uFLIP: Understanding the Energy Consumption of Flash Devices

Matias Bjørling
IT University of Copenhagen
Copenhagen, Denmark

Philippe Bonnet
IT University of Copenhagen
Copenhagen, Denmark

Luc Bouganim
INRIA and U.Versailles
Le Chesnay, France

Bjørn Þór Jònsson
Reykjavik University
Reykjavik, Iceland

Abstract

Understanding the energy consumption of flash devices is important for two reasons. First, energy is emerging as a key metric for data management systems. It is thus important to understand how we can reason about the energy consumption of flash devices beyond their approximate aggregate consumption (low power consumption in idle mode, average Watt consumption from the data sheets). Second, when measured at a sufficiently fine granularity, the energy consumption of a given device might complement the performance characteristics derived from its response time profile. Indeed, background work which is not directly observable with a response time profile appears clearly when energy is used as a metric. In this paper, we discuss the results from the uFLIP benchmark applied to four different SSD devices using both response time and energy as metric.

1 Introduction

Energy efficiency is emerging as a major issue for data management systems, as power and cooling start to dominate the cost of ownership [2]. The key issue is to tend towards energy proportionality [1]—systems whose energy consumption is a linear function of the work performed, ideally with no energy consumed in idle mode. In this context, Tsirogiannis et al. [9] have argued that flash devices are very promising components. But, are there significant difference across flash devices? How much does power consumption depend on the IO patterns that are submitted? How stable are power measurements for a given device? Basically, how can we reason about the energy consumption of flash devices? This is the topic of this paper.

The first issue is: *How to measure energy efficiency?* At the scale of a data center[1], a few metrics have been introduced to measure and compare energy efficiency (e.g., Power Usage Effectiveness = total power for the facility / power for the data center). At the scale of a server or its components, the relevant metrics are the electrical power—expressed in Watts (W)—, or the energy—expressed in Joules (J)—drawn by the system. In this paper, we advocate measurements with a high temporal resolution (100 MHz sampling rate) that allow us to reason about the behavior of a flash device.

Copyright 2010 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.
Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

[1] See e.g., http://www.google.com/corporate/green/datacenters/measuring.html for some reference measurements

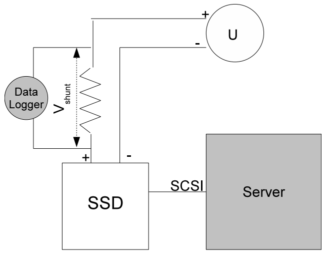

Figure 1: Diagram of the current shunt insertion set-up for measuring SSD energy consumption.

The second issue is: *What to measure?* We have designed a benchmark, called uFLIP, to cast light on all relevant usage patterns of current, as well as future, flash devices [5, 3]. uFLIP is a set of nine micro-benchmarks based on IO patterns, or sequences of IOs, which are defined by 1) the time at which the IO is submitted, 2) the IO size, 3) the IO location (logical block address, or LBA), and 4) the IO mode, which is either read or write. Each micro-benchmark is a set of experiments designed around a single varying parameter, that affects either time, size, or location. In [5], we measured and summarized the response time for individual IOs. In this paper, we focus on the energy profile of each experiment. While energy consumption cannot be traced to individual IOs, we can associate energy consumption figures to IO patterns, which helps us understand further the behavior of the devices.

Previous work covers different aspects of flash devices energy consumption: e.g., Tsirogiannis et al. focused on server power break-down resulting in key insights about the balance between idle and active modes, as well as CPU, hard disks and flash-based SSDs [9]; Mohan et al. [6] devised an analytical model for the power consumption of a flash chip (not a complete flash device); and Park et al. [7] focused on the design of low-power flash-based solid state drives (SSD). The study most similar to ours in its goals was conducted by Seo et al. [8]. They performed a few micro- and macro-benchmarks on three SSD devices as well as two hard drives. The micro-benchmarks focus on sequential reads, random reads, sequential writes and random writes as a function of IO size; while the macro-benchmarks are derived from file systems benchmarks. They use a multimeter to read the power value at 1KHz. The results exhibit a similar behavior between the three SSDs, which differs from the HDDs' behavior. The energy profile of each SSD corresponds to its throughput profile. The authors suggest that background work is specially important to explain the energy cost of random writes, but no details are given. In contrast, our study focus on a complete range of micro-benchmarks (the uFLIP benchmark), with a sampling rate of 100 MHz and a thorough study of the cases where the energy profile does not correspond to the performance profile. We show that there exist significant differences among flash devices in terms of energy profile.

Our contribution is the following: (1) we describe a set-up for measuring energy consumption of flash devices at high resolution; (2) we present the result of the uFLIP benchmark, using energy as a metric, for four flash-based solid state drives (SSD); and (3) we derive some lessons learned from our experiments.

2 Energy Measurement Set-Up

Our goal is to reason about power consumption at a fine granularity. For our measurements, we rely on shunt insertion: we measure the voltage across a shunt, which is proportional to the current flowing through the flash device. We use an oscilloscope equipped with a data logger to sample the voltage. The diagram in Figure 1 illustrates our set-up. The current shunt is installed on the high-side of an independent power source to guarantee stable voltage, the flash device being connected to the server via the data lines of the SATA connector.

We use a 1 Ohm resistor (±5% error) as a shunt, and we sample voltage at 100 MHz, i.e., at the order of ten μsec, which allows us to oversample the flash SSD IOs that are performed in tens of μsec. This does

SSD	Capacity (GB)	Idle (W)	Active (W)	Mean Idle Measured (W)	Std Dev Idle Measured
Memoright	32	1	2,5	0,96	0,3
MTron	16	0,5	2,7	0,99	0,05
Intel X25-E	32	0,6	2,4	0,54	0,05
Intel X25-E	64	0,6	2,6	0,57	0,05

Table 3: Capacity, advertised power characteristics (from the data sheet) and measured consumption in idle state

not give us the energy consumption of each IO, but we are working at the appropriate resolution to reason about the evolution of energy consumption in time. We obtain the energy consumed between two consecutive measurements E_t in Joule as follows: $E_t = (V_{shunt}(t)/R) * (U - V_{shunt}(t)) * \Delta$, where $V_{shunt}(t)$ is the voltage that we measure with the oscilloscope at instant t (in V), R is the resistor (in Ohm), U is the constant voltage provided by the power supply and Δ is the time elapsed between two consecutive measurements. We obtain the energy spent during an experiment by summing the relevant E_t.

We cannot rely on the response time measured in uFLIP to determine the duration of a run, as some work might be performed by the SSD after the last IO has completed. Indeed, this is one of the phenomenon we are interested in. We thus have to determine the beginning and end of a run from the energy trace. To do so, we first characterize the idle state for each device as a baseline. For each uFLIP run, we start collecting data approximately one second before the first IO is submitted and a few seconds after the last IO has completed to capture eventual asynchronous behavior. We use robust statistics to detect significant deviations from the idle state.

3 Results

We analyzed energy measurements from four different SSDs (see Table 1). The MTron SSD is from the first generation of devices on the market, while the device from Memoright and the two devices from Intel are more recent. Based on our work with uFLIP, we know that it is very hazardous to generalize our results to all SSDs. At best, these SSDs are representatives from interesting classes of flash devices.

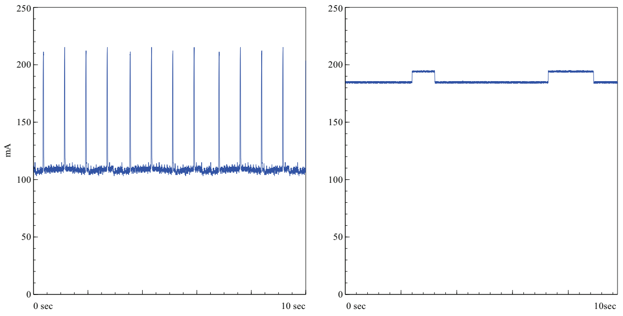

Figure 2: Intel SSD (left) and Memoright (right) in idle state

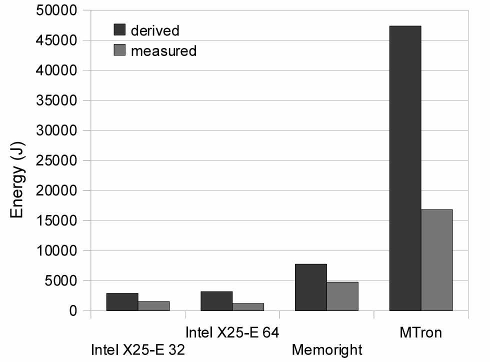

Figure 3: Total power consumption by device for the uFLIP benchmark

3.1 Idle State

In idle state, a flash device is not serving any IOs. An ideal device would not consume any energy in that state. In practice, flash devices consume some energy when idle, but much less than hard disks which make them attractive in the contextof energy efficient systems [9, 2]. The product data sheet give a typical power consumption in idle mode.

As explained in Section 2, we measure current draw in idle mode for all devices in order to establish a baseline for our uFLIP measurements. These results are summarized in Table 1. We make two observations. First, the measured results are not far from the figures from the data sheets (except for the MTron). Second, we observe that the Intel and Memoright devices actually perform regular work in idle mode, as Figure 2 illustrates, while the MTron does not (not shown).

3.2 uFLIP Energy Results

Figure 3 presents an overview of the total energy consumed while executing the uFLIP benchmark (1074 runs of hundreds of IOs). We compare the energy consumption actually measured with an energy estimate derived as the time it takes uFLIP to complete multiplied by the active power figure from the data sheet (see Table 1). We make two observations. First, the derived results are off by a factor of two or three. The derived energy consumed is consistently higher than the measured energy. Second, the energy consumed for the execution of the benchmark varies by an order of magnitude between the most efficient device in our study (Intel X25-E 64GB) and the least efficient (MTron).

uFLIP is composed of 9 micro-benchmarks. The most straightforward micro-benchmark (called *granularity*) concerns the performance of baseline IO patterns: sequential reads (SR), sequential writes (SW), random reads (RR), random writes (RW) as a function of IO size. We detail the results for this micro-benchmark in the rest of this section. Note that the scale of the y-axis is different for each device.

Figure 4 (a) and (b) show the response time and energy profiles for the Intel X25-E 64 GB. The results are similar for both Intel devices in our study, so we only show one model. With the Intel SSDs, the RR pattern utilizes more power than all other patterns—write included. It confirms the measurement based on response time and shows that write performance is obtained at the cost of suboptimal reads. Indeed, the high energy consumption for RR reveals a significant amount of work, which is not dictated by the characteristics of the underlying flash chips.

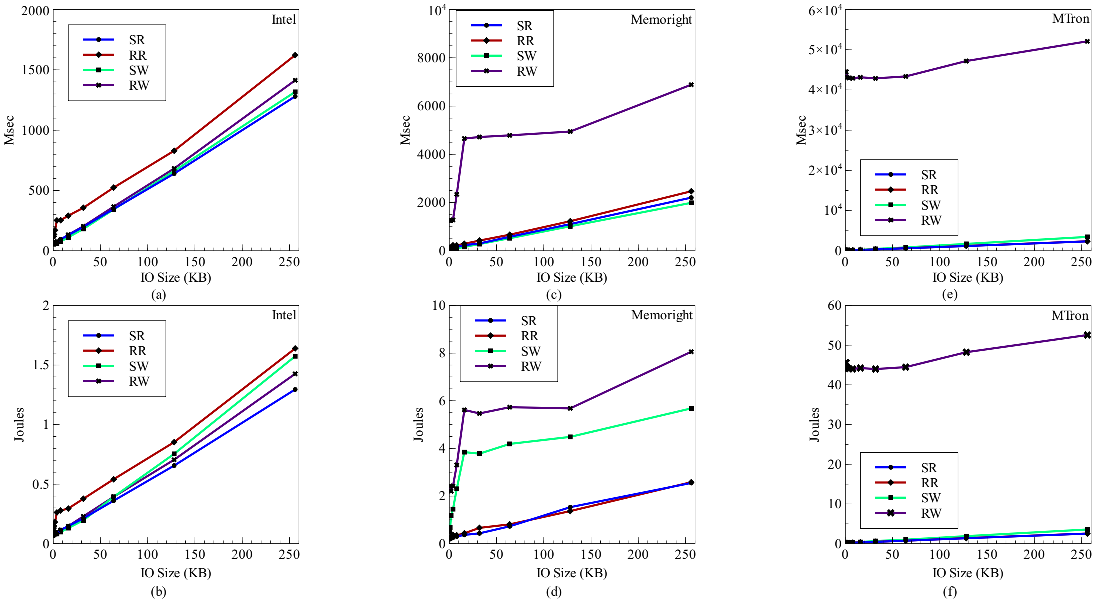

Figure 4: Time and Energy Profiles for the Intel X25-E 64GB (a),(b); Memoright (c),(d) and MTron (e),(f) for baseline patterns

Figure 4 (c) and (d) show the response time and energy profiles for the Memoright SSD. We can see that RW require additional work compared to other operations. While the energy required by SW is 50% lower than the energy required by RW, response time is five times faster. Obviously, the Memoright performs some work to hide the latency of SW. While SW response time is proportional to IO size, the energy profile reveals that energy-efficient SW should be done at the largest granularity.

Figure 4 (e) and (f) show the response time and energy profiles for the MTron SSD. Three interesting characteristics emerge from these graphs. First, the cost of sequential and random reads is very similar in terms of time and energy. Second, SW uses both more time and power than reads. Third RW are very slow but increase slowly with IO size. This means that even if the disk only writes 512 bytes, it is still doing nearly the same work as it would have for writing 256 KB.

This simple micro-benchmark exhibits significant differences between three classes of flash devices. In the next section, we explore further what the energy profiles tell us about the flash devices.

SSD	Memoright		MTron		Intel	
Category	SW	RW	SW	RW	SW	RW
Granularity	GWP		NS	S	S	None
Alignment	Ex		S		None	
Locality	GWP		NS	S	None	GWP
Partitioning	Ex		NS	GWP	None	S
Order	GWP		NS	GWP	None	GWP
Parallelism	Ex		NS		S	
Mix	GWP		S		None	GWP
Pause	Not applicable					
Bursts	Not applicable					

Table 4: Category of background work that depends on the submitted IO patterns. Ex - Extended, GWP - Grows with parameter, NS - Not significant, S - Small are the type of background work observed on the 9 uFLIP micro-benchmarks (see [5] for a complete description of these micro-benchmarks).

3.3 Device Characterization

One of the benefits of the energy profile is that it can reveal asynchronous activity, i.e., background work whose influence on response time is indirect. We identify three forms of background work[2]:

1. Background work independant of the activity of the SSD. We have seen in Section 3.1 that the Intel and Memoright devices exhibited a form of regular background work in idle state, while the MTron SSD does not.

2. Background work performed systematically whenever IOs are performed, regardless of the nature of those IOs—in our measurements this appears as a tail on each measurement. The MTron and MemoRight exhibit such a tail on every run throughout the benchmark, while the Intel devices do not. Figure 5 shows these tails on an example run (SR baseline pattern).

3. Background work that depends on the submitted IO patterns. A close look at the results from the numerous experiments revealed background work based on the submitted IO pattern. A first observation is that no device performed additional background work for experiments where only read operations where performed. We thus focused on experiments that contain writes and we classified the background work

[2] We refer the interested readers to [4] for more details about background work.

we observed in five categories: Extended (*Ex* - several seconds independantly of the micro-experiment parameter); Small (*S* - subsecond duration of background work independantly of the micro-experiment parameter); Grows with parameter (*GWP* - duration of background work increases regularly as a function of the micro-experiment parameter), and not significant (*NS*). Table 2 summarizes the type of background work per device and per micro-benchmark (see [5] for a complete description of each micro-benchmark).

Interestingly, Table 2 explains some of the behavior we observed in Figure 4 for the granularity micro-benchmark. On Figure 4 (b), we can see that, for the Intel device, the energy cost of SW increases faster than the cost of other operations. Indeed, SW trigger background work, while RW do not. On Figure 4 (d), for the Memoright device, we observed that the energy cost of SW is much higher than what could be deduced from the response time profile. Table 2 shows that both SW and RW trigger background work that grows with the IO size. A closer look reveals that the background work performed for SW lasts consistently much longer than the background work triggered by RW as illustrated in Figure 6 for an IO size of 128 KB.

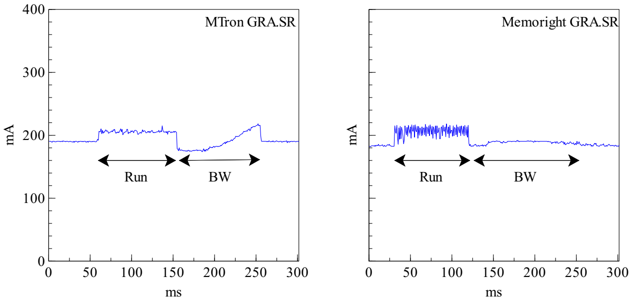

Figure 5: Systematic background work (BW) for MTron (left) and Memoright (right)

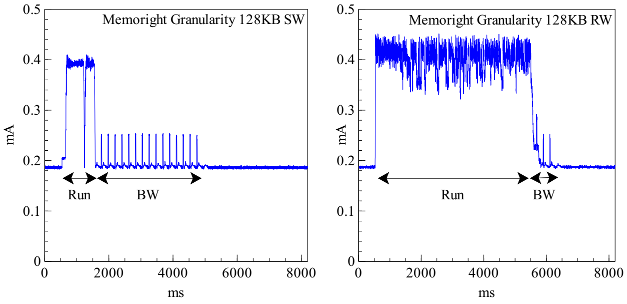

Figure 6: Background work (BW) for the granularity benchmark on Memoright SSD for SW (left) and RW (right) with an IO size of 128 KB.

3.4 Discussion

Measure. Do not simulate.. Flash devices are complex devices with layers of undocumented software and hardware. In particular, the Flash Translation Layer that is responsible for block management, wear leveling and error correction is a black box for DBMS designers. There is no straightforward way to define flash device model neither in terms of response time [3], nor in terms of energy consumption. The results from this section show that measuring response time and deriving energy based on the power figures from the data sheet is not an accurate method for estimating power consumption. Until we understand the characteristics of flash devices, the only way to reason about their energy consumption is to measure it.

Energy profiles reveal interferences. The energy profiles might reveal patterns of background work—which may or may not depend on the submitted IOs. This is very useful information for us in the context of the uFLIP benchmark. Indeed, we can rely on this information to plan the 1074 runs of a uFLIP execution so that we avoid interferences and minimize the pause in between runs. As a complement to the methodology we presented in [5], we can measure the energy consumption for baseline patterns and deduce the type of pause that we must introduce: minimal pause for read runs, pause depending on the nature of the run so that we can guarantees that there is no interference in between write runs.

No energy proportionality. The current generation of SSD is not energy-proportional. First, the devices that we studied consume energy in idle mode—some devices even perform systematic background work in idle mode. Second, flash devices implement trade-offs that require extra energy for some IO patterns (e.g., RW on MTron and Memoright or RR on Intel). Third, all devices implement a form of background work that is not compatible with energy proportionality. When that is said, the Intel devices come pretty close.

4 Conclusion

Our goal with this study was to get a better understanding of flash devices. We devised a set-up to perform high-resolution energy measurements. We ran the uFLIP benchmark. Our measurements cast some lights on the nature of background work, on the accuracy of energy estimates, and on the significant differences between classes of flash devices. Whether flash devices will evolve into energy proportional devices is an open issue. At least, flash devices should have a well-defined energy profile that can be leveraged to build energy efficient data management systems.

References

[1] Luiz André Barroso and Urs Hölzle. The case for energy-proportional computing. *Computer*, 40:33–37, 2007.

[2] Luiz André Barroso and Urs Hölzle. *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*. Synthesis Lectures on Computer Architecture. Morgan & Claypool Publishers, 2009.

[3] Matias Bjørling, Lionel Le Folgoc, Ahmed Mseddi, Philippe Bonnet, Luc Bouganim, and Björn Þór Jónsson. Performing sound flash device measurements: some lessons from uFLIP. In *SIGMOD Conference*, pages 1219–1222, 2010.

[4] Matias Bjørling Energy Consumption of Flash-based Solid State Drives Technical report, IT University of Copenhagen, 2010.

[5] Luc Bouganim, Björn Þór Jónsson, and Philippe Bonnet. uFLIP: Understanding flash IO patterns. In *CIDR*, 2009.

[6] Vidyabhushan Mohan, Sudhanva Gurumurthi, and Mircea R. Stan. Flashpower: A detailed power model for nand flash memory. In *DATE*, pages 502–507, 2010.

[7] Jinha Park, Sungjoo Yoo, Sunggu Lee, and Chanik Park. Power modeling of solid state disk for dynamic power management policy design in embedded systems. In Sunggu Lee and Priya Narasimhan, editors, *Software Technologies for Embedded and Ubiquitous Systems*, volume 5860 of *Lecture Notes in Computer Science*, pages 24–35. Springer Berlin / Heidelberg, 2009.

[8] Euiseong Seo, Seon-Yeong Park, and Bhuvan Urgaonkar. Empirical analysis on energy efficiency of flash-based SSDs. In *HotPower*, 2008.

[9] Dimitris Tsirogiannis, Stavros Harizopoulos, and Mehul A. Shah. Analyzing the energy efficiency of a database server. In *SIGMOD '10: Proceedings of the 2010 international conference on Management of data*, pages 231–242, New York, NY, USA, 2010. ACM.