Building Energy-Efficient Systems for Sequential I/O Workloads

John D. Davis
Microsoft Research – Silicon Valley Lab

Suzanne Rivoire
Sonoma State University

ABSTRACT

Solid-state drives (SSDs) have the potential to replace traditional rotational hard disk drives (HDDs) as the main storage media for computer systems. The power and performance characteristics of SSDs differ from those of HDDs, requiring the designers of hardware systems to reevaluate how they build and tune their systems. In this paper, we design an SSD-based system for highly energy-efficient sequential I/O. Using this system, we break the current record for the 10 GB category of the JouleSort energy-efficiency benchmark by more than a factor of 2. Furthermore, a single SSD combined with an ultra-low power processor is able to beat the previous record by 10%.

In addition, using a variety of hardware configurations, we contrast the impact of tuning on the power and performance of SSD-based systems with traditional HDD-based systems. We also demonstrate that by trading latency for power, we can achieve similar energy efficiency across a variety of systems, from embedded-class to desktop- and server-class systems. In other words, we show that energy-efficient computing does not require ultra-low-power components. Finally, we use this data to project the characteristics of an ideal energy-efficient data-intensive computing system using currently available components.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Design Studies, D.4.8 [**Performance**]: Measurements

General Terms

Measurement, Performance, Design

Keywords

Sort benchmark, Solid-State Drives (SSD), Performance, Tuning.

1. INTRODUCTION

Energy efficiency has become a driving metric used to evaluate computer systems. While it has always constrained small-scale devices like cell phones and laptops, it has emerged as a concern for large systems, especially in the data center [7, 8, 14]. Power consumed by the data center, although still small in absolute terms, is growing rapidly, doubling between 2000 and 2005 [24].

Data centers provide high performance for many common applications by using memory-resident databases and data structures, and these systems are limited by memory capacity and power consumption. However, another class of applications is I/O-bound, requiring multiple spindles to achieve maximum I/O performance. The push to design more energy-efficient computer systems combined with the emergence of much faster, higher-bandwidth, and lower-power I/O devices has created a new opportunity to rebalance the system for emerging data-intensive applications.

Current-generation solid-state drives (SSDs) have much lower latency and can sustain much higher random I/O bandwidth than any mechanical hard disk drives (HDDs) [39]. Some SSDs also have sequential performance that is much higher than that of any single HDD. This order-of-magnitude improvement presents new opportunities for hardware and software system builders. Furthermore, SSDs also promise low power at these higher performance levels [21].

Unfortunately, SSDs are not a panacea. NAND flash-based SSDs have low capacity, wear out, and cost at least two orders of magnitude more per gigabyte than HDDs. Based on manufacturers' specifications [15, 21, 28, 38], high-performance I/O systems can be built with a small number of devices that are more power-efficient than traditional HDDs, but have limited capacity. In addition, these devices have a complex Flash Translation Layer (FTL) that maintains the abstraction of the traditional block-addressable disk interface. The FTL performs wear leveling to extend the life of the SSD [2], data placement and management (similar to a log-structured file system) [9], error correction, and more [16, 22, 23], unbeknownst to the file system or operating system. The opacity of the FTL makes it difficult to reason about tuning SSD performance.

In this paper, we evaluate the energy efficiency of SSD-based hardware systems for data-intensive workloads compared to otherwise identical systems built with power-efficient HDDs. We use the JouleSort benchmark [36] as our driving application. The sort workload has been used in the database community since 1985 [4, 33] as a simple and representative way to identify emerging technologies for sequential-I/O-intensive tasks. The JouleSort sorting benchmark is designed to capture the energy efficiency of sorting systems.

We design SSD-based systems that break the current JouleSort record for datasets of 100 million records (10 GB) by more than a factor of 2. Furthermore, we design systems whose energy efficiency breaks the current record, although by a smaller margin, using a variety of devices and configurations.

In addition to breaking the previous JouleSort record, this work makes the following contributions. First, we demonstrate SSDs' benefits for full-system energy efficiency on sequential I/O workloads. Next, we investigate the tuning parameters for the benchmark and filesystem required to maximize performance and analyze the differences between tuning HDDs and SSDs for sort. In addition, we evaluate the energy efficiency of a variety of SSD-based systems and demonstrate that energy-efficiency plateaus can be achieved with many classes of components, making different tradeoffs between power and performance to maintain energy

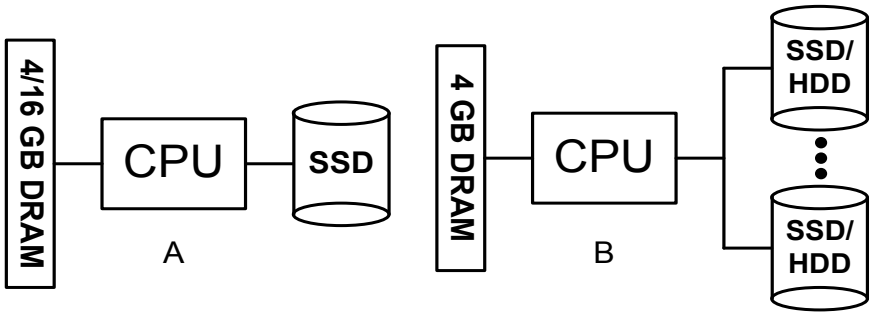


Figure 1. Hardware configurations evaluated: (A) single-SSD system with 4 or 16 GB of RAM and (B) multiple SSDs or HDDs combined for a high-performance I/O system.

consumption. We thus demonstrate that full-system energy efficiency does not require low-power components. Finally, we use the measured results to identify the characteristics of an ideal energy-efficient sorting system using current technology and estimate its JouleSort performance.

The rest of this paper is organized is follows. Section 2 describes our experimental infrastructure and the hardware evaluated. Section 3 presents our measured results for a variety of systems. Section 4 evaluates our SSD optimizations, discusses the process of building an energy-efficient system around SSD technology, and estimates what we believe to be the best energy efficiency obtainable with today's technology. We address related work in Section 5 and conclude with Section 6.

2. Experimental Infrastructure

This section describes the JouleSort benchmark; our hardware and measurement infrastructure; and the components and systems we evaluate in this paper. The components include seven different processors, two different memory configurations, and six different storage devices: three SATA SSDs, one PCI-Express x4 SSD, and two SATA HDDs. Whenever possible, we use the same hardware and software components across all of the systems.

2.1 JouleSort Benchmark Details

External sort workloads have been used for almost 25 years to evaluate the performance and price-performance of computer systems [4, 33]. Sort is a simple sequential-I/O-based workload that stresses the I/O subsystem as well as the processor, memory, and filesystem. It allows us to evaluate the end-to-end performance impact of new storage technologies, such as solid-state disks. The JouleSort variant of the benchmark allows us to evaluate system-level energy efficiency for this I/O-intensive workload.

The JouleSort workload is an external sort of 100-byte-long records with 10-byte keys. It requires that the input file be read from, and the output file written to, nonvolatile storage. There are three benchmark classes corresponding to different data set sizes. For our experiments, we used the 10 GB (10^8 records) file size due to the capacity limitations of our SSDs.

The metric for the JouleSort benchmark is the full-system energy (average wall power multiplied by wall-clock time) expended over the sort's execution, reported as records sorted per Joule. All of the reported results in this paper use the average power and time over five consecutive benchmark runs. Overall, the standard deviations of power and time were usually small, and these are shown as error bars on the graphs.

2.2 Hardware Infrastructure

We examined two types of hardware configuration in this paper, as Figure 1 illustrates. The first configuration, shown in Figure 1A, is used to establish a baseline characterization of the sorting capabilities of different processors and disk drives. In this configuration, we use a single SSD that contains both the OS and the sort data.

The second configuration, shown in Figure 1B, is the one we benchmark. In this configuration, multiple SSDs or HDDs are connected to the system via on-motherboard SATA connectors, a PCI-Express 4-port SATA adapter card, and/or a Fusion-io ioDrive. Unlike the first configuration, this configuration uses a separate disk for the OS image, and the other devices are used purely for storing the sort input and output files and temporary data. We use this configuration to achieve maximum JouleSort performance.

Our physical measurement infrastructure has three main components: the system under test (SUT), the digital power meter used to collect the full AC power consumption, and the computer that collects the AC power data from the meter (PC). Figure 2 illustrates the connections between these components. We interpose the WattsUp? Pro digital power meter [13] between the SUT and the wall power to capture the AC power and power factor once per second. We connect this meter to a separate PC over USB and use an internally developed tool to automatically capture and log the power measurements.

2.3 Software Infrastructure

The systems under test run either Windows Server 2008 or Linux 2.6.18_92.el5.x86_64. In either case, we use Ordinal

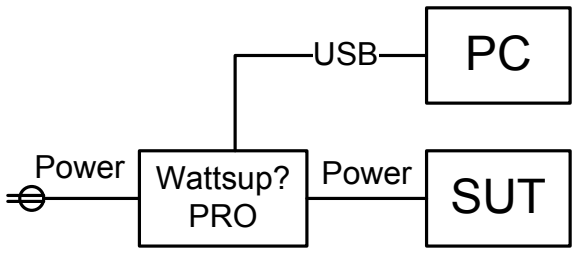


Figure 2. Measurement infrastructure.

*Table 1 – Systems evaluated in this paper. *Addressable memory.*

System Under Test (SUT)	CPU	Memory	Disk(s)	OS, FS
1A (embedded)	Intel Atom, single core, 1.6 GHz [1]	4 GB DDR2-800	1 Micron SSD	Windows Server 2008, NTFS
1B (embedded)	Intel Atom, dual core, 1.6 GHz [49]	4 GB DDR2-800	1 Micron SSD	
1C (embedded)	Via Nano U2250, single core, 1.6 GHz [47]	2.37 GB DDR2-800*	1 Micron SSD	
1D (embedded)	Via Nano L2200, single core, 1.6 GHz [47]	2.86 GB DDR2-800*	1 Micron SSD	
2 (mobile)	Intel Core2 Duo, dual core, 2.26 GHz, 25 W TDP [5]	4 GB DDR3-1066	1 Micron SSD (OS + Sort)	Windows Server 2008, NTFS
3A (desktop)	AMD Athlon, dual core, 2.2 GHz, 65 W TDP [30]	4 GB DDR2-800	1 Micron SSD (OS + Sort)	Windows Server 2008, NTFS
3B (desktop)			1 OS HDD + 7 Micron SSDs	Linux, XFS
3C (desktop)			1 OS HDD + 7 Samsung SSDs	
3D (desktop)			1 OS HDD + 7 Laptop 7200 RPM HDDs	
3E (desktop)			1 OS HDD + 3 Micron SSDs + 1 Fusion-io 80 GB ioDrive	
3F (desktop)			1 OS HDD + 1 Fusion-io 80 GB ioDrive	Linux, XFS/ext3
4A (server)	AMD Opteron, quad core, 2.0 GHz, 50 W TDP [44]	16 GB DDR2-800	1 OS SSD + 1 Fusion-io 80 GB ioDrive + 16GB DRAM	Windows Server 2008, NTFS
4B (server)			1 OS SSD + 1 Fusion-io 37 GB ioDrive + 16GB DRAM	
4C (server)			1 OS SSD + 1 Fusion-io 22 GB ioDrive + 16GB DRAM	
4D (server)			1 OS SSD + 5 Micron SSDs	
4E (server)			1 OS SSD + 5 Samsung SSDs	
4F (server)			1 OS SSD + 5 Intel SSDs	
4G (server)			1 OS SSD + 5 Laptop 7200RPM HDDs	

Technology's commercial NSort software [32] to perform the sort. This software uses asynchronous I/O to overlap reading, writing, and sorting operations. If the system contains enough memory to hold the entire data set, NSort performs a one-pass sort. Otherwise, it performs a two-pass sort, creating a set of temporary files on disk.

For each platform, we tuned the NSort parameters for I/O transfer sizes for the input, output, and temporary files; number of processes used; and in-memory sorting method. In general, we use large file transfer sizes, all of the available processors with on-demand processor frequency scaling, and radix sort. We turn off all unnecessary OS services.

For instrumentation, we use the Unix/Cygwin utility *time* to capture the benchmark run time. We then correlate this timing information with the power measurements taken over the entire run to compute the average power during the sort.

2.4 Systems evaluated

We evaluate the energy efficiency of a variety of different system configurations, as shown in Table 1. The systems in the first two groups (SUTs 1A-1D and 2) use an embedded or mobile processor, 4 GB of memory (although not all of the memory is addressable on all systems), and a single SSD. These systems' I/O interfaces cannot accommodate more than two or three SATA devices.

The main system under test, SUT 3, has a dual-core Athlon (desktop) processor with a thermal design power (TDP) of 65 W and 4 GB of memory. While larger memory configurations are possible, they are not required for sort and only penalize the system with higher system power. This system has four SATA ports on the motherboard and two first-generation PCI-Express slots, one of which is occupied by a graphics extender card. We evaluate this system with six different storage configurations (SUTs 3A-3F).

The server system, SUT 4, is based on a quad-core Opteron processor with 16 GB of memory, enough to do an in-memory (one-pass) sort of the 10 GB data set. The system is a dual-socket server, but only one socket and its related memory slots are populated for these experiments.

Each system's power supply is power-factor-corrected and has a power factor range of 0.95 to 1.0. The power supply for the embedded and mobile configurations is rated at up to 110 W. The desktop power supply is 80-plus efficient and is rated at 400 W; the server power supply is 560 W. The power supplies for all the

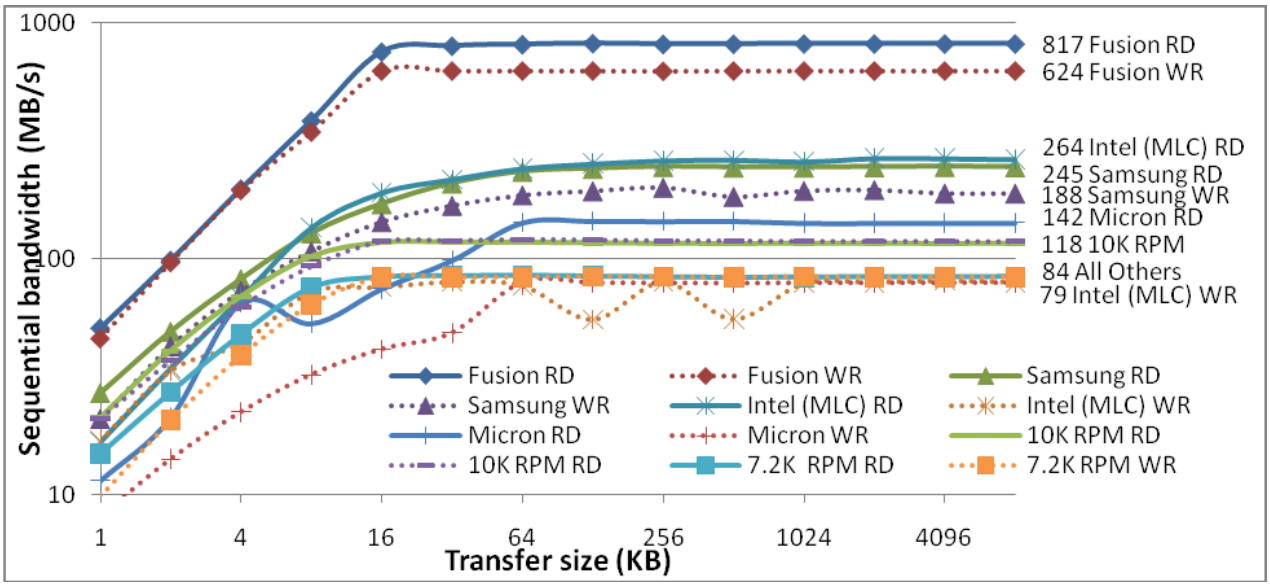


Figure 3. Log-log plot of storage device read (RD) and write (WR) ideal sequential bandwidth vs. transfer size for 256 MB chunks using the ATTO benchmark with the transfer rate listed on the right side of the graph in MB/s

systems are rated much higher than the maximum power draw of any of the tests that we observed.

The storage subsystems of these devices include several different kinds of components. The 7200 RPM laptop drives used in configurations 3D and 4G [40] are an updated version of the drives used in the system that set the initial JouleSort record [36]. The Intel SSD used in configuration 4F is a consumer-grade MLC device with very low active power (0.15 W) [21]. However, measured results show a much higher active power for other applications [29]. The Micron and Samsung devices are enterprise SLC SATA SSDs [28, 38], and the Fusion-io drive is a high-performance PCI-Express SSD [15].

Although cost is outside the scope of this paper, the range of prices extends from $0.20-$1.20/GB for the HDDs to $5-$35/GB for the SSDs. For the SSD-based configurations in this paper, the storage subsystem is responsible for the vast majority of the system cost.

3. Results

In order to understand the maximum capabilities of the disk drives, we first characterize the HDD and SSD raw performance in isolation in Section 3.1. We then run sort on the single-SSD configurations to prune the parameter-tuning space for the larger configurations. This step, discussed in Section 3.2, dramatically reduces the trials required for the systems with multiple storage devices. Section 3.3 presents our full-system sort results, which demonstrate a new level of energy efficiency for the JouleSort benchmark. This section also examines the differences between tuning HDDs and SSDs for energy-efficient sorting.

3.1 HDD and SSD raw performance

Our initial characterization of storage drives encompasses the five drives discussed in Section 2.4 as well as a 10,000 RPM SATA enterprise HDD. The data in Figure 3, on a log-log graph, was collected using the ATTO hard drive benchmark, which is part of the HBA utilities [6]. This benchmark varies the I/O transfer size for a given file size to discover the drive's maximum read and write bandwidth. All of these devices reach their maximum read and write bandwidth at transfer sizes between 16 KB and 64 KB.

As expected, the Fusion-io ioDrive provides by far the best bandwidth for a single device; it is also the most expensive single device. The ioDrive can be configured to have three different capacities with different write performances, but this benchmark did not show significant bandwidth differences among these three settings.

The 10,000-RPM HDD's bandwidth was comparable to that of the remaining drives, but its active power was much higher than that of the other devices considered, and so this device was removed from further consideration in the study. The laptop HDDs are similar to the Micron SSDs in both their write bandwidths and their power consumption (2 W). The consumer-grade Intel SSD shows the greatest asymmetry between read and write performance of any of the disks examined. The Intel SSD was included in this study because of its low active power.

Based on this initial I/O device characterization, we expected that the JouleSort benchmark record could be broken using the 7200 RPM laptop HDDs, or any of the SSDs, so we included these drives in our sorting systems.

3.2 Single-drive sort performance

Using embedded, mobile, and desktop systems (SUTs 1B, 2, and 3A), we tuned the NSort parameters to provide the highest possible performance. Because these configurations use only a single disk, we would expect the performance and efficiency of the systems to be limited by the lack of I/O bandwidth. However, these experiments provide a tractable set of parameters to vary in our later experiments with the multi-disk configurations.

We first determined the optimal transfer sizes for both the input/output files and the temporary files generated for two-pass sorts. Using the Micron SSDs, we varied these two transfer sizes from 4 KB to 8 MB and found that the best system performance,

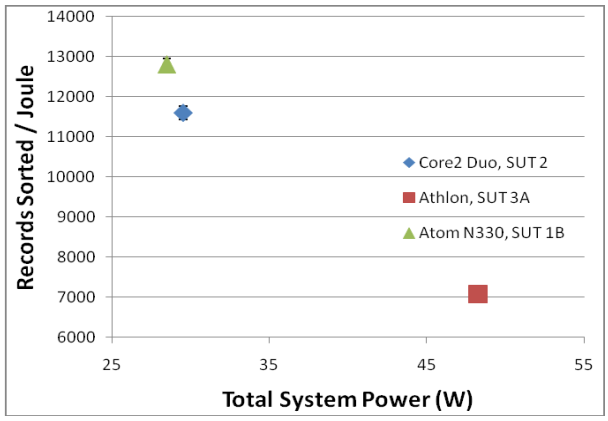


Figure 4. Records sorted per Joule vs. system power.

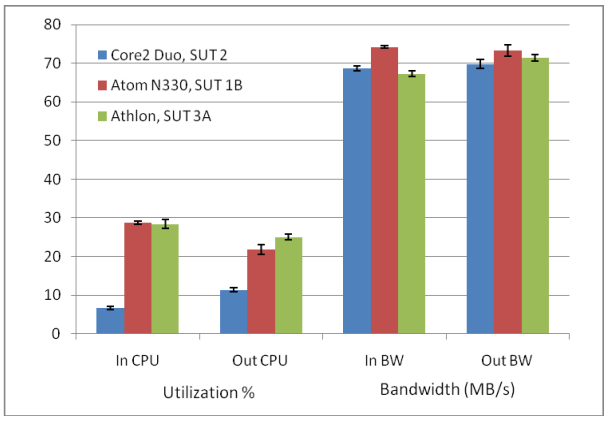


Figure 5. CPU utilization and SSD bandwidth. Both figures show results for single-disk configurations.

highest SSD bandwidth and lowest CPU utilization were obtained with transfer sizes set to 4MB for the input and output files and 2 MB for the temporary files. This graph is omitted for brevity.

Using these parameters, we ran the JouleSort benchmark on each system to produce Figures 4 and 5. Figure 4 shows the benchmark score (records sorted per Joule) of each system vs. the total system power. It shows that the mobile system matches the energy efficiency of the previous JouleSort winner with only a single SSD: 11,596±164 records sorted per Joule. The embedded system, using the dual-core Atom processor, exceeds the previous record, sorting 12,801±147 records per Joule. Furthermore, the absolute system power of these systems is 1/3 that of the previous winner. We have traded performance for lower power to arrive at similar energy consumption for the 10 GB sort.

Figure 5 shows the CPU utilization and I/O bandwidth of each system for each phase of the sort. As expected, all of these systems are bottlenecked by the single disk drive; that is, they sustain roughly the same SSD bandwidth. Even though the Core 2 Duo processor operates at 2.26 GHz and the Athlon operates at 2.2 GHz, the CPU utilization of the Core 2 Duo is significantly lower than that of the Athlon for both the input and output phases of the sort. Furthermore, the CPU utilization of the dual-core Atom is comparable to that of the Athlon. Thus, the Core 2 Duo-based system, given a different I/O subsystem, could sustain more sort bandwidth than the Atom- or Athlon-based systems; we build upon this observation in Section 4.

3.3 Full-system sort performance

After having characterized the individual disks and single-disk systems, we now evaluate the JouleSort benchmark performance of multi-disk desktop and server configurations. Finally, in addition to parameter tuning, we did perform some system-level power optimizations: We disconnected any unnecessary fans and accessories to reduce total system power consumption, and we changed inefficient power supplies to 80-plus power supplies, which improved the power factor and reduced the system power by 10%. Power delivery design can be improved further with fewer conversions and right-sizing the power supplies [27], which will benefit all classes of applications. Table 2 provides a summary of the performance for all systems tested. Note that SUT 3B did not use on-demand CPU frequency control, but manually controlled the CPU frequency to reduce the system power. This technique was not generally used, but could be used to further improve the results in Table 2.

3.3.1 Overall performance results

For optimal results on an energy-efficiency benchmark, all components of the system should be fully utilized. Since power consumption varies sublinearly with utilization, failing to fully exploit a component means wasting power. The desktop configurations (SUTs 3B-3F) came very close to this balanced ideal; the dual-core Athlon CPU was fully utilized during the output phase, as was the array of disks containing the output file. However, the input phase of the sort, which is less computationally intensive, was bottlenecked by the writes to temporary files. Table 2 shows the results for this system. The configuration with HDDs (SUT 3D) performs almost identically to the previous JouleSort winners. The configurations with SSDs, on the other hand, handily shatter the JouleSort record while consuming slightly less power than SUT 3D.

Configuration 4, the server configuration, is far from balanced; the quad-core Opteron CPU is severely underutilized in all of the results shown (fully utilizing only 2 of the 4 cores). However, Configuration 4 contains enough DRAM to sort the entire data set in memory. The advantages of a fully in-memory sort are twofold: the first is simply the time saved by not having to make the second pass to disk. The second is that the disks no longer need to be divided into two separate groups – one for the input/output files and one for the temporary files – whose bandwidths must match as closely as possible. Instead, all of the disks can be harnessed into one RAID0 array; this is the reason that the Fusion-io device can be used by itself, without any other disks, in Configuration 4C. This configuration surpasses the old 10 GB JouleSort record by more than a factor of 2. However, these results depend on Configuration 4 being able to sort the entire data set in one pass, which is currently impossible for larger data sets. The results from Configuration 3, using a two-pass sort, are more scalable.

3.3.2 Effect of in-core sorting algorithm

External sorts can use either radix sort or merge sort as an in-core sorting algorithm, and this choice affects the I/O characteristics of the transfers to temporary files in a 2-pass sort. In radix sort, the

*Table 2. Overall 10 GB sort results of configurations tested. Configurations 4C-4G are able to sort the entire data set in memory without the need for two I/O passes. (*) OzSort results are for a 100 GB data set; its records/J score for a 10 GB data set would likely be higher (within 10%).**This configuration used 2 GHz for the input and 2.2 GHz for the output phases of the sort and not on-demand processor frequency settings like all the other configurations.*

Configuration	Storage type	Time (sec.)	Power (W)	Records/J
CoolSort [29]	*Laptop HDD (5.4K rpm)*	*86.6 ± 0.4*	*99.3 ± 0.2*	*11628 ± 41*
OzSort [35]	*Desktop HDD (7.2 rpm)*	*n/a**	*104.2 ± 0.3*	*11597 ± 24**
1B	1 SSD (Micron)	274±3.16	29.5±0.01	12801±147
2	1 SSD (Micron)	292±3.55	29.5±0.3	11596±164
3B	7 SSDs (Micron)**	56.0 ± 0.3	88.9 ± 0.4	20079 ± 145
3C	7 SSDs (Samsung)	58.1 ± 0.3	81.0 ± 0.3	21241 ± 148
3D	7 Laptop HDD (7.2K rpm)	98.0 ± 2.2	90.3 ± 0.7	11307 ± 270
3E	SSD (Fusion-io + Micron)	56.2 ± 0.2	85.9 ± 0.2	20706 ± 92
4C	SSD (Fusion-io)	31.7 ± 0.08	127.3 ± 0.5	24755± 122
4D	SSD (Micron)	43.9 ± 1.2	124.9 ± 0.8	18247 ± 529
4E	SSD (Samsung)	33.0 ± 0.7	128.5 ± 0.9	23648 ± 547
4F	SSD (Intel)	40.6±0.4	118.5±0.6	20791±210
4G	Laptop HDD (7.2K rpm)	48.7 ± 0.2	129.8 ± 0.6	16069 ± 112

first pass consists of reading data sequentially from the input file and distributing its records into buckets, which are located in temporary files on a different set of disks. For merge sort, on the other hand, each chunk of the input file is sorted and then sequentially written to disk. Since the final output file is sequentially written in either case, merge sort's writes are purely sequential while radix sort's are more random. On traditional disks, when the data to be sorted is well distributed over a known interval, as it is in the sort benchmark, bucket (radix) sort tends to be slightly higher performing [32]. On laptop disks, this higher performance directly results in greater energy efficiency [36]. As Figure 6 shows, this result holds on SSDs despite their relative weakness with random writes (compared to sequential writes); the combination of on-disk write cache and large transfer sizes mitigates the increased randomness of the writes to temporary files.

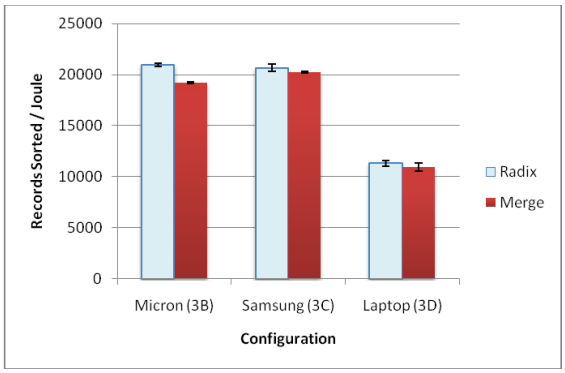


Figure 6. Effect of sort algorithm on sort performance.

3.3.3 *Effect of LVM RAID stripe size*

For maximum bandwidth, we stripe the input and output files across an array of disks when possible. The choice of stripe size affects both the power and performance of HDDs differently from the way it affects SSDs. Figure 7 shows the desktop system's power consumption as it varies with transfer size for SUTs 3B (Micron SSDs), 3C (Samsung SSDs), and 3D (laptop HDDs). The system power with SSDs is largely invariant with stripe size. The system with HDDs, however, shows a clear increase in power for the smallest stripe sizes; small stripes increase mechanical arm movement by increasing the number of disk seeks.

In terms of performance, however, SSDs are more sensitive to the LVM stripe size than HDDs, as shown in Figure 8. For the Micron and Samsung SSDs, using a small stripe size can cut the performance nearly in half compared to the optimal larger stripe sizes, since SSDs pay a serious performance penalty if the stripe size is smaller than the granularity of the logical page size and/or erase operation. Beyond that threshold, performance increases much more gradually.

Beyond the issue of stripe size, the error bars in these figures also show that the performance of SSDs is significantly more variable than the performance of HDDs. Furthermore, the factors that cause variability in HDD performance, such as placement of data on disk, can be tuned to some degree, but many of the factors that cause variability in SSDs reside in the FTL and are opaque at the system level.

3.3.4 *Effect of disk transfer size*

Figure 9 shows the effect of I/O transfer size on energy efficiency for Configuration 3B; other two-pass configurations showed similar results. Like the stripe size, the transfer size should

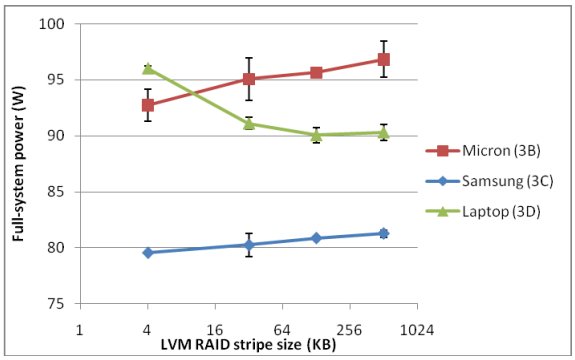


Figure 7. Variation of total system power with LVM stripe size on the desktop system. Note that the vertical axis starts at 75W.

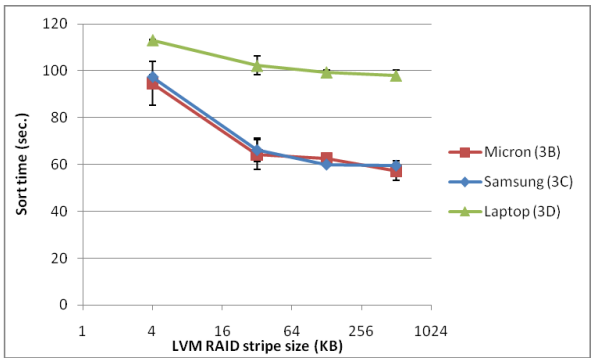


Figure 8. Variation of sort time with LVM stripe size on the desktop system.

exceed the granularity of the SSDs' logical page size and/or erase operation.

For the one-pass sorts using Configuration 4, we saw an advantage to larger transfer sizes for the input and output files. For this configuration, transfer size also affected sort performance when the output file was overwritten from a previous run rather than deleted and recreated for each run, as shown on the right side of Figure 10. With transfer sizes over 16 MB, performance is the same whether the file is overwritten or re-created. With smaller transfer sizes, however, performance is drastically lowered when the file is overwritten instead of erased. However, after running these experiments again, we have observed performance penalties for both scenarios, as shown in Figure 10. The performance penalty is as much as 50% for the Samsung drives, 36% for the Micron drives, and 29% for the laptop drives for an 8 MB transfer size. This performance degradation only manifests in the output phase by dramatically reducing the output bandwidth. The fact that both the laptop drives and the SSDs are affected suggests that the filesystem plays a role; the greater magnitude of the impact for SSDs suggests that the FTL can further exacerbate it. We are currently investigating this result.

4. Discussion

There are several implications that fall out of these results in the context of data-intensive computing. First, we provide a metric for determining how well these systems can be tuned. We then discuss our tuning methodology. Finally, we propose an energy-efficient system using currently available components that specifically targets data-intensive computing. This involves both processor selection and system-level optimizations.

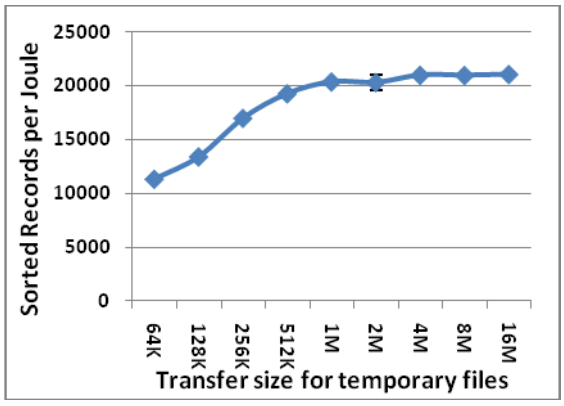


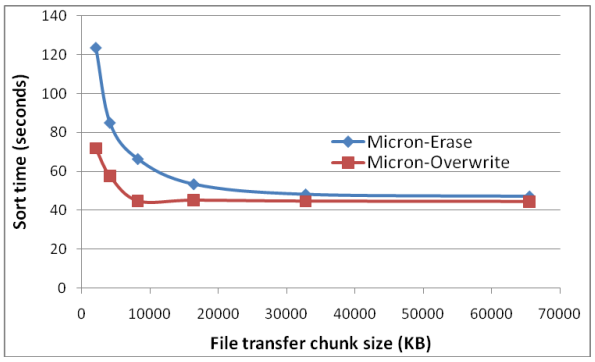
Figure 9. Effect of transfer size on JouleSort score for Configuration 3B (desktop with Micron SSDs).

4.1 Storage system tuning

In order to gauge the effectiveness of tuning the available filesystem- and application-level parameters, we can compare the I/O bandwidth achieved during sort with the peak sequential performance measured by the ATTO benchmark. The power and performance results for the laptop HDD-based system and the Micron SSD-based system demonstrate that it is possible to optimally tune the system, since they achieve the maximum read and write storage bandwidth reported by the ATTO benchmark.

For the other SSD-based systems, we were able to achieve either peak read or write bandwidth, but not both. In the Fusion-io case, because we are able to demonstrate higher write bandwidth with other SSDs, we can conclude that the write bandwidth of the Fusion-io ioDrive is limited by its FTL rather than the I/O interface. Likewise, the Samsung SSDs have comparable peak read bandwidth to the Intel SSDs (using the ATTO benchmark), but provide lower read bandwidth for sort, also demonstrating a possible software and/or FTL limitation. Finally, the Intel SSDs reached their peak write bandwidth, but not their peak read bandwidth. Overall, the combination of devices tested show that the physical I/O interface is not the bandwidth-limiting factor for the write bandwidth, but may be the limiting factor for the read bandwidth for SUTs 4E and 4F, which achieve only about 65% of their theoretical maximum read bandwidth, assuming no RAID overhead (this is the case for SUTs 4D and 4G).

There are two avenues of tuning, the traditional file system parameters and adjusting application parameters to compensate for the opaque FTL that manages the SSDs. We used traditional HDD benchmarks to get some insight into sequential read/write behavior of the SSDs and HDDs to direct our tuning. These benchmark results were especially useful for the in-memory one-pass sorts because they sequentially read and write data from persistent storage, either SSDs or HDDs. However, in the two-pass sort scenario, temporary files for binning and sorting are stored on the persistent storage and thus have more random reads and writes. By using larger file transfer sizes, these accesses can appear to be more sequential in nature. We provide the application parameters used to obtain the best results per platform



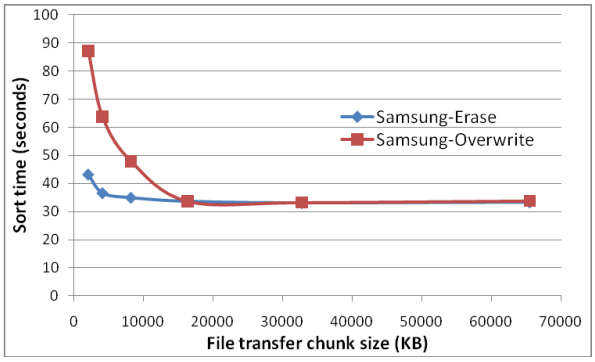


Figure 10. Sort time vs. I/O transfer size for SUT 4D and 4E when output file is overwritten or erased and recreated. In either case, output bandwidth is throttled.

in Table 3. Table 3 shows the input, output, and temporary file chuck sizes for each configuration that reported results in Table 2. These values provided the highest aggregate bandwidth and the shortest execution time for the sort. Single-pass sorts do not use temporary storage and are reported as N/A in the table.

Table 3. Application parameters for best JouleSort configuration for the multi-disk SUTs.

Configuration	Storage Type	Nsort FS parameters In, Out, Temp
1B	1 SSD (Micron)	1M, 1M, & 2M
2	1 SSD (Micron)	4M, 4M, & 2M
3B	7 SSDs (Micron)	8M, 8M, & 1M
3C	7 SSDs (Samsung)	8M, 8M, & 4M
3D	7 Laptop HDD (7.2K rpm)	16M, 16M, & 1M
3E	SSD (Fusion-io + Micron)	8M, 8M, & 16M
4C	SSD (Fusion-io)	64M, 64M, & N/A
4D	SSD (Micron)	32M, 32M, & N/A
4E	SSD (Samsung)	32M, 32M, & N/A
4F	SSD (Intel)	1M, 16M, & N/A
4G	Laptop HDD (7.2K rpm)	32M, 32M, & N/A

Our experiments have uncovered additional questions about SSDs, such as the one illustrated by the erase-vs-overwrite problem in Figure 10. Investigating these questions is currently hampered by the SSD's FTL because it obscures the operations and complexity of the SSD. Furthermore, we attempted to use system tracing to understand the actions of the OS and filesystem, but the considerable performance overhead of the system tracing masks the problem entirely. Additional investigation is part of our future work.

4.1.1 Filesystem parameters

As shown in Figure 3, all the SSDs and HDDs achieve their maximum bandwidth with transfer chunk sizes of 64 KB or larger for a sequential read or write of 256 MB. The 10 GB sort provides the opportunity to transform the input and output phases into largely sequential operations, thereby maximizing the bandwidth of either the SSD or HDD. However, in order to achieve the highest input and output phase bandwidth, we used transfer chunk sizes, set as a parameter for NSort, that were 2-3 orders of magnitude larger than 64 KB. Furthermore, we achieved the highest performance by decoupling the input and output file transfer sizes. In the two-pass sort case, we found that using the same output file transfer size for the temporary file transfer size garnered the highest output phase bandwidth.

The system was sensitive to the Linux Volume Manager (LVM) stripe size. For a four-disk array (SUTs 3B-3D), the minimum LVM size had to be greater than 256 KB for efficient input/output file storage, as can be derived from Figure 3 and shown in Figure 8. SSD performance is more sensitive to the LVM stripe size compared to the HDDs as shown in Figure 8. Furthermore, as shown in Figure 7, system power increased for the SSDs because the overall system could sustain higher bandwidth and utilization, while for the HDDs, the system power decreased due to reduced arm movement. Overall, the SSD-based systems increased power slightly, but yielded over 2x the performance, demonstrating that if the CPU and storage bandwidth are not fully utilized, there is the opportunity to try and squeeze more performance out of the system without greatly increasing the power.

4.1.2 Compensating for the FTL

The FTL manages the NAND Flash and provides an HDD abstraction. In order to do this, the FTL must hide aspects of the NAND Flash behavior from the file system. It does this by processing some requests in the background, over provisioning NAND Flash to guarantee a certain number of free, good blocks for bad block replacement, and log writes. Of all the SSDs, the Fusion-io ioDrive was the only device that provided any FTL tuning. This utility provided write performance tuning that reduced the size of the SSD. However, we observed no write bandwidth improvements using these settings. Ideally, if the FTL provided performance information, tuning flash-based I/O systems would be vastly easier.

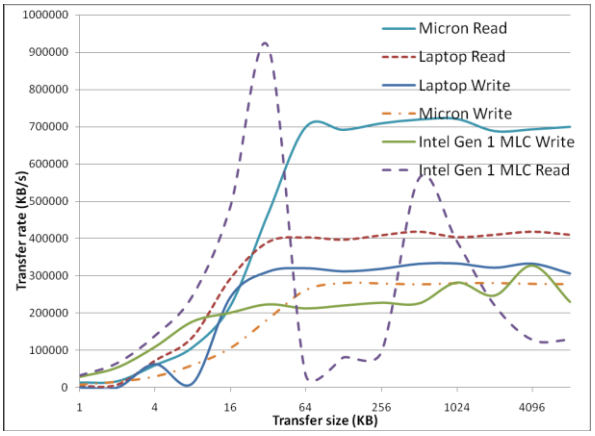


Figure 11. Read and write performance of 5 drive striped volumes run the ATTO benchmark.

With respect to the sort benchmark, we used large transfer chunks that span the LVM stripe and provide more sequential I/O opportunities for the SSDs. This also benefitted the HDDs. Providing large sequential IOs did not allow the SSDs to use their write caches because the transfer size was larger than the write cache, thereby forcing any normally interleaved background activities to happen and reduce the benchmark performance by disrupting the SSD IO operations. However, because the reading and writing were sequential for the 1-pass sort, this reduced the page and block mapping pressure caused by random writes, reduces fragmentation in the SSD that could lead to additional writes, and reduces related data movement for wear-leveling and garbage collection purposes.

Finally, single SSD behavior does not necessarily predict multiple striped SSD behavior for all the SSDs. As shown in Figure 11, in the cases of the HDDs and Micron SSDs, the striped volume created using the storage manager in Windows Server 2008 demonstrates predictable aggregate performance. The striped volume performance is the sum of the individual drives, as one would expect. The performance results for Figure 11 mirror the trends from Figure 3. However, as demonstrated by the Intel SSD results, these SSDs provided a more challenging environment to tune applications. They are very sensitive to the transfer size and stripe size. The ATTO benchmark is a very small benchmark and as a result may be hard for the adaptive FTL to adapt to. Although not provide here, similar results were found for the Intel SLC SSDs. Further experiments need to be run on the second generation Intel MLC SSDs to determine if a firmware update has fixed this behavior.

4.2 Building Systems

There has been a shift in the academic and industrial communities towards small building blocks using ultra-low power processors or *physicalization* for data-intensive computing [3, 10, 20, 26, 45], but we argue for better provisioned processors because SSDs drastically reduce seek time and provide greater bandwidth, requiring more powerful processors.

4.2.1 Processors for data-intensive computing

The use of SSDs for data-intensive computing has reduced the seek time dramatically and provided greater bandwidth, requiring more powerful CPUs to process the data. Previously, processors coupled with HDDs had to wait longer for less data. In this high-latency, low-bandwidth regime, using ultra-low-power processors is all that is required to match the I/O capability. However, moving forward, processors will require significant single-thread performance to process the data furnished by flash and other high-bandwidth, low-latency non-volatile memory technologies.

Revisiting Figures 4 and 5 provides some interesting observations for processor selection for this sort benchmark. If we were using only a single SSD, configuration 1B would provide the best power-performance, with 30% CPU utilization (out of 200%). However, configuration 2 uses the same SSD and OS, but only achieves about 90% of that performance. The interesting caveat is that configuration 2 had far less CPU utilization, only 6-7% (out of 200%).

As demonstrated in Figure 5, even a two-pass sort on a single SSD can require significant CPU performance. Although not included in Figure 5, the single-pass sort required intensive CPU resources that would be lacking from the embedded-class

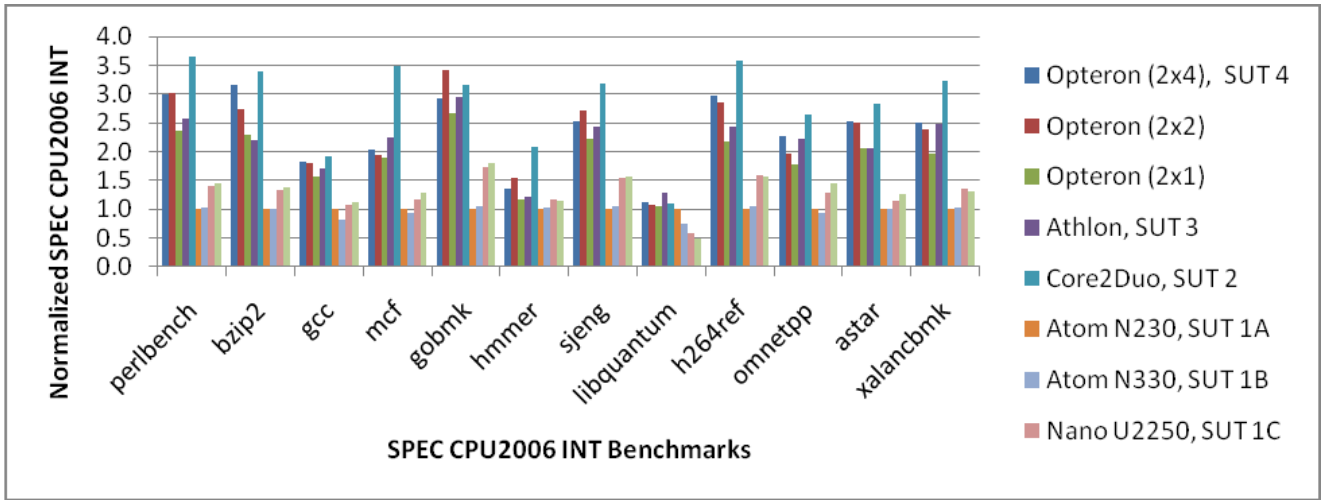


Figure 12. Per core SPEC CPU2006 integer performance normalized to the Atom N230 for the systems (embedded, mobile, desktop, and server processors) from Table 1 and legacy Opteron servers.

processors. To compare the single-thread performance of all the platforms, we used the SPEC CPU2006 integer benchmark, as reported in Figure 12. We used the same binary on all systems: 64-bit binaries compiled with Visual Studio. The results are normalized to the Intel Atom N230 single core processor, configuration 1A. In addition, we added results that span three generations of AMD Opteron server processors starting with a dual socket single core server (2x1), a dual socket dual core server (2x2), and the current generation dual socket quad core server (2x4). These results were included to quantify single core performance over time and with respect to the multi-core environment.

There are two surprising results. First, the Intel Core2Duo performs on par or better (especially for mcf and h264ref) than all other processors including the server processors. Second, and more surprising, is the fact that the single core Atom processor performs so well on the libquantum benchmark. As verified in Figure 12, the Core2 Duo has the largest single core performance capability compared to other processors.

Finally, only configurations 3 and 4 support ECC DRAM memory. Memory is the conduit to the processor, and memory errors are on the rise, especially for large systems [48]. In the context of the data center or large data working sets where the answer must be correct, systems without ECC are at a serious disadvantage and must rely on slow software solutions. We view ECC as a requirement for any data-intensive computing system. Since ECC support is not available for embedded and mobile systems, they are currently infeasible in this context despite their outstanding power-performance characteristics.

4.2.2 Building an ideal JouleSort system

The systems evaluated in this paper still waste I/O bandwidth and/or CPU utilization. This is especially true for the server-class SUTs. Thus, while we have broken through the current records-per-Joule barrier, these systems can accommodate significantly better results. Using the data from the single-SSD experiments on the Athlon (SUT 3B), Atom (SUT 1B), and the Intel Core2 Duo (SUT 2), we are able to extrapolate performance results for an even more efficient system using a hypothetical Intel chipset with the multi-disk array. This is possible because the bandwidth and sort times are the same on the two systems, SUT 2 and 3B, for both phases of the JouleSort benchmark. In fact, the CPU utilization is lower on the Intel Core2 Duo processor. Furthermore, we are able to add additional SSDs to the Athlon configuration and observe a linear relationship between the increased bandwidth and processor utilization. In order to determine the CPU power curve for both processors, we used an internally developed application that can exercise that processor at an arbitrary percentage utilization and simultaneously measure the power consumed by the CPU activity, the only fully utilized component. This enables us to extract the CPU and storage bandwidth power from the Athlon multi-disk experiments and extrapolate what the Intel Core2 Duo power would be given the same multi-disk configuration. Finally, we calculated the new full-system power based on the multi-disk array and an extrapolated CPU utilization based on the bandwidth. Using this methodology, we estimate that an Intel Core2 Duo system like SUT 2 could achieve about 29,000 records sorted per Joule with the appropriate I/O subsystem. This is a conservative estimate for an unbalanced extrapolated system: the CPU is still not necessarily fully utilized and the I/O read and write bandwidth are asymmetric. If we used an Intel SLC SSD with more symmetric bandwidth, we believe we could significantly improve the system and achieve at least 25% better with a fully balanced system or around 40,000 records sorted per Joule.

Our results also indicate that we can trade sort latency for absolute or peak system power, allowing increased flexibility for system provisioning in the data center, battery design for mobile devices, or other power-delivery-constrained scenarios. As we show in Figure 13, various systems can provide the same number of records sorted per Joule, but at very different system power levels: there is almost a 5x difference in the total system power. Likewise, we have observed that systems tend to cluster at particular performance plateaus. The plateau reached by previous JouleSort winners was around 11,000 records sorted per Joule.

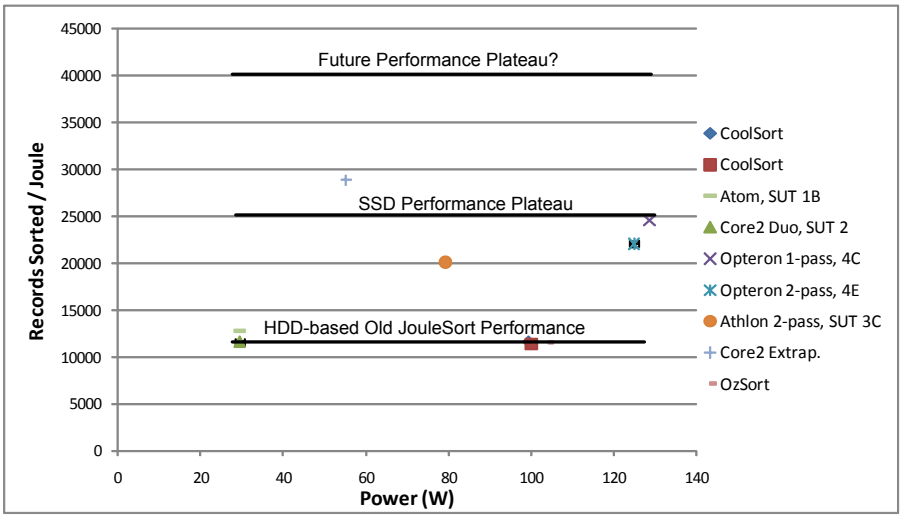


Figure 13. JouleSort 2-pass performance plateaus indicated by the black lines between the points for various system configurations demonstrating the latency-system power trade-off. The plateaus are defined by the storage and interconnect technology.

We have demonstrated that SSDs have ushered in a new plateau in the JouleSort benchmark. While we believe our results are on the lower end of this new plateau, the extrapolated results are much closer to where it will be in the future for this technology. Given our experience, we can look forward to the next plateau in Figure 13, which will develop when technologies like phase-change memory come to market.

5. Related Work

We discuss three areas of related work: energy-efficiency benchmarking, energy-efficient system design for data-intensive computing, and performance and power characterizations of SSDs. First, researchers have recently proposed several energy-efficiency benchmarks addressing different enterprise workloads. JouleSort [36], the benchmark we examine in this paper, is a sequential-I/O-intensive workload. In contrast, the SPECpower_ssj benchmark measures a CPU- and memory-intensive workload: server-side Java processing [43]. To account for the chronic underutilization of such servers in the data center [7], SPECpower_ssj exercises systems under a variety of loads, from light utilization to peak. Finally, TPC has begun work to incorporate an energy-efficiency metric into the TPC-C transaction processing benchmark [35, 46].

In the area of system design, power constraints in the data center have led researchers to look beyond traditional server-class disks to laptop disks and solid-state devices. Arrays of laptop disks [34] were used to set JouleSort records in 2007 and 2009 [42]. However, while laptop disks have a better ratio of bandwidth to power consumption than traditional server-class disks, their absolute performance is lower.

In recent years, solid-state drives have emerged to provide high-performance storage combined with power consumption similar to that of laptop disks. Previous JouleSort experiments with very low-end embedded processors and solid-state storage did not dethrone the combination of mobile processor and laptop disks [37]. In this case, the bottleneck was the disk interface rather than the processor or the disks themselves. With the emergence of solid-state disks for the server market, Graefe conjectured that they would be used to set a JouleSort record in the near future [17]. However, the main disadvantage of solid-state devices is their current high cost per gigabyte, which outweighs their energy cost savings for many workloads in non-power-constrained data centers [31].

Recently, researchers have proposed combining ultra-low-power processors and solid-state disks to meet the challenges of data-intensive computing workloads, which require a higher ratio of I/O to processor capability than traditional server systems provide [3, 10, 45]. Our work complements these projects by showing that ultra-low-power processors are not required for energy-efficient I/O-intensive computing.

Finally, the increasing interest in SSDs has led to several studies characterizing their performance and power consumption, a task made more difficult by the proprietary flash translation layer (FTL) in each device that maps logical block requests to physical operations on the flash hardware. Agrawal et al. [2] and Dirik et al. [12] studied the effects of SSD manufacturers' design choices on device performance in simulation, illuminating the cost/performance tradeoffs. Grupp et al. [19] characterize the performance, power, and reliability of the raw flash memory from a variety of manufacturers' SSDs, showing interactions between performance and reliability that can be exploited by the FTL.

Gray, in noting SSDs' potential in the enterprise domain, characterized their performance, and noted their problems with random writes [18]; Chen et al.'s more recent work found qualitative differences in random read and write performance between low- and high-end SSDs [11]. Seo et al. characterized the power consumption and energy efficiency of SSDs, finding that random writes not only have lower performance but also consume more power than sequential writes [41].

At the macrobenchmark level, Lee et al. contrasted the performance of solid-state and traditional hard disk drives for three common database operations: transaction commits, multiversion concurrency control, and sort and hash operations on temporary tablespaces [25]. Their findings on the performance of the sort workload are not identical to ours. Our results agree with their conclusion that, despite its increased random I/O, hashing/bucket sort does not fare worse than merge sorts on SSDs because of on-disk write caches. However, our results differ from their conclusion that SSD sort performance improves with decreasing transfer size for several reasons: first, our data set is orders of magnitude larger than the one used in their work (10 GB vs. 200 MB); second, the sort in their work is part of a DBMS, while our system is running only sort; third, these differences are exacerbated by the fact that they use a merge sort rather than a radix sort, which means that the benefits of large transfer sizes must be traded off with the disadvantage of smaller fan-in to the merge pass.

6. Conclusions

By using emerging SSD technology, we have produced record-breaking JouleSort benchmark scores using embedded, mobile, desktop, and server systems. We have shown that ultra-low-power processors are not required for energy-efficient computing and may not be the most energy-efficient solutions. Our experimental results demonstrate the ability to trade latency for power, maintaining equivalent energy consumption with a variety of different systems. In the context of a power-constrained data center, these results show that we can maintain the same energy efficiency, but trade reduced peak system power for increased latency, providing flexibility with respect to power and infrastructure provisioning.

We have examined the problem of tuning SSD-based systems for full-system energy efficiency and shown that their performance is more sensitive to tuning than that of HDDs. Given peak sequential bandwidth as a metric, we optimally tuned an HDD-based and SSD-based system. If performance feedback from the FTL were available, such as device utilization, request queue depth, caching statistics, garbage collection, and I/O arrival times, we would be able to tune the systems and attribute performance deficits more easily. Finally, it should be noted that the best tuning parameter values differ between systems that are very similar, such as SUT 1B and SUT 2 or any of the SUT 4 configurations.

We show that an array of SATA SSDs can match the performance of a PCI-Express SSD, demonstrating that the SATA physical interface can sustain similar high bandwidth. Not being able to

sustain peak bandwidths close to the aggregate individual read bandwidths of the Intel and Samsung SSDs indicates a bottleneck in the SATA interface. Likewise, the peak sort bandwidth was 25% below the peak bandwidth reported by the ATTO benchmark, indicating additional overhead from the OS, filesystem, and/or FTL. Unfortunately, several of our configurations lacked an adequate number of I/O interfaces to fully utilize the CPU. By re-engineering systems to incorporate more I/O interfaces, we can shift even more problems from being I/O-bound to being compute-bound. We have been limited by the building blocks provided by industry and have demonstrated by extrapolation that even more energy-efficient systems can be built, given the right interfaces. Regardless, our results still show a factor of two improvement over the previous benchmark winner.

Overall, SSDs, larger memories, and faster interconnects have been the enabling technology for this dramatic efficiency improvement, and we expect further improvements on this foundation. The challenges faced in future work will revolve around the software, [opaque] firmware, and hardware interactions of the SSDs and other new non-volatile memories for traditionally I/O-bound applications.

7. References

[1] Acer AspireRevo, available at http://aceraspirerevo.com/

[2] N. Agrawal, V. Prabhakaran, et al., "Design tradeoffs for SSD performance," in *Proc. of the USENIX Annual Technical Conf. (USENIX)*, Boston, MA, June 2008.

[3] D. Andersen, J. Franklin, et al., "FAWN: a fast array of wimpy nodes," in *Proc. of the 22nd ACM Symp. on Operating Systems Principles (SOSP)*, Big Sky, MT, Oct. 2009.

[4] Anon., "A measure of transaction processing power," *Datamation*, vol. 31, no. 7, April 1985, pp. 112–118.

[5] Mac mini, available at: http://www.apple.com/macmini/

[6] ATTO Disk Benchmark, available at http://www.attotech.com/software/disc/app0307.html

[7] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," *IEEE Computer*, vol. 40, no. 12, Dec. 2007, pp. 33–37.

[8] L. A. Barroso and U. Hölzle, *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*, Morgan-Claypool, 2009.

[9] A. Birrell, M. Isard, et al., "A design for high-performance flash disks," *ACM SIGOPS Operating Systems Review*, vol. 41, no. 2, April 2007, pp. 88–93.

[10] A. M. Caulfield, L. M. Grupp, and S. Swanson, "Gordon: using flash memory to build fast, power-efficient clusters for data-intensive applications," in *Proc. of the 14th Intl. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, Washington DC, March 2009.

[11] F. Chen, D. A. Koufaty, and X. Zhang, "Understanding intrinsic characteristics and system implications of flash memory based solid state drives," in *Proc. of the 11th Intl. Conf. on Measurement and Modeling of Computer Systems (SIGMETRICS)*, Seattle, WA, June 2009.

[12] C. Dirik and B. Jacob, "The performance of PC solid-state disks (SSDs) as a function of bandwidth, concurrency, device architecture, and system organization," in *Proc. of the 36th Intl. Symp. on Computer Architecture (ISCA)*, Austin, TX, June 2009.

[13] Electronic Educational Devices, "Operators manual: Wattsup? and Wattsup? PRO," available at https://www.wattsupmeters.com/secure/downloads/manual_rev_9_corded0812.pdf

[14] United States Environmental Protection Agency (EPA) Energy Star Program, "Report on server and data center energy efficiency," Aug. 2007.

[15] Fusion-io, "ioDrive data sheet," available at http://www.fusionio.com/PDFs/Data_Sheet_ioDrive_2.pdf

[16] E. Gal and S. Toledo, "Algorithms and data structures for flash memories," *ACM Computing Surveys (CSUR)*, vol. 37, no.2, June 2005, pp.138–163.

[17] G. Graefe, "The five-minute rule 20 years later (and how flash memory changes the rules)," *Communications of the ACM (CACM),* vol. 52, no. 7, July 2009, pp. 48–59.

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

[19] L. M. Grupp, A. M. Caulfield, et al., "Characterizing flash memory: anomalies, observations, and applications," in *Proc. of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture,* New York, NY, December 2009.

[20] J. Hamilton, "CEMS: low-cost, low-power servers for internet-scale services," in *Proc. of the 4th Biennial Conference on Innovative Data Systems Research (CIDR)*, Asilomar, CA, Jan. 2009.

[21] Intel, "Intel X18-M/X25-M SATA solid state drive product manual," available at http://download.intel.com/design/flash/nand/mainstream/mainstream-sata-ssd-datasheet.pdf

[22] T. Kgil, D. Roberts, and T. Mudge, "Improving NAND flash based disk caches," in *Proc. of the 35th Intl. Symp. on Computer Architecture (ISCA)*, Beijing, China, June 2008.

[23] H. Kim and S. Ahn, "BPLRU: a buffer management scheme for improving random writes in flash storage," in *Proc. of the 6th USENIX Conf. on File and Storage Technologies (FAST)*, San Jose, CA, Feb. 2008.

[24] J. G. Koomey, "Estimating total power consumption by servers in the U.S. and the world," Technical report, Lawrence Berkeley National Laboratory, Feb. 2007.

[25] S. Lee, B. Moon, et al., "A case for flash memory SSD in enterprise database applications," in *Proc. of the Intl. Conf. on Management of Data (SIGMOD)*, Vancouver, BC, June 2008.

[26] K. Lim, P. Ranganathan, et al., "Understanding and designing new server architectures for emerging warehouse-computing environments," in *Proc. of the 35th Intl. Symp. on Computer Architecture (ISCA)*, Beijing, China, June 2008.

[27] D. Meisner, B. T. Gold, and T. F. Wenisch, "PowerNap: eliminating server idle power," in *Proc. of the 14th Intl. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, Washington DC, March 2009.

[28] Micron Technology, Inc., "RealSSD™ P200 2.5-inch SATA NAND flash SSD," available at http://download.micron.com/pdf/datasheets/realssd/realssd_P200_2.5.pdf

[29] C. Moullas, "Intel X25-m Gen 2 & OCZ Vertex Power Consumption," available at http://www.cmoullas.net/reviews/44-ssd/90-intel-ocz-ssd-power-consumption

[30] MSI AA-780E Motherboard, available at: http://www.msi.com/index.php?func=proddesc&maincat_no=388&prod_no=1746#

[31] D. Narayanan, E. Thereska, et al., "Migrating server storage to SSDs: analysis of tradeoffs," in *Proc. of the EuroSys Conf.*, Nuremberg, Germany, March-April 2009.

[32] C. Nyberg and C. Koester, "Ordinal Technology—NSort home page," available at http://www.nsort.com/

[33] C. Nyberg, M. Shah, and N. Govindaraju, "Sort benchmark home page," available at http://sortbenchmark.org/

[34] A. E. Papathanasiou and M. L. Scott, "Power-efficient server-class performance from arrays of laptop disks," University of Rochester Computer Science Department Technical Report 837, May 2004.

[35] M. Poess and R. O. Nambiar, "Energy cost, the key challenge of today's data centers: a power consumption analysis of TPC-C results," in *Proc. of the Very Large Databases Conf. (VLDB)*, Auckland, New Zealand, Aug. 2008.

[36] S. Rivoire, M. A. Shah, et al., "JouleSort: a balanced energy-efficiency benchmark," in *Proc. of the Intl. Conf. on Management of Data (SIGMOD)*, Beijing, China, June 2007.

[37] S. Rivoire, M. A. Shah, et al., "Models and metrics to enable energy-efficiency optimizations," *IEEE Computer*, vol. 40, no. 12, Dec. 2007, pp. 39–48.

[38] Samsung server SSD, available at http://www.samsung.com/global/system/business/semiconductor/product /2009/6/11/202005SS805_Spec_200902.pdf

[39] P. Schmid and A. Roos, "14-way SSD hard drive roundup: flash SSDs compared," available at: http://www.tomshardware.co.uk/flash-ssd-hard-drive, review-31258.html

[40] Seagate Technology, LLC, "Product overview: Seagate Momentus 7200.3," available at http://www.seagate.com/docs/pdf/marketing/po_momentus_7200_3.pdf

[41] E. Seo, S. Y. Park, and B. Urgaonkar, "Empirical analysis on energy efficiency of flash-based SSDs," in *Proc. of the First Workshop on Power-Aware Computing and Systems (HotPower)*, San Diego, CA, Dec. 2008.

[42] R. Sinha and N. Askitis, "OzSort: Sorting 100GB for less than 87kJoules," April 2009, available at http://sortbenchmark.org/OzJoule2009.pdf

[43] Standard Performance Evaluation Corporation (SPEC), "SPEC power_ssj2008," available at http://www.spec.org/power_ssj2008/

[44] Supermicro AS-1021M-T2+B, available at: http://www.supermicro.com/Aplus/system/1U/1021/AS-1021M-T2+.cfm

[45] A. S. Szalay, G. Bell, et al., "Low-power Amdahl-balanced blades for data intensive computing," in *Proc. of the Second Workshop on Power-Aware Computing and Systems (HotPower)*, Big Sky, MT, Oct. 2009.

[46] Transaction Processing Performance Council (TPC), "TPC-Energy," available at http://www.tpc.org/tpc_energy/default.asp

[47] VIA Mini-ITX Board (1.6GHz VIA Nano CPU), Available at: http://www.via.com.tw/en/products/embedded/boards/index.jsp

[48] K. Yelick, "How to waste a parallel computer," keynote talk at *36th Intl. Symp. on Computer Architecture (ISCA)*, Austin, TX, June 2009.

[49] Zotac IONITX-A-U, available at: http://www.zotacusa.com/zotac-ionitx-a-u-atom-n330-1-6ghz-dual-core-mini-itx-intel-motherboard.html