power consumption behavior of the Optane SSD, including its idle vs. active power consumption, read vs. write power consumption, and energy proportionality by comparing it to traditional storage technologies such as spinning magnetic disks and flash SSDs. Second, we analyze the impact of ULL disk IO on overall energy efficiency of the entire system. We believe that energy efficiency is a crucial but commonly neglected issue in system design, and hope that our observations can aid both industry and academia in developing future energy-efficient systems.

## 2 Related Work

Various recent work investigated the performance characteristics of ULL SSDs [11–15]; however, except for limited analysis of individual devices [16,17], their power consumption characteristics are largely unknown and their impact on

the overall energy efficiency of the entire system has unfortunately not yet been thoroughly studied. Existing energy efficiency research in the area of storage systems has mainly focused on HDDs and traditional flash-based SSDs, such as techniques to reduce HDD R/W head movement [18] or to power-off/spin-down a subset of HDDs organized in a RAID structure [19]. Further research focused on improving HDD energy proportionality [20], reducing the power consumption of data center storage systems composed of HDDs and flash SSDs [21], characterizing the power consumption of HDD based enterprise-scale backup storage systems [22], and reducing HDD spin-ups in archival storage systems using data grouping techniques [23]. More recently, researchers analyzed the design tradeoffs of flash SSDs for improved energy efficiency [24] and the impact of SSD RAIDs on server energy consumption [25].

New generation ULL SSDs are generally equipped with alternative NVM designs, such as phase-change memory used in 3D XPoint technology [12], and a different controller design based on the ULL performance goals and the physical characteristics of the underlying NVM [12, 13]. In addition, ULL storage performance is expected to impact the overall energy efficiency of the system due to increased pressure on system software, especially considering the new *blk-mq* layer, IO schedulers, and NVMe driver, legacy performance optimizations performed with older storage characteristics in mind, and the current interrupt based IO handling mechanism. Existing assumptions do not apply to ULL SSDs and a new, systematical study is necessary to understand their power consumption characteristics and impact on overall energy efficiency considering recent changes in the Linux kernel.

### 3 Energy Characterization and Efficiency

#### 3.1 Methodology

We evaluate energy efficiency using four storage devices in an enterprise level system, which use three different storage technologies as summarized in Table 2. The first is a spinning magnetic disk, a 3.5" WD Black 7200 rpm HDD on a SATA 3.1 bus. The second and third are NAND flash based traditional SSDs by Samsung. The 850 EVO is connected to a SATA bus and the 960 EVO is an M.2 NVMe using a PCIe 3 adapter. The fourth is a representative of a ULL device, an Intel Optane 900P, using their proprietary phase change memory based 3D XPoint technology. These four devices are at a similar price point and include a range of technologies, interfaces, performance, and energy efficiency, all of which represent three generations of storage mechanics and assumptions:

**Spinning magnetic storage**, such as an HDD, is well understood with decades of research and software optimizations based on its internal mechanics. HDDs are fundamentally sequential devices; performance suffers from random access, but they can modify data in place. Motors keep spinning the platter while the device is idle, yielding high idle

Technology	Interface	Model	Capacity
Magnetic	SATA 3.1	WD Black 7200 rpm	4 TB
Flash	SATA 3.1	Samsung 850 EVO	1 TB
Flash	PCIe 3	Samsung 960 EVO	500 GB
3D XPoint	PCIe 3	Intel Optane 900P	280 GB

Table 2: Summary of tested storage devices

power consumption. Power saving states that spin down the platter are known to be costly both in latency and wear.

**Traditional solid-state storage**, is fully electronic with no moving parts based on NAND flash. These devices must “erase before write” and an erase unit is typically many times larger than a write unit, eliminating in-place updates. Floating gate transistors wear out after an excessive number of writes, thus the controller must dynamically manage allocations, wear leveling, and garbage collection.

**New generation ULL storage**, based on either a new NVM technology or an optimized flash-based design, is characterized by low data access latency (sub-10  $\mu$ s). The representative we measure here is Intel’s Optane 900P SSD, block addressable storage based on Intel and Micron’s proprietary 3D XPoint technology, a form of phase change memory. Unlike flash-based SSDs, the Optane SSD performs in-place updates, there is no internal garbage collection, and it is a true random access storage device [13].

#### 3.2 Experimental Setup

We attached our *test* storage devices to a Dell PowerEdge R230, which has a single socket Intel Xeon E3-1230 quad-core CPU (3.4 GHz) and 64 GB RAM. We installed CentOS 8.1 and upgraded the Linux kernel to version 5.5.0 (current stable). The system also has a Dell-certified SSD used as the OS disk, which was present in all experiments but not used for testing. During experiments, only one *test* storage device was physically installed at a time.

Power consumption was measured using an Onset HOBO UX120-018 Data Logger [26], an in-line power meter connected to the power supply of our test system, which does not include a redundant power supply. This meter records voltage, current, active power, active energy, apparent power, and power factor at a polling rate of once a second and stores it internally until it is read out through a USB interface. These measurements are later synchronized to our workloads using time stamps recorded by our scripts. Both system clocks are synchronized to the same time server.

#### 3.3 Workloads

For each storage device, we measured the system’s overall power consumption when the device is idle and active under light to heavy IO workloads for a range of request sizes. Workloads were generated using *fio*, the Flexible IO tester [27], version 3.18. In order to capture the maximal range of effects on the system, we wish to push the storage device to its limits. We submit asynchronous direct requests to the storage device, that bypass the page cache, and tested a range of specific IO behaviors similar to the workloads generated

by Wu et al. [13]. The request sizes tested were 1 KB and powers of 2 from 4 KB through 256 KB. Modern storage, like modern computers and applications, are highly parallel. We tested both a single thread and four threads (since we used a quad-core CPU), and with a range of requests outstanding to the application, referred to as IO depth or queue depth. These outstanding requests may be queued in the block layer, device driver, or in the device itself. From the application’s perspective they are submitted asynchronously, but not yet completed. We used *ext4* as the file system and *mq-deadline* as the *blk-mq* scheduler since they are current default choices in Linux. In order to record a sufficient number of power measurements from the in-line meter, we ran each workload three times for a fixed two minutes and averaged results.

### 3.4 Idle vs. Active Power Consumption

**Observation 1.** *Idle power consumption of Optane SSD is higher than that of traditional SSDs but lower than HDD.*

As an initial investigation, we installed a single *test* storage device at a time and allowed the system to boot and sit idle for 10 minutes while we measured the power consumption of the entire system. Our test system at idle, with no test storage device installed, used an average of 29 W over the 10 minute test period. With a single HDD added, it used 39 W. Using the flash-based SATA and NVMe SSDs had approximately the same idle system power of 31 W. With the Optane SSD installed, the system consumed noticeably more power than traditional SSDs, 34 W; but less than the HDD. These values of power consumption for the entire system at idle are illustrated by the bottoms of the bars in Figure 1.

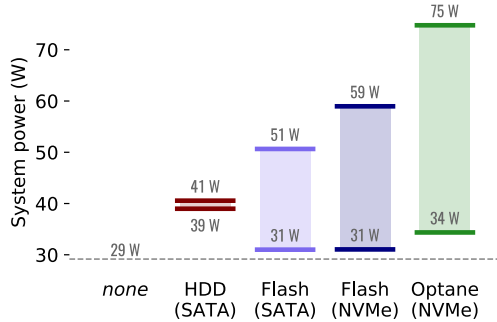


Figure 1: Total system power with a single storage device

HDDs have a notoriously high idle power consumption without performing any work due to their constantly spinning disks, damaging energy efficiency. This impact increases when more HDDs are added to system, such as with a RAID array [25]. This problem was significantly alleviated by the introduction of fully electronic solid-state flash media. However, we caution that new ULL storage devices may reintroduce this behavior to provide continuously low data access latency. We observed this behavior in the Optane SSD, and believe that its contributors could be a high performance controller design, internal physics of 3D XPoint technology, and the lack of support for Autonomous Power State Transitions (APST).

State	Oper.	Max Power	Perf. Rank	Enter latency	Exit latency
0	Yes	6.04 W	1st	0	0
1	Yes	5.09 W	2nd	0	0
2	Yes	4.08 W	3rd	0	0
3	No	40.0 mW	4th	210 $\mu$ s	1.5 ms
4	No	5.0 mW	5th	2.2 ms	6.0 ms

Table 3: Power states of Samsung 960 Flash NVMe SSD

APST, a feature configurable through the NVMe interface [1], allows the controller to autonomously transition the storage device into lower power states when idle. Using the command line tool *nvme-cli* [28], our tested Flash NVMe reports five APST power states as shown in Table 3, but the Optane SSD has only one. Each state is flagged as “operational” or “non-operational.” An operational state can service IO requests without violating its maximum power. The controller will automatically enter an operational state if the IO request doorbell is signaled while in a non-operational state. Table 3 also includes relative read/write performance rankings and the latency required to enter/exit power states.

APST plays an important role in idle energy consumption. If the storage device supports APST, then the host may actively instruct the controller to change states, or allow the controller to change states autonomously based on a given amount of idle time. For instance, the Linux kernel configures the controller of an APST supported device to enter the next lower-power non-operational state after an idle time of 50 $\times$  the sum of its enter and exit latency. It also forbids power states with an exit latency greater than a user-given QoS latency. Due to its associated performance cost, ULL SSDs might refrain from supporting APST; however, low idle power consumption is crucial for energy efficient systems and we believe that further research is necessary to implement idle power saving features while providing ULL IO performance.

**Observation 2.** *Active power consumption increases as device latency decreases.*

In order to analyze active power consumption, we run storage workloads across a range of behaviors from low to high intensity, and record overall power consumption of the system. The top of each bar in Figure 1 represents the maximum power across all workloads for each device. Notice that with lower device latency, active power consumption increases. The power range of the system from idle to active is, of course, not due solely to the storage device, as there are other system components such as CPU and RAM that are impacted by ULL IO. Newer storage technologies are often introduced with the appeal of greater performance, but it should be noted that there is a cost in the form of energy, especially due to their impact on other system components, as we discuss later.

### 3.5 Read vs. Write Power Consumption

Many performance issues of flash technology stem from its “erase before write” characteristic and consequent write amplification phenomenon, yielding inconsistent performance

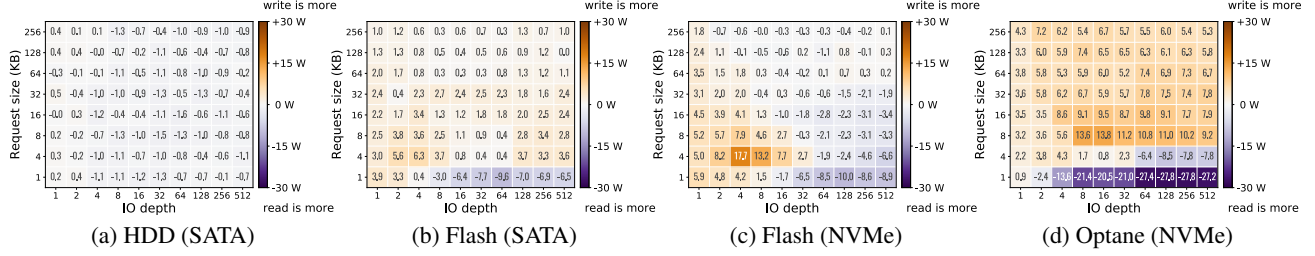


Figure 2: Difference in power (watts) between random write and random read ( $W - R$ ) *fio* microbenchmarks

during garbage collection. Moreover, it takes longer to write to flash memory than to read from it, which creates what we call “RW asymmetry,” in this case performance asymmetry. On the other hand, Wu et al. observed that the Optane SSD yields a similar performance for reads and writes [13], which makes it *performance symmetric*. In this section, we explore RW asymmetry of storage generations in terms of energy.

**Observation 3.** *From older to newer storage generations, devices move from energy symmetric to energy asymmetric.*

Figure 2 shows the difference between power consumption of writes and reads ( $W - R$ ) for random workloads. The HDD (Fig. 2a) drew approximately the same power for reads and writes, and is therefore *energy symmetric*. This is mainly due to the motor spinning the platter whether the RW head is reading or writing. We observed that flash-based SSDs (Figs. 2b & 2c) are *roughly energy symmetric*, causing approximately the same power consumption for reads and writes; but asymmetries begin to emerge. On the other hand, we realized that the Optane SSD (Fig. 2d) is *energy asymmetric* and it consistently used more power for writes than for reads, except for tiny requests of 1 KB that are suggested to be avoided [13]. For tiny requests, the bottom row shows greater power consumption for reads than for writes. While the energy asymmetry behavior deserves more research to understand the internal characteristics involved; it can potentially be exploited for energy-efficient hybrid drive designs combining 3D XPoint with flash or magnetic storage media, where requests can be redirected to the more energy-efficient option depending on their RW energy consumption.

### 3.6 Energy Proportionality

Barroso and Hölzle describe energy proportional machines in their seminal work as consuming energy in proportion to the amount of work performed [29]. A characteristic feature of this proportionality is a wide dynamic power range. The ideal energy proportional machine uses zero power at zero utilization (idle) and its maximum power at its peak utilization.

**Observation 4.** *Newer technological advancements in the storage subsystem lead to better energy proportionality.*

With newer innovations from each device, the overall system shows a greater dynamic power range. We define storage performance utilization as throughput (IOPS) normalized to peak throughput for each device. Figure 3 illustrates system power usage (power normalized to the peak power observed

in our system) as a function of storage performance utilization for 4 KB requests. We show the trend using a quadratic curve.

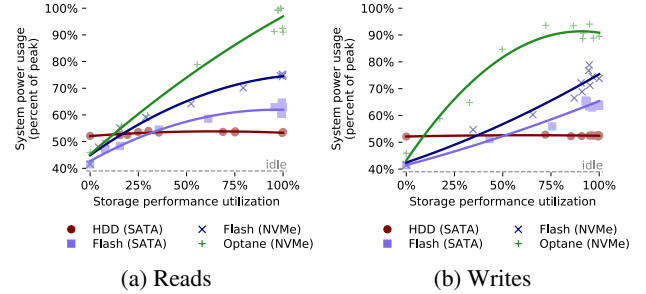


Figure 3: Energy proportionality

Based on Figure 3, the HDD is clearly not energy proportional. Its power usage is flat as it consumes approximately the same power regardless of the amount of work it performs. The traditional flash-based SATA and NVMe SSDs, on the other hand, are more energy proportional as they show an increase in IO performance with greater power usage. However, the Optane SSD is clearly the most energy proportional as it shows the greatest range of power usage, spanning more than half the peak system power, which makes it ideal for use in energy-efficient data centers.

### 3.7 Impact on System Software

Energy consumption is affected not only by the storage device, but also by the system software that runs it. An increase in system software load can cause other components such as CPU and memory to consume more power.

**Observation 5.** *As device latency decreases, the pressure on system software increases, resulting in increased overall energy consumption.*

With each newer storage technology, we observed the expected latency improvement in Table 1. Figure 4a shows the latency of 4 KB random reads for the devices tested, where the traditional SSDs (middle) are around two orders of magnitude faster than the HDD, and an order of magnitude slower than the Optane SSD. As we put more requests into the system by increasing the IO depth, this relationship did not change. However, with lower latency, a device can simply serve more requests in the same amount of time, and this increase in throughput puts greater pressure on system software.

One major indicator of the pressure on system software is the number of storage interrupts (NVMe or AHCI) per

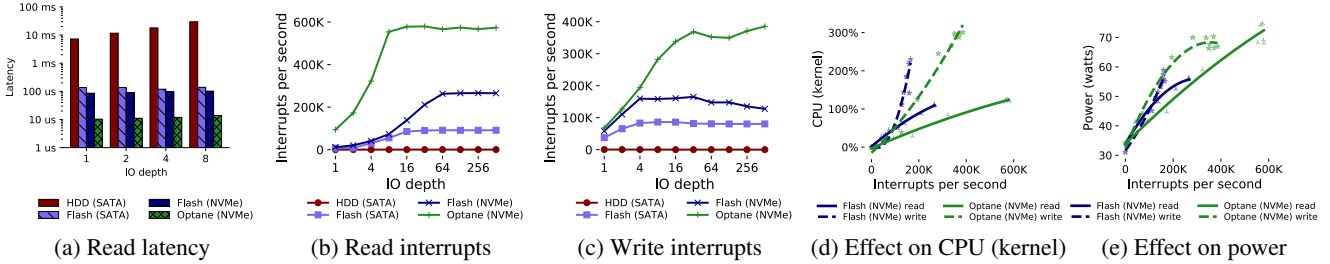


Figure 4: Impact on system software

second that the OS handles. We measured the number of interrupts during each storage workload by querying the kernel (`/proc/interrupts`) beforehand and afterwards. The SATA devices use only one AHCI interrupt number; however, in order to allow for greater parallelism, the NVMe driver with multi-queue (`blk-mq`) uses a separate interrupt number per queue, each with its own processor affinity. For our devices that use the NVMe interface, we sum the interrupts for all queues. Figures 4b and 4c show the average number of interrupts per second for 4 KB random reads and writes, with increasing IO depth. As we push the system, the storage interrupt rate can get quite high: nearly 600K/sec, which is a few orders of magnitude greater than even the rate of timer interrupts. This raises questions about the energy efficiency of using interrupt based IO, in addition to its known performance hit due to context switches. An interesting research question here is whether full or hybrid polling techniques can be more energy efficient than the existing interrupt based design for ULL devices.

Increased pressure on system software is also indicated by greater kernel CPU utilization; however, we realized that CPU utilization by itself might be misleading to design energy-efficient systems; researchers should always consult overall power consumption. Figures 4d and 4e show CPU utilization and power consumption, respectively, as a function of storage interrupts per second for 4 KB random reads and writes. In Figure 4d, writes for the Optane SSD caused around 300% system CPU usage with the interrupt rate of 400K/sec, while reads causing only 100% system CPU usage had an interrupt rate of 600K/sec. A similar imbalance exists for the Flash NVMe as well. However, their overall power consumption in Figure 4e is not directly proportional to their CPU utilization, where peak power consumption for reads is slightly greater than writes. Nevertheless, for the same interrupt rate (performance) in Figure 4e, writes are clearly more power

hungry than reads, even for a performance symmetric device such as the Optane SSD. One obvious contributor is Optane’s energy asymmetric behavior as shown in Section 3.5; however, we believe that write-based performance optimizations designed with flash characteristics in mind, including IO merging, prefetching, log structuring, and buffering, might exacerbate this situation further, and deserves more research.

### 3.8 Impact on Overall Energy Efficiency

For overall energy efficiency, we measure the amount of work done per unit energy. Since we have two measurements of performance, we provide two metrics of energy efficiency: one based on bandwidth, the other on throughput. The first metric, “bytes per joule” is the ratio of bandwidth (bytes per second) to power (watts) since one watt is one joule per second. This metric adjusts for requests of various sizes. The second metric, “IOs per joule” is the ratio of throughput (IOPS) to power (watts), and is used to compare for a fixed request size.

**Observation 6.** *For the same IO depth, energy efficiency as bytes per joule increases as request size increases.*

Figure 5 shows the overall energy efficiency as bytes per joule for random read workloads, where the Optane SSD is the “greenest” choice by performing the most work per unit energy. We observed the same trend for writes as well. In addition, we notice that for any given IO depth and storage device, larger requests consistently provide better overall energy efficiency measured as bytes per joule. Even though a larger request has more data to be transferred than a smaller request, it seems that the energy cost of transferring additional data in one request is less significant than the cost of managing the request itself and the pressure put on system software.

**Observation 7.** *For the same request size, energy efficiency as IOs per joule is coupled to internal device parallelism.*

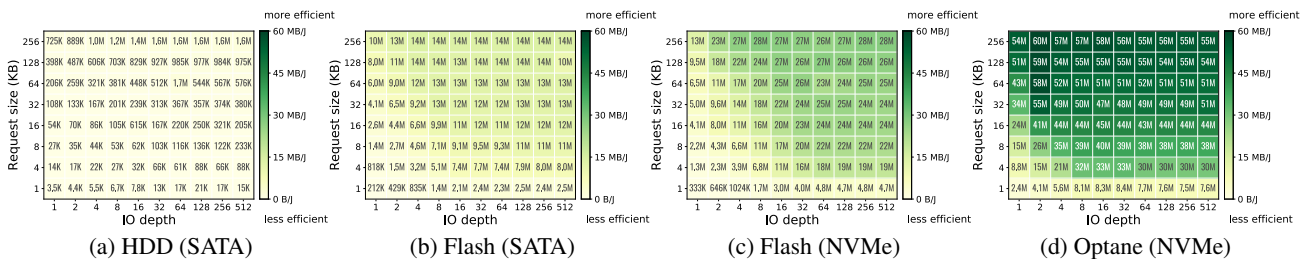


Figure 5: Energy efficiency in bytes transferred per joule



Figure 6 interprets the overall energy efficiency in IOs per joule, which we use to clearly observe the most energy-efficient IO depth given a request size for our tested NVMe devices. For the Optane SSD shown in Figure 6a, power consumption increases as IO depth increases for the same request size; however, this increased power consumption does not correspond to increased throughput after its internal parallelism is saturated. Others have observed saturated performance for ULL SSDs beyond an IO depth of 7 in Optane SSD [13] and 6 in Z-SSD [14]. With this work, we complement these observations by also showing a negative impact on energy efficiency beyond this parallelism saturation. Therefore, we see that on the right side of Figure 6a, overall energy efficiency begins to decrease for the Optane SSD. On the other hand, we do not observe the traditional Flash NVMe showing this behavior in Figure 6b due to its richer internal parallelism; in other words, increased power consumption corresponds to continuously increased IO performance for the Flash NVMe.

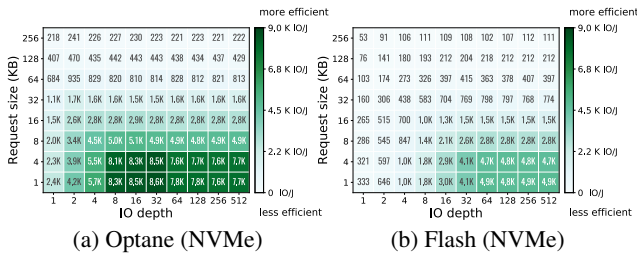


Figure 6: Energy efficiency in IOs per joule

Observations 6 and 7 outline best practices for overall energy efficiency. As described in Section 3.6, the Optane SSD has the greatest energy proportionality since its power consumption scales better than previous storage generations based on its range of throughput. Although Optane’s peak power consumption is higher, it still yields a better energy efficiency as measured in bytes per joule and IOs per joule.

## 4 Conclusion

We characterized the power consumption behavior of the Optane SSD and observed that it has undesirable idle power consumption, it consumes higher power at peak load compared to older storage generations, and it is energy asymmetric where writes cost more energy. Nevertheless, the Optane SSD stands out as the most energy-efficient and energy-proportional storage generation making it ideal for data centers. We further analyzed the energy implications of system software design, storage hardware design, and IO behavior for ULL performance, by detailing open questions in building energy-efficient systems supporting new generation ULL storage devices.

## Acknowledgments

We thank the anonymous reviewers and Myoungsoo Jung, our shepherd, for their feedback. This research was supported in part by the U.S. National Science Foundation (NSF) under grants CNS-1657296, CNS-1828521, and OIA-1849213.

## 5 Discussion

In this section, we introduce open questions requiring discussion and further research:

- I. **Rethink system software for energy efficiency:** Are there opportunities to make system software more energy efficient? Given the negative impact of low latency block devices on operating systems, are there ways to reduce or simplify system software to save energy? What would be the effect of classic or adaptive/hybrid polling techniques on energy efficiency compared to existing interrupt based design? Can polling be more energy efficient than interrupts for ULL devices? How about richer vs. simpler IO submission/completion mechanisms of *blk-mq* and the *NVMe* driver; can they be designed in a more energy efficient way considering ULL device behaviors? What are the implications of file system designs, IO scheduling algorithms, and write-based optimizations such as IO merging, prefetching, log structuring, and buffering on the energy efficiency of ULL storage?
- II. **Investigate read/write asymmetry:** What is the cause of read/write imbalance in power consumption for the 3D XPoint technology? Why do writes use more system CPU but trigger fewer interrupts? Are there any existing system-level design choices based on previous storage technologies that get in the way of 3D XPoint? Otherwise, previous work indicates an equal treatment of reads and writes in 3D XPoint in terms of performance [13].
- III. **Investigate the energy efficiency of alternative user- and kernel-space IO interfaces:** What is the effect of kernel-bypass storage architectures, such as SPDK [30] on overall energy efficiency? Would bypassing the kernel and implementing the NVMe driver in user space be more energy-efficient, or cause greater energy consumption due to its enforced polling based IO completion mechanism? What is the energy efficiency of the new *io\_uring* kernel IO interface, that allows asynchronous IO in polled mode with a dedicated polling thread [31–33]?
- IV. **Investigate energy-efficient hybrid drives with ULL SSDs:** What are the energy-efficient design strategies for combining ULL SSDs with traditional SSDs or HDDs? Can we use request redirecting techniques exploiting energy asymmetry in 3D XPoint? What about when the ULL SSD is used as a caching layer?
- V. **Mind energy efficiency as a whole:** What are the implications of recent storage performance optimizations (both at the device and system software levels) on overall energy efficiency? Due to their low power nature compared to other system resources like CPUs and memory, is the effect of the storage subsystem on the overall energy efficiency undermined? Do storage performance optimizations lose sight of overall energy efficiency?

## References

- [1] NVM express base specification, rev. 1.4.  
[https://nvmexpress.org/wp-content/uploads/NVM-Express-1\\_4-2019.06.10-Ratified.pdf](https://nvmexpress.org/wp-content/uploads/NVM-Express-1_4-2019.06.10-Ratified.pdf).
- [2] Matias Bjørling, Jens Axboe, David Nellans, and Philippe Bonnet. Linux block io: Introducing multi-queue SSD access on multi-core systems. In *Proceedings of the 6th International Systems and Storage Conference, SYSTOR '13*, pages 22:1–22:10, New York, NY, USA, 2013. ACM.  
<http://doi.acm.org/10.1145/2485732.2485740>.
- [3] Arash Tavakkol, Juan Gómez-Luna, Mohammad Sadrosadati, Saugata Ghose, and Onur Mutlu. Mqsim: A framework for enabling realistic studies of modern multi-queue SSD devices. In *16th USENIX Conference on File and Storage Technologies (FAST 18)*, pages 49–66, Oakland, CA, 2018. USENIX Association.  
<https://www.usenix.org/conference/fast18/presentation/tavakkol>.
- [4] Budget fair queueing (bfq) storage-I/O scheduler.  
[http://algo.ing.unimo.it/people/paolo/disk\\_sched/](http://algo.ing.unimo.it/people/paolo/disk_sched/).
- [5] Kyber multiqueue I/O scheduler. <https://patchwork.kernel.org/patch/9672023/>.
- [6] Mq-deadline multiqueue I/O scheduler.  
<https://github.com/torvalds/linux/blob/master/block/mq-deadline.c>.
- [7] Intel optane ssd 900p series product brief.  
<https://www.intel.com/content/dam/www/public/us/en/documents/product-briefs/optane-ssd-900p-brief.pdf>.
- [8] Samsung SZ985 Z-NAND SSD. [https://www.samsung.com/us/labs/pdfs/collateral/Samsung\\_Z-NAND\\_Technology\\_Brief\\_v5.pdf](https://www.samsung.com/us/labs/pdfs/collateral/Samsung_Z-NAND_Technology_Brief_v5.pdf).
- [9] Shigeo (Jeff) Ohshima. Scaling flash technology to meet application demands.  
<https://flashmemorysummit.com/English/Conference/Keynotes.html>.
- [10] Goke 2311 ssd.  
<https://www.bloomberg.com/press-releases/2019-09-30/goke-2311-series-drives-toshiba-memory-xl-flash-based-nvme-ssds>.
- [11] Gyun Lee, Seokha Shin, Wonsuk Song, Tae Jun Ham, Jae W. Lee, and Jinkyu Jeong. Asynchronous I/O stack: A low-latency kernel I/O stack for ultra-low latency SSDs. In *2019 USENIX Annual Technical Conference (USENIX ATC 19)*, pages 603–616, Renton, WA, July 2019. USENIX Association.  
<https://www.usenix.org/conference/atc19/presentation/lee-gyun>.
- [12] F. T. Hady, A. Foong, B. Veal, and D. Williams. Platform storage performance with 3d xpoint technology. *Proceedings of the IEEE*, 105(9):1822–1833, Sept 2017.  
<https://doi.org/10.1109/JPROC.2017.2731776>.
- [13] Kan Wu, Andrea Arpaci-Dusseau, and Remzi Arpaci-Dusseau. Towards an unwritten contract of intel optane SSD. In *11th USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage 19)*, Renton, WA, July 2019. USENIX Association.  
<https://www.usenix.org/conference/hotstorage19/presentation/wu-kan>.
- [14] Sungjoon Koh, Changrim Lee, Miryeong Kwon, and Myoungsoo Jung. Exploring system challenges of ultra-low latency solid state drives. In *10th USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage 18)*, Boston, MA, July 2018. USENIX Association. <https://www.usenix.org/conference/hotstorage18/presentation/koh>.
- [15] Jie Zhang, Miryeong Kwon, Donghyun Gouk, Sungjoon Koh, Changlim Lee, Mohammad Alian, Myoungjun Chun, Mahmut Taylan Kandemir, Nam Sung Kim, Jihong Kim, and Myoungsoo Jung. Flashshare: Punching through server storage stack from kernel to firmware for ultra-low latency SSDs. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*, pages 477–492, Carlsbad, CA, October 2018. USENIX Association.  
<https://www.usenix.org/conference/osdi18/presentation/zhang>.
- [16] Ivy B. Peng, Maya B. Gokhale, and Eric W. Green. System evaluation of the intel optane byte-addressable nvm. In *Proceedings of the International Symposium on Memory Systems, MEMSYS '19*, page 304–315, New York, NY, USA, 2019. Association for Computing Machinery.  
<https://doi.org/10.1145/3357526.3357568>.
- [17] S. Koh, J. Jang, C. Lee, M. Kwon, J. Zhang, and M. Jung. Faster than flash: An in-depth study of system challenges for emerging ultra-low latency ssds. In *2019 IEEE International Symposium on Workload Characterization (IISWC)*, pages 216–227, 2019.  
<https://doi.org/10.1109/IISWC47752.2019.9042009>.
- [18] Hai Huang, Wanda Hung, and Kang G. Shin. Fs2: Dynamic data replication in free disk space for improving disk performance and energy consumption. *SIGOPS Oper. Syst. Rev.*, 39(5):263–276, October 2005.  
<https://doi.org/10.1145/1095809.1095836>.

- [19] Eduardo Pinheiro and Ricardo Bianchini. Energy conservation techniques for disk array-based servers. In *Proceedings of the 18th Annual International Conference on Supercomputing, ICS '04*, page 68–78, New York, NY, USA, 2004. Association for Computing Machinery. <https://doi.org/10.1145/1006209.1006220>.
- [20] Akshat Verma, Ricardo Koller, Luis Useche, and Raju Rangaswami. Srcmap: Energy proportional storage using dynamic consolidation. In *Proceedings of the 8th USENIX Conference on File and Storage Technologies, FAST'10*, page 20, USA, 2010. USENIX Association. <https://www.usenix.org/conference/fast-10/srcmap-energy-proportional-storage-using-dynamic-consolidation>.
- [21] Tom Bostoen, Sape Mullender, and Yolande Berbers. Power-reduction techniques for data-center storage systems. *ACM Comput. Surv.*, 45(3), July 2013. <https://doi.org/10.1145/2480741.2480750>.
- [22] Zhichao Li, Kevin M. Greenan, Andrew W. Leung, and Erez Zadok. Power consumption in enterprise-scale backup storage systems. In *Proceedings of the 10th USENIX Conference on File and Storage Technologies, FAST'12*, page 6, USA, 2012. USENIX Association. [https://www.usenix.org/legacy/events/fast12/tech/full\\_papers/Li.pdf](https://www.usenix.org/legacy/events/fast12/tech/full_papers/Li.pdf).
- [23] A. Wildani and E. L. Miller. Semantic data placement for power management in archival storage. In *2010 5th Petascale Data Storage Workshop (PDSW '10)*, pages 1–5, Nov 2010. <https://doi.org/10.1109/PDSW.2010.5668053>.
- [24] Seokhei Cho, Changhyun Park, Youjip Won, Sooyong Kang, Jaehyuk Cha, Sungroh Yoon, and Jongmoo Choi. Design tradeoffs of SSDs: From energy consumption's perspective. *ACM Trans. Storage*, 11(2), March 2015. <https://doi.org/10.1145/2644818>.
- [25] Erica Tomes and Nihat Altiparmak. A comparative study of HDD and SSD RAIDs' impact on server energy consumption. In *19th IEEE International Conference on Cluster Computing (CLUSTER 2017)*, Honolulu, Hawaii, September 2017. <https://doi.org/10.1109/CLUSTER.2017.103>.
- [26] Onset. *HOBO® UX120-018 Data Logger (datasheet)*. <https://www.onsetcomp.com/datasheet/UX120-018>.
- [27] Jens Axboe. *Flexible I/O Tester*. <https://github.com/axboe/fio>.
- [28] nvme-cli: NVM-Express user space tooling for linux. <https://github.com/linux-nvme/nvme-cli>.
- [29] Luiz André Barroso and Urs Hölzle. The case for energy-proportional computing. *Computer*, 40(12):33–37, 2007. <https://doi.org/10.1109/MC.2007.443>.
- [30] Z. Yang, J. R. Harris, B. Walker, D. Verkamp, C. Liu, C. Chang, G. Cao, J. Stern, V. Verma, and L. E. Paul. Spdk: A development kit to build high performance storage applications. In *2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 154–161, 2017. <https://doi.org/10.1109/CloudCom.2017.14>.
- [31] Jens Axboe. Efficient io through io\_uring. [https://kernel.dk/io\\_uring.pdf](https://kernel.dk/io_uring.pdf).
- [32] Jens Axboe. Faster io through io\_uring. [https://kernel-recipes.org/en/2019/talks/faster-io-through-io\\_uring/](https://kernel-recipes.org/en/2019/talks/faster-io-through-io_uring/). Kernel Recipes, 2019.
- [33] Vishal Verma and John Kariuki. Improved storage performance using the new linux kernel i/o interface. In *2019 Storage Developer Conference (SDC)*, SDC'19. SNIA, Sep 2019. <https://www.youtube.com/watch?v=vjudg0lCEhQ>.