USB Flash Drives as an Energy Efficient Storage Alternative

Olga Mordvinova #1, Julian Martin Kunkel #2, Christian Baun *3, Thomas Ludwig **4, Marcel Kunze *5

University of Heidelberg
Department of Computer Science
69120 Heidelberg, Germany
1 mordvinova@informatik.uni-heidelberg.de
2 kunkel@informatik.uni-heidelberg.de

* *Karlsruhe Institute of Technology, Institute for Scientific Computing*
76344 Karlsruhe, Germany
3 baun@iwr.fzk.de
5 kunze@iwr.fzk.de

** *University of Hamburg, Department of Informatics, c/o DKRZ*
20146 Hamburg, Germany
4 ludwig@dkrz.de

***Abstract*—Increasing energy costs and the trend to green solutions are generating growing interest in eco-friendly computing. Flash storage technology satisfies low-energy requirements and falling prices for flash storage have recently made it possible to replace conventional hard disks with solid state drives. These can provide higher performance when deployed properly but they are still too expensive for the low-cost server market.**

In this paper we analyze inexpensive flash drives and conventional hard drives with a focus on energy efficiency. Our main goal is to decide whether replacing hard drives with USB flash drives is a reasonable and economic way to build energy-efficient servers. Our evaluation shows that despite the high throughput of hard disks, flash drives have a better performance per joule. In some cases, inexpensive USB flash can compete not only with hard disks but also with solid state drives.

I. Introduction

Large-scale computing consumes large amounts of electrical power. This increasing financial and environmental barrier to scalable computing explains the trend toward "green" computing in recent years [?].

Google was one of the first major players to build their cluster architecture with a view to the price-performance ratio using cost-efficient servers [?]. Cost efficiency is calculated here over the lifetime of the server. This includes not only inexpensive machine components but also reduced maintenance and factors such as energy and cooling costs.

There are many ways to reduce power consumption. One is component consolidation. Some machines can be virtualized and unused servers within a cluster can be shut down and restarted as and when they are needed. According to one estimate, average storage utilization in data centers is 50%, so storage consolidation alone, e.g. to 75% of disk utilization, can result in 25% reduction of power and cooling requirements [?]. Another way to reduce power costs is to decrease cooling demands by exploiting cold natural environments. For example, Microsoft is reducing cooling costs by building a data center in Siberia. A third approach is to use energy-efficient components. For example, a low-power processor like Intel Atom provides more computation per watt than high-performance processors like Intel Xeon or AMD Opteron. Also, common storage devices can be replaced by more energy-efficient devices like flash storage. Falling prices for flash storage enables enterprises to replace conventional hard disk drives (HDD) by solid state drives (SSD) or even appliances like RAM-SAN [?]. When deployed properly, SSDs can provide higher performance [?], [?] but are still too expensive for some application scenarios in the the low-cost server market.

We analyzed the energy efficiency of flash storage devices that can be used instead of hard disk drives in commodity server hardware. In this paper, we consider whether, and if so when, it is reasonable to replace standard hard disk drives by low-cost USB flash drives. For this purpose, we conducted a series of experiments with different I/O patterns to identify performance per joule.

The rest of this paper is organized as follows. In section II, we review the latest work in the field of flash storage. In section III, we discuss some technical aspects of flash and disk storage that influence performance and energy consumption. In section IV, we evaluate some energy efficiency and performance aspects of these storage devices. Finally, in section V, we summarize our results.

II. Related Work

Because of its energy efficiency characteristics, flash deployment is currently under intensive analysis.

Modular web server systems like XAMPP [?] use USB flash drives mainly because of their portability, but this has the side effect of reducing power consumption. Project FAWN offers a new power-efficient alternative for data-intensive computing [?], [?]. Based on a large number of slow but efficient nodes that each draw only a few watts and use CompactFlash as storage, this system achieves very good performance in terms of queries per joule, even compared with SSDs. The best performance (factor 8) was achieved for seek-bound workloads, i.e. read-mostly workloads with random access patterns for small objects from a large corpus of data. These workloads are of growing importance in web applications. Other studies consider storing a large sample in a B-FILE structure designed to cope with flash constraints in order to reduce latency and energy consumption [?], [?].

Performance aspects of flash memory in different use cases are handled in [?], [?], [?], [?]. The read and write performance of SSDs, a CompactFlash drive, and hard disk drives are compared in [?]. Random writes are identified as the weak spot of flash memory, although flash memory outperforms hard disks for small contiguous reads (less than 512 KB).

Using benchmark `postmark`, Shin [?] compares file systems to find a suitable system for flash memory and finds that `Btrfs` is best for in sequential writes, followed by `XFS` and `ext2`. Some issues relevant to SSD performance, namely data placement, parallelism, write ordering, and workload management are examined in [?] in view of industrial trends.

Various aspects of power-aware cache management for hard disks are discussed in [?] with the aim of reducing writes by aggregating several operations. Performance analysis is measured in terms of queries per joule or I/O operations per watt [?], [?]. From this point of view, flash memory outperforms hard disks both for reads (about 250 times better) and writes. Gray [?] computed I/O operations per $ for flash storage and hard disks taking the power consumption and initial costs of the storage device into account. Compared to hard drives with 10 000 or 15 000 revolutions per minute (rpm), flash provides 10–14 times more I/O operations (IOPS) per $ for reads. However, for writes, disks provide 7–10 times more IOPS per $.

The cited references focus on analysis of flash memory on solid state drives. Commodity USB flash drives seem to be regarded as secondary, and the area of the energy efficiency is not examined [?]. Our paper closes this gap by comparing the conventional UBS flash drives with commodity hard disk drives and other flash devices in terms of performance and energy efficiency.

III. Technology Storage Aspects

A. Hard Disk Storage

Hard disks provide non-volatile memory and store data on rotating platters with magnetic surfaces. For each surface, a read/write head can be positioned on a particular track. Depending on the spindle speed (in rpm) of a disk, the performance increases and so does power consumption. The size and number of platters influence the energy required to spin them, as well as the overall capacity.

An important characteristic of a I/O subsystem's performance is the access time, i.e. the time required for a computer to retrieve required data from a storage device. Two major factors contribute to the access time of a hard disk drive:

- Average seek time: Time needed for the access arm to reach the specified track. The seek time of modern hard drives ranges from 5 to 15 ms.
- Average rotational latency time: Delay of the disk-rotation for positioning the read/write head over the track. This value depends solely on rotational speed. The hard disk drives with speed between 4 200 and 15 000 rpm have an average rotational latency between 2 and 7 ms, calculated as follows:

$$\text{Average rotational latency/ms} = \frac{30\,000}{\text{Spindle speed/rpm}}$$

B. Flash Storage

Flash storage, like main memory, has no moving parts and stores information as electric charge. But compared to RAM, flash does not need power to keep the information. Flash devices are primarily of two types: NOR and NAND. NOR devices have faster and simpler access procedures but lower storage capacity, which suits them for program storage. We do not discuss NOR architecture further. NAND flash offers significantly higher storage capacity and is more suitable for storing large amounts of data. There are two NAND flash realizations: single-level cell (SLC) and multi-level cell (MLC). SLC allows each memory cell to store only one bit of information whereas MLC allows storage of two or four bits per cell. SLC memory offers better performance and higher endurance but is more expensive than MLC. For this reason, only high-end SSDs are based on SLC whereas low-cost SSD and USB flash drives are based on MLC cells.

The key characteristics of NAND flash that directly influence an I/O subsystem's performance are discussed in detail in [?]. In summary, all read and write operations happen at page granularity, which, depending on the manufacturer, is typically 512–4096 byte. Several pages are combined to blocks typically of 128, 256, 256, or 512 KB. Each block is accessed by a single data line, thus finer-grained random access is impossible. Erase operations can be performed only block-wise and this is the basis for every data modification. A page can be be modified, i.e. written, only after erasing the entire block to which the page belongs. However, once a block is erased, all the pages in the block can be written at once into the erased cells. Page write cost (not taking block erase into account) is typically higher than read, and the block erase procedure makes some writes even more expensive. In particular, for an in-place update, before the erase and write can proceed, any useful data residing on other pages in the same block must be copied to a new block; this internal copying incurs a considerable overhead.

IV. EVALUATION

To measure the performance of the different storage subsystems and thus to evaluate the respective use cases, we conducted experiments on I/O performance, metadata performance, and energy efficiency. To check that the test results were consistent, we repeated each test three times. Results between different runs showed low variance.

A. Test Environment

Our test environment consisted of an off-the-shelf server with the following components: Main board with Intel P35 chip set, CPU Intel Core 2 Duo E6750 2.66 GHz FSB1333, 2 GB RAM DDR2 2048 MB Kit PC800 CL5, 380 W ATX power supply, and Linux (Ubuntu 8.04) operating system with Kernel 2.6.24. We selected file systems ext2, ext3, FXS, and VFAT as commonly used both on hard disk and on flash storage. To measure energy consumption of the entire server system we used the energy cost meter EKM 2000 from Olympia[1], which can record accumulated energy consumption over a long time period. This feature was used to record the total energy consumption of each experiment once the experiment finished.

Each tested storage component was used as primary built-in storage. The following devices were chosen as examples of commodity storage:

- Hard disk drive: Western Digital RE2 WD4000YR SATA (400 GB, 16 MB cache and 7200 rpm). A power-saving mode was not activated. Purchase cost was € 60 and cost per GB € 0.15.
- Solid state drive: Samsung MCBQE32G5MPP-03A PATA UDMA/66 (32 GB, SLC). Purchase cost was € 370 and cost per GB € 11.56.
- USB flash drive: Samsung K9HCG08U1M-PCB00 NAND flash modules (8 GB, block size 512 KB + 16 KB, MLC, write cycle of 25 ns). Purchase cost was € 8 per every drive and cost per GB € 1.
- CompactFlash drive: SanDisk 8 GB Extreme Ducati Edition (connected with a CF-to-IDE adapter). Purchase cost was € 90 and cost per GB € 11.25.

B. Performance Measurements

1) Read Access Time: Because there are no moving parts, flash memory has no seek time and a nearly uniform access time for each page. Depending on the model, access time for SSDs is between 0.1 and 0.5 ms and less than 2 ms for USB flash drives. Table I shows the read access times of the four tested storage components, measured using `h2benchw`[2]. The solid state drive has the lowest access time followed by the CompactFlash drive and the USB flash drive. The HDD access time is over ten times slower compared to the USB flash drive and over 65 times slower than the solid state drive. Because the seek time on flash does not depend on the physical location of data, its read performance is almost constant and deterministic across the entire storage. File fragmentation has no impact on read performance of flash memory.

2) Sequential Read and Write: We measured sequential read/write performance using the tool `dd`[3] with a block size of 4 KB. We measured the commonly used Linux file systems ext2, ext3, XFS, and VFAT, which are still the standard file systems for removable media like CompactFlash and USB flash drives. For comparison, we analyzed raw read/write performance of the devices. To improve the availability of USB flash storage, we also tested common RAID configurations.

As tables II and III show, the hard disk drive provides the best I/O performance, followed by the solid state drive, the CompactFlash drive, and the USB flash drive. A set of four equal USB flash drives connected in a RAID 0 configuration performs similarly to the solid state drive and about three times better than a single USB flash drive. During read, all file systems provide a similar performance to that of raw access. The usage of ext3 reduces the write performance for all tested storage devices. Since ext3 is a journaling file system and has to perform more metadata modifications, it performs worse for write access than ext2. Even if XFS has to handle a journal, it exhibits a good write performance, which can be explained by allocation groups [**?**] and delayed allocation. Given the features of XFS and the fact that it is a journaling file system, using XFS on USB flash drives seems appropriate. The usage of RAID 5 reduces the write performance to 5 MB/s, making it useless for applications. We think this is because the RAID 5 driver inside the Linux kernel is not optimized for flash usage.

3) Random Read and Write: To measure random read and write performance, we used the I/O benchmark IOzone[4]. Figure 2 shows that for block sizes smaller than 256 KB the random read of all flash drives outperforms HDD read. However, the random write performance (see figure 1) of flash drives is worse than that of hard disk drives. Among the flash drives, the solid state drive delivers the best performance, whereas the the USB flash drive is the slowest.

The three flash storage components show a clear decrease of random read performance for a block size of 256 KB. This is because of the read-ahead technique employed by the kernel to improve sequential read performance. The current Linux kernel uses a read-ahead window of 128 KB. If an application requires 8 KB to be read sequentially from the drive, 128 KB is cached from the disk and then 8 KB is returned to the application. The next request for 8 KB is accelerated because the requested data is already in cache. Generally, during random access, read-ahead is switched off. The Linux kernel's read-ahead is activated if more data is read randomly than the size of the the read-ahead window (which can be adapted). This reduces the performance in out random-I/O tests because too much data is read and resources are wasted. Figure 3 shows measurements with a read-ahead window of 4096 KB, which gives improved performance with a block size greater than 128 KB for all tested storage devices.

[1] `http://www.olympia-vertrieb.de/index.php?id=91&L=1`
[2] `ftp://ftp.heise.de/pub/ct/ctsi/h2benchw.zip`
[3] `http://www.softpanorama.org/Tools/dd.shtml`
[4] `http://www.iozone.org`

TABLE I
READ ACCESS TIME OF FLASH DEVICES AND HARD DISK DRIVE [MS]

Access Time	Hard Disk Drive	Solid State Drive	CompactFlash Drive	USB Flash Drive
Minimal	2.98	0.15	0.20	0.48
Average	13.02	0.20	0.64	1.28
Maximal	25.48	1.26	2.35	2.00

TABLE II
READ PERFORMANCE ON FLASH DEVICES AND HARD DISK DRIVE [MB/S]

File System	Hard Disk Drive	Solid State Drive	CF Drive	USB Flash Drive	2 USB Flash Drives (RAID 1)	4 USB Flash Drives (RAID 0)	4 USB Flash Drives (RAID 5)
Device	63.8	58.5	29.2	18.1	18.1	48.4	48.2
ext2	62.7	57.7	29.9	18.2	18.2	48.1	48.0
ext3	63.1	57.8	29.6	18.2	18.2	48.1	48.1
XFS	64.1	57.8	30.1	18.2	18.2	47.4	47.7
VFAT	31.8	57.3	29.1	17.2	21.8	46.7	46.2

TABLE III
WRITE PERFORMANCE ON FLASH DEVICES AND HARD DISK DRIVE [MB/S]

File System	Hard Disk Drive	Solid State Drive	CF Drive	USB Flash Drive	2 USB Flash Drives (RAID 1)	4 USB Flash Drives (RAID 0)	4 USB Flash Drives (RAID 5)
Device	64.3	39.0	30.4	16.1	13.8	46.1	4.5
ext2	61.0	31.4	30.2	11.4	9.7	42.5	3.3
ext3	58.9	25.4	25.1	3.7	3.8	32.8	3.3
XFS	65.8	36.2	25.4	14.4	12.8	41.8	2.5
VFAT	61.0	38.3	26.7	12.0	13.2	30.7	3.6

TABLE IV
METADATA PERFORMANCE ON FLASH DRIVES AND HARD DISK DRIVE [S]

File System	Hard Disk Drive	Solid State Drive	CF Drive	USB Flash Drive	2 USB Flash Drives (RAID 1)	4 USB Flash Drives (RAID 0)	4 USB Flash Drives (RAID 5)
ext2	9.0	39.2	145.5	160.3	163.9	196.8	230.0
ext3(1)	1.7	2.3	3.5	30.9	25.5	24.4	30.5
ext3(2)	16.2	61.2	88.6	714.0	840.0	594.8	1110.9
ext3(3)	1.8	2.1	3.5	29.6	25.3	24.3	26.1
XFS	303.7	76.0	105.3	820.3	1058.1	851.2	2004.3
VFAT	8.9	75.5	84.0	11109.5	aborted	aborted	aborted

(1)`data=ordered` (2)`data=journal` (3)`data=write back`

4) Metadata Performance: With the file system benchmark tool `fileop` we measured the time needed to complete the file operations for 27 000 files and empty files (see table IV). The measured file operations are `mkdir`, `rmdir`, `create`, `read`, `write`, `close`, `stat`, `access`, `chmod`, `readdir`, `link`, `unlink` and `delete`.

We measured three journaling methods in ext3: *ordered* (all data is forced directly to the file system prior to its metadata being committed to the journal), *journal* (all data is committed into the journal prior to being written into the file system), and *write back* (data may be written into the file system after its metadata has been committed to the journal).

The evaluation showed that the journaling method *write back* is the best for the USB flash drive. The method *journal* is inappropriate for flash memory. The best metadata performance was measured for ext2 and can be explained by the absence of a journal and therefore decreased metadata activity. The metadata performance of XFS is disappointing because of the intensive journaling activities. Evaluation of the metadata performance of VFAT was aborted due to the unacceptable test duration. An explanation for the poor metadata performance of flash memory overall is its lack of a cache buffer. Modern hard disk drives contain a cache buffer of 8–32 MB DRAM. Write access is collected inside the buffer and sorted, to gain a significant speed-up. Only a few SSDs, e.g. Intel X25-M SSD[5] or MemoRight GT MR25.2-064S SSD[6], contain a 16 MB cache buffer.

Our experiments show that hard disks deliver better performance in I/O-bound scenarios that involve either writes or I/O patterns with large contiguous access. We also saw that metadata modifications on file systems running on USB flash drives perform worse. By contrast, in read-mostly workloads, flash performance is almost the same as that of a hard drive. In scenarios where seek is often required, flash storage greatly outperforms HDD storage. All of the tested file systems were designed for hard disks. Flash storage may perform better if the file system is adapted for the technical properties of flash.

[5]`http://www.intel.com/design/flash/nand/`
[6]`http://www.memoright.com/en/index.asp`

Examples of such systems are JFFS2[7], YAFFS[8], or LogFS[9].

C. Performance per Joule

We devoted special attention to the energy efficiency (P) measured as performance per joule (1 J = 1 Ws), where performance is interpreted as the amount of data accessed per joule. Alternatively it represents the sustained throughput a device can deliver per watt. The main values for the calculation are sustained throughput (T) and power consumption during operation (E). The equation for calculationg efficiency P is specific to a storage device. The equation for hard drives is:

$$P = \frac{T}{E}$$

Compared to hard disk drives, there is a high variance of solid state drive read and write performance. Thus, for the SSDs we take the mean of read throughput (T_R) and write throughput (T_W) in the equation as T .

We can also consider the variance in power consumption for USB flash drives, because it is significantly different. For USB flash drives, the calculation of performance per joule involves the power consumption during read (E_R) and power consumption during write (E_W):

$$P = \frac{1}{2}\left(\frac{T_R}{E_R} + \frac{T_W}{E_W}\right)$$

The energy consumption of the storage interfaces are not considered in the equations, although we expect USB2 to consume more energy in the host than, say, SATA.

Tables V, VI, and VII show a selection of current HDDs, SSDs, and USB flash drives with their performance and energy characteristics. The specifications for the storage shown are taken from the manufacturers as the basis for the calculations of performance per joule in the above formulas. We consider these values as averages since the values in a specification are a compromise between conservative worst-case information and information attractive for the market. Note the efficiency of the power supply is not considered in the tables.

The results show that even if hard disks provide a better average I/O throughput than flash storage devices, the performance per joule of inexpensive commodity USB flash devices is much better. For instance, the tested hard disk drive WD4000YR provides a performance per joule of 6.1 MB/J compared with 57.6 MB/J for the tested Samsung flash drive.

D. Energy Costs

We calculate energy costs per year (C_Y) assuming energy cost per kWh is €0.18:

$$C_Y = E * 24 * 365 * 0.18 \; [\text{kW} * \frac{\text{hours}}{\text{day}} * \frac{\text{days}}{\text{year}} * \frac{€}{\text{kWh}}]$$

As table V shows, the idle power consumption of hard disks with a form factor of 3.5″ is 3.5–7 W, which under

[7] http://sourceware.org/jffs2/
[8] http://www.yaffs.net
[9] http://logfs.org

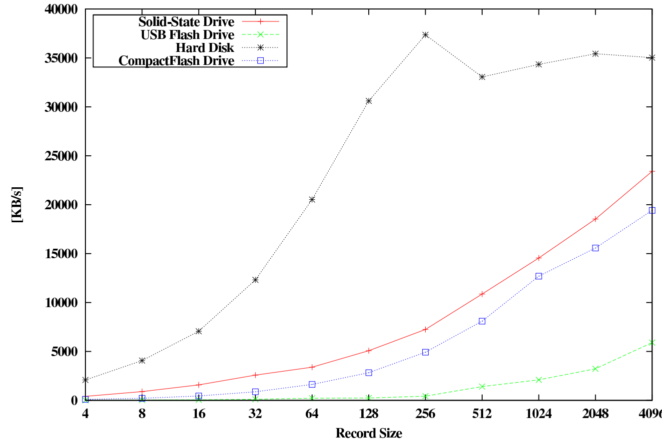

Fig. 1. Random write performance

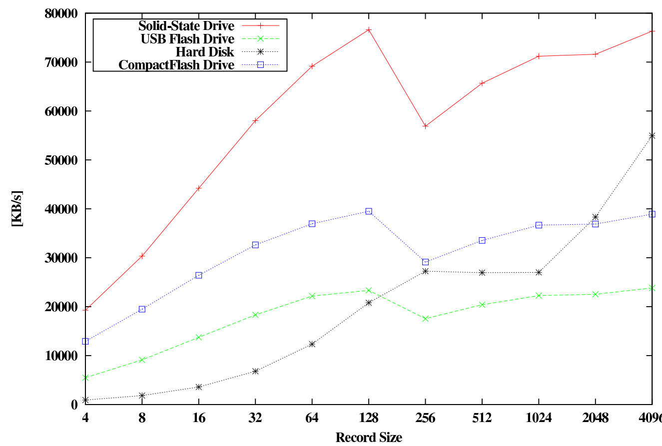

Fig. 2. Random read performance

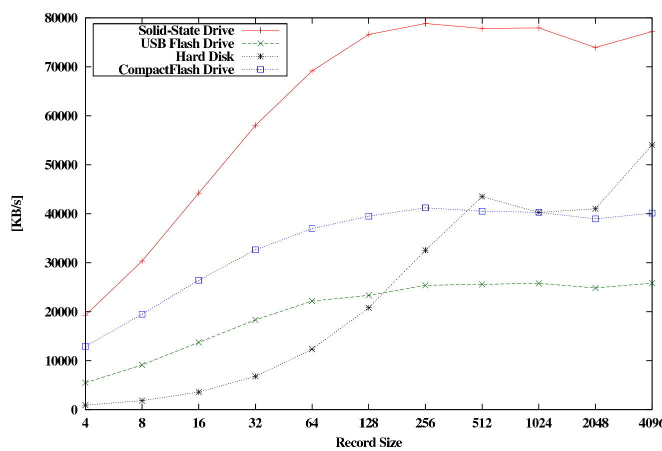

Fig. 3. Random read performance without read-ahead

load increases to 6–15 W, whereas hard disk drives with a form factor of 2.5″ have an idle power consumption of 0.5–1.5 W, which increases under load to 1.5–3 W. The tested drive WD4000YR has an approximate idle energy consumption of

TABLE V
SELECTION OF MODERN HARD DISK DRIVES

Model	RPM	Form Factor	Capacity [GB]	Throughput (sustained) [MB/s]	Power Consumption (transfer) [W]	(idle) [W]	Performance per Joule [MB/J]
Seagate ST3450856SS Cheetah 15K.6	15000	3.5”	450	140	17.3	12.4	8.1
Western Digital WD1001FALS	7200	3.5”	1000	80	8.4	7.8	9.5
Western Digital RE2 WD4000YR	7200	3.5”	400	65	10.8	8.9	6.1
Samsung HD103UI EcoGreen	5400	3.5”	1000	65	6.2	5.0	10.5
Seagate ST9250421AS Momentus	7200	2.5”	250	60	2.1	0.7	28.6
Hitachi Travelstar 5K500	5400	2.5”	500	50	1.9	0.7	26.3

TABLE VI
SELECTION OF MODERN SOLID STATE DRIVES

Model	Type	Form Factor	Capacity [GB]	Throughput (read) [MB/s]	(write) [MB/s]	Power Consumption (transfer) [W]	(idle) [W]	Performance per Joule [MB/J]
Samsung MCCOE64G5-MPP	SLC	2.5”	64	90	80	0.8	0.2	106.3
Mtron MSP-SAA7535032	SLC	2.5”	32	115	110	2.4	1.6	46.9
Crucial CT64GBFAA0	MLC	2.5”	32	125	55	2.1	1.6	42.9
Samsung MCBQE32G5-MPP	SLC	2.5”	32	55	40	0.2	0.1	237.5
Hama 00090853	SLC	2.5”	32	60	30	1.8	0.8	25.0

TABLE VII
SELECTION OF MODERN USB FLASH DRIVES

Model	Type	Capacity [GB]	Throughput (read) [MB/s]	(write) [MB/s]	Power Consumption (read) [mW]	(write) [mW]	(idle) [mW]	Performance per Joule [MB/J]
Samsung Flash Drive	MLC	8	18	16	215	375	200	57.6
SanDisk Cruzer Mini	MLC	1	13	8	130	150	80	75.0
Super Talent STU1GSMBL	MLC	1	14	5	70	80	55	126.7
CmMemory Core	MLC	1	12	9	65	75	50	150.0
SanDisk Cruzer Mini	MLC	0.5	16	5	130	130	80	80.8

TABLE VIII
HTTP_LOAD TEST ON USB FLASH DRIVES, SOLID STATE DRIVES AND HARD DISK DRIVES [S]

File System	Hard Disk Drive	Solid State Drive	USB Flash Drive	2 USB Flash Drives (RAID 1)	4 USB Flash Drives (RAID 0)	4 USB Flash Drives (RAID 5)
ext3	403.0	43.4	132.2	137.6	72.3	49.8
ext3 (log on tmpfs)	403.0	38.3	120.4	127.1	43.3	36.9

8.9 W and 10.75 W during read/write. In 24/7 usage of this disk, energy costs accumulate to a price between €14 and €17 per year.

Decreased power consumption of flash memory (see tables VI, VII) can be explained by absence of mechanical parts. The tested USB flash drive has an idle power consumption of 200 mW, which increases during read access to 215 mW and during write access to 375 mW. So the energy cost of a single USB flash drive idling for one year is €0.32. However, the energy cost for constant read or write access is €0.34 or €0.59 per year respectively. Therefore, the energy cost of using one USB flash drive 24/7 is between €0.32 and €0.59 per year. For the tested SSD, the energy cost in 24/7 usage is between €0.16 and €0.32 per year. Using SSD instead of the tested WD400DYR hard disk drive we reduce the storage subsystem energy consumption by a factor of about 65, and using one USB flash drive by a factor of about 34

As tables V, VI, and VII show, the performance per joule of flash drives is much better than that of hard disks. Since we always deal with the whole server, we measured the achieved performance per joule for the whole system under a sequential I/O-bound workload both with and without flash storage. The tested server was equipped with hard drive and ext3 and consumed 104 W during read (63 MB/s) and 97 W during write operations (59 MB/s). This means the performance per joule was 621 KB/J for sequential read and 622 KB/J for sequential write respectively. The energy consumption of the server with one and four USB flash drives for this activity was almost the same. For this reason, but also because this scenario is more realistic and provides better performance, we compare here the flash bundle of RAID 0. The server equipped with four USB drives and ext3 consumed 84 W both during sequential read (48 MB/s) and sequential write (33 MB/s). The measured performance per joule of this configuration was 586 KB/J respectively 400 KB/J.

In summary, even if the performance per joule of a single-component flash drive is much better than that of a hard disk, in the case of an entire system under a continuous sequential

TABLE IX
ENERGY METRICS FOR HTTP_LOAD TEST

Metric	Hard Disk Drive	Solid State Drive	USB Flash Drive	4 USB Flash Drives (RAID 0)
Power [W]	95	100	88	100
Request/Time [1/s]	248	2304	756	2310
Request/Energy [1/J]	2.7	23.3	8.6	23
Total Energy [kJ]	38.3	4.3	11.6	7.2
Energy Costs [cent]	0.180	0.022	0.054	0.036

I/O-bound workload, flash performs less well than a hard drive for writes and is similar for reads. If a server's I/O subsystem is used infrequently, flash storage provides more energy efficiency.

E. Appropriate scenario

The above results suggest that read-mostly and random-I/O workloads are appropriate usage scenarios for USB flash storage. For this reason a web server scenario is an appropriate one for flash deployment. Since a web server typically runs 24/7 with intermittent and limited I/O, it can exploit the power-saving potential of USB flash drives. The primary activity of a web server is often to read static content and deliver it to clients. Write access is only required for content updates, database updates, or access logging. In our evaluation, the Apache 2 web server hosted 241 498 HTML pages and images with an average size of 8.7 KB. The total volume of the content was 2.0 GB. In the test the server did not accessed a database.

With the benchmark tool `http_load`[10] we measured the time needed to fetch 100 000 files from the server using 40 concurrent requests. For this, a second machine was connected by Gigabit Ethernet. Normally, the Apache web server logs each HTTP request. However, in server farms the logfiles are often forwarded to specific nodes. To simulate this behavior and to reduce write access, a separate experiment was performed, where the logfile was placed on `tmpfs`. As table VIII shows, the write access of the logfile has an significant impact on flash performance, especially for the four USB flash drives in RAID 0. For the experiment with writing the logfile on the storage device, we also measured energy consumption during the test and analyzed multiple energy-related metrics (see table IX).

When equipped with just one USB flash drive, the web server consumes 88 W for 132 s to perform 100 000 fetches. Thus 756 requests per second and 8.6 requests per joule are achieved, which amounts to 11 634 J (about 0.003 kWh). Replacing the USB flash drive by the SSD, the web server needs 43 s for 100 000 fetches.

Using for reliability reasons four USB flash drives, the performance increased by 35% with RAID 5 and by 53% with RAID 0. Table VIII shows that with four USB flash drives in a RAID 0 configuration, the tool `http_load` needs 72.3 s for 100 000 fetches, and with RAID 5 it needs only 49.8 s.

The server with the installed hard disk drive consumed 95 W during this test. However, `http_load` needed 403 s for 100 000 fetches. Due to the longer runtime, only 248 requests per second and 2.7 requests per joule were achieved. Compared to the USB flash drives, the overall energy consumption of the hard disk setup is much higher, at 38.3 kJ (about 0.11 kWh).

Out tests demonstrated that the most energy-efficient solution is a solid state drive. An almost optimal solution is offered by sets of four USB flash drives. In this experiment, the hard disk drive consumes factor 5 more energy than the server with four USB flash drives.

The web server experiment showed that USB flash storage is appropriate for read-mostly and random-I/O workloads. We can conclude here that USB flash is suitable for applications with a similar I/O pattern requirement. To identify further scenarios for USB flash drive deployment, we need to recall the relatively high failure rate and limited capacity of flash drives compared with standard hard disk drives [**?**]. Storage system reliability and capacity can be increased by using RAID with flash sticks. Given these limitations, we regard CPU-bound applications in a cluster as another appropriate use case for flash storage. Flash can be used instead of internal hard disk drives for saving temporary data at computation nodes. Another scenario is storage of the operating system on flash sticks in a cluster. In these cases, the limited capacity, relatively poor reliability, and slow write performance of flash are acceptable.

V. CONCLUSION

Using flash storage instead of hard disk drives can decrease the energy consumption of a computer system. Even if hard disks deliver better performance in sequential I/O-bound applications, the performance per joule of commodity USB flash drives is better. Considering an entire system's energy cost in 24/7 usage, replacing the tested hard disk drive by a single USB drive reduces the energy consumption of the storage subsystem by factor of about 34.

If a server system is under sequential I/O workload, its overall performance per joule is worse with flash than with a hard drive for writes and is similar for reads. In idle mode, the system with flash storage offers more energy efficiency. Higher initial acquisition costs per gigabyte for flash storage in comparison to hard drives are balanced by lower energy costs over the long term.

The specific requirements of an application and the resulting I/O patterns play an essential role in deciding the most suitable storage solution. Our evaluation and the results of our

[10] `http://www.acme.com/software/http_load/`

web server tests demonstrate that read-mostly and random-I/O workloads are appropriate usage scenarios for USB flash storage.

Solid state drives are improving rapidly and becoming competitive with high performance hard disk drives. This year, Intel began to ship the X25-E solid state drive [**?**] with a read performance of 250 MB/s and a write performance of 170 MB/s. Further developments in flash memory will end the performance leadership of hard disks in the near future and will probably suit flash for all fields of application. In particular, inexpensive and energy-efficient USB flash drives are an option in the low-cost server area.

Acknowledgment

The authors would like to thank Maxym Gerashchenko for providing his energy cost meter and Andrew Ross for his help in improving numerous details in the paper.

References

[1] L. A. Barroso, "The price of performance," *ACM Queue*, vol. 3, no. 7, 4 2005.

[2] L. A. Barroso, J. Dean, and U. Hölzle, "Web search for a planet: The Google cluster architecture," *IEEE Micro*, vol. 23, no. 2, pp. 22–28, 2003.

[3] LSI. (2008) The performance impact of energy efficiency. [Online]. Available: http://www.lsi.com/powerconsumption/ESG_Power_Efficiency_WP_082807.pdf

[4] Texas Memory Systems. (2009) Product description of RamSan-400. [Online]. Available: http://www.ramsan.com/products/ramsan-400.htm

[5] A. Birrell, M. Isard, C. Thacker, and T. Wobber, "A design for high-performance flash disks," *SIGOPS Oper. Syst. Rev.*, vol. 41, no. 2, pp. 88–93, 2007.

[6] G. Mathur, P. Desnoyers, D. Ganesan, and P. Shenoy, "Capsule: an energy-optimized object storage system for memory-constrained sensor devices," in *SenSys '06: Proc. of the 4th international conference on Embedded networked sensor systems*, New York, USA, 2006, pp. 195–208.

[7] Apache Friends. (2009) XAMPP - an easy to install Apache distribution. [Online]. Available: http://www.apachefriends.org/en/xampp-windows.html

[8] V. Vasudevan, J. Franklin, D. Andersen, A. Phanishayee, L. Tan, M. Kaminsky, and I. Moraru, "FAWNdamentally power-efficient clusters," in *Proc. HotOS XII*, Monte Verita, Switzerland, 2009.

[9] D. G. Andersen, J. Franklin, M. Kaminsky, A. Phanishayee, L. Tan, and V. Vasudevan, "FAWN: A fast array of wimpy nodes," in *Proc. 22nd ACM Symposium on Operating Systems Principles (SOSP)*, Big Sky, USA, 2009.

[10] S. Nath and P. B. Gibbons, "Online maintenance of very large random samples on flash storage," *Proc. VLDB Endow.*, vol. 1, no. 1, pp. 970–983, 2008.

[11] S. Nath and A. Kansal, "FlashDB: dynamic self-tuning database for NAND flash," in *IPSN '07: Proc. of the 6th international conference on Information processing in sensor networks*, New York, USA, 2007, pp. 410–419.

[12] D. Shin, "About SSD," in *LSF'08: Linux Storage & Filesystem Workshop*, 2008.

[13] D. Dumitru. (2007) Understanding flash SSD performance. [Online]. Available: http://www.storagesearch.com/easyco-flashperformance-art.pdf

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

[15] N. Agrawal, V. Prabhakaran, T. Wobber, J. D. Davis, M. Manasse, and R. Panigrahy, "Design tradeoffs for SSD performance," in *USENIX Annual Technical Conference*, Boston, USA, 2008, pp. 57–70.

[16] Q. Zhu, F. M. David, C. F. Devaraj, Z. Li, Y. Zhou, and P. Cao, "Reducing energy consumption of disk storage using power-aware cache management," in *HPCA '04: Proc. of the 10th International Symposium on High Performance Computer Architecture*, Washington, USA, 2004, p. 118.

[17] D. Zeinalipour-Yazti, S. Lin, V. Kalogeraki, D. Gunopulos, and W. A. Najjar, "Microhash: An efficient index structure for flash-based sensor devices," in *FAST*, San Francisco, USA, 2005, pp. 31–44.

[18] D. Robbins, "Advanced filesystem implementor's guide, part 9," IBM developerWorks, 2002.

[19] J. A. Chandy and S. Narayan, "Reliability tradeoffs in personal storage systems," *SIGOPS Oper. Syst. Rev.*, vol. 41, no. 1, pp. 37–41, 2007.

[20] Intel. (2009) Product description of Intel X25-E extreme SATA solid-state drive. [Online]. Available: http://www.intel.com/design/flash/nand/extreme/index.htm