

Prolegomena

Department Editors: Kevin Rudd and Kevin Skadron



Architecting Storage for the Cloud Computing Era

SUDHANVA GURUMURTHI
University of Virginia

..... We're at an interesting crossroads in computer architecture. The past two decades have seen advances in processor design—from microarchitectural techniques to boost instruction- and thread-level parallelism to the shift from single-core to multicore processors. The number of cores on the die will likely continue to grow, thereby providing a tremendous amount of processing power within a single chip.

At the same time, applications are becoming more data intensive. Already there exist several applications that can handle massive amounts of data and are used by millions of people every day. These include traditional data-intensive applications, such as transaction processing, email, and search engines, as well as newer applications spanning areas such as social networking, photo and video sharing, and blogging. According to an International Data Corporation 2008 report, the amount of data generated and stored

worldwide is expected to reach 1,800 exabytes (18×10^{20} bytes) by 2011.¹ What is especially interesting about this trend is that although individual users will generate a significant fraction of this data, most of it will likely be stored and managed within data centers. This gives rise to new *cloud-based* computing and storage models in which user data and even the applications that use the data are hosted at data centers in the “cloud” and users access the resources over a network.

Supporting this confluence of trends in architecture and applications requires new techniques and mechanisms for storing large amounts of data and providing efficient access to them, and efficiently transporting the data between the cores and storage.

Meeting these requirements is challenging. Data is typically stored and accessed from disk drives. Although disks provide high density and have the

lowest cost-per-gigabyte of all storage media, they are slow and power hungry because they are electro-mechanical devices. (A single enterprise disk drive consumes 10 to 15 watts of power.) The use of multicore processors further exacerbates the performance gap between the CPU and disk, as Figure 1 illustrates. This graph shows the normalized CPU performance for various processors for 20-Kbyte random reads on IDE and SATA disks over time. The performance gap between the CPU and disk, which is already significant, becomes much wider for multicore processors, and this divergence gets worse over time as more cores are added to the die. Given that applications are becoming more data intensive, I/O will be a significant component; therefore, we must narrow this gap as much as possible.

The typical approach to building high-performance, high-capacity, and reliable storage systems is to use multiple disk drives to create storage arrays. The multiple disks provide for parallel I/O and distribute (and possibly mirror) data across the disks along with error detection/correction bits to provide fault tolerance. Given the rate at which new data is generated in these applications and their I/O intensive nature, storage systems that house the data will tend to be large and consume a significant amount of power, driving up electricity and cooling costs within the data center.

Editors' Note

Rapid changes in hardware technology and system requirements are opening up a variety of exciting lines of research that could dramatically change the design of future computer systems. This article presents the first in a periodic series of *prolegomena*—brief introductions to recent and exciting research topics, in which the main challenges and vision for future research are outlined. The goal of this feature is to facilitate discussion and collaboration. In this first prolegomenon, Sudhanva Gurumurthi, an assistant professor of computer science at the University of Virginia, discusses the storage challenges posed by datacenter-scale workloads.

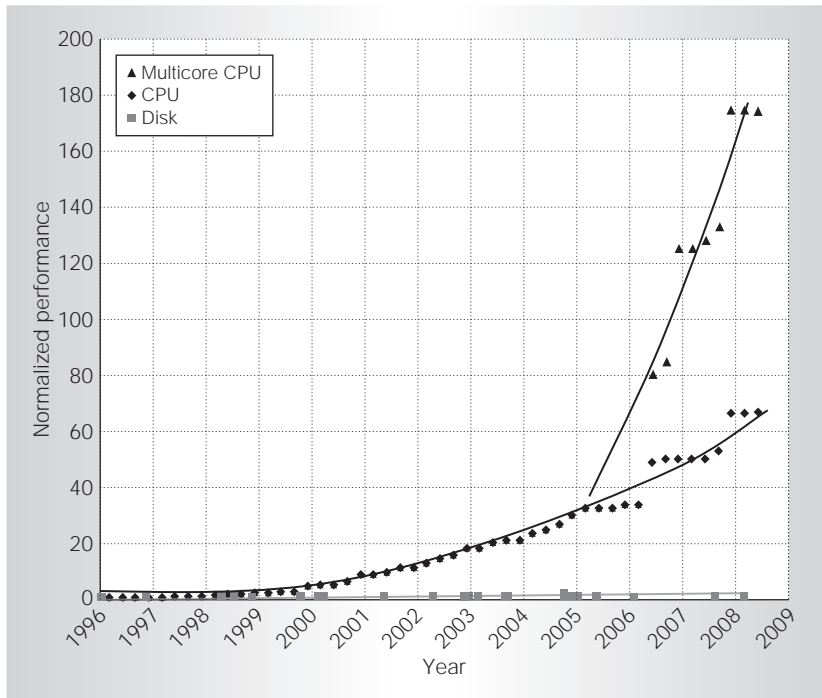


Figure 1. Normalized CPU performance of single-core and multicore processors for 20-Kbyte random read operations.² (Source: Intel Measurements; Copyright 2009 Intel Corporation)

Two broad and crosscutting research efforts in the computer architecture and systems communities aim to address these challenges. Although researchers have made many contributions in both areas, space considerations limit the discussion here to selected projects.

Adopt the use of solid-state disks

A disk's performance depends on various factors, including the speed at which the spindle rotates (RPM), which affects the rotational latency and data transfer speed for I/O requests; the time taken to move the disk arms, which affects seek time; and the size of the on-board disk cache, which affects the hit rate and hence a request's overall latency. The spindle, which consists of a motor and platter stack, tends to be the highest power consumer within the drive. Modern disks rotate at a high RPM (10,000 to 15,000 revolutions per minute), and physical constraints such as heat

dissipation make it difficult to keep increasing the RPM to boost performance.³ Moreover, because the spindle keeps rotating even when the disk isn't servicing requests, a disk's idle power tends to be high. Finally, the mechanical positioning overheads within a disk can significantly degrade performance.

One way to overcome these difficulties is to move from hard disk drives (HDDs) toward disks that use some form of solid-state, nonvolatile memory, the most popular of which is NAND flash. Flash is a form of electrically erasable programmable read-only memory (EEPROM) that uses floating-gate transistors to store data. Flash memory can be designed so that a cell stores either a single bit (known as a single-level cell, or SLC) or multiple bits of data (known as multilevel cell, or MLC). Flash memory's cost-per-gigabyte has dropped significantly over the past few years, making it economically viable for use in mass storage devices.

NAND flash provides much faster access times for reads and writes than HDDs (microseconds compared to milliseconds), especially for random I/O. However, NAND flash has its own set of idiosyncrasies. NAND flash supports memory reads and writes at a *page* granularity, where a page is typically 2-4 Kbytes; but performs an erase at a coarser granularity of a *block*, where a block consists of multiple pages. There is asymmetry in read and write latencies, with the latter being slower. The write latency is typically higher for MLC NAND flash than for its SLC counterpart. In addition, flash memory doesn't support in-place writes. Writing a page requires erasing the block to which the data is to be written before the actual write can be performed, and the erase latency is around 1 or 2 milliseconds. Finally, flash memory has finite write endurance and can wear out after 10,000 to 100,000 write-erase cycles, with MLC flash being on the lower end of the endurance spectrum. A flash translation layer (FTL) in the solid-state drive (SSD) handles most of these idiosyncrasies. The FTL consists of algorithms for efficiently mapping the software-visible logical disk blocks to the physical addresses on flash and also performs wear-leveling and cleaning operations to handle endurance and capacity issues.⁴

Researchers have been exploring various aspects of SSD architecture, including the interconnection between the flash chips and the controller and techniques for managing the internal data movement,^{5,6} and FTL optimizations.⁷ They are even exploring new server architectures that integrate processors and flash much more tightly to boost performance and energy efficiency.^{8,9} A key challenge in flash memory research at the architecture level is the lack of sufficient understanding about how real flash chips behave with regard to performance, power, and reliability. Researchers have resorted to using data sheets, which don't provide adequate information for

PROLEGOMENA

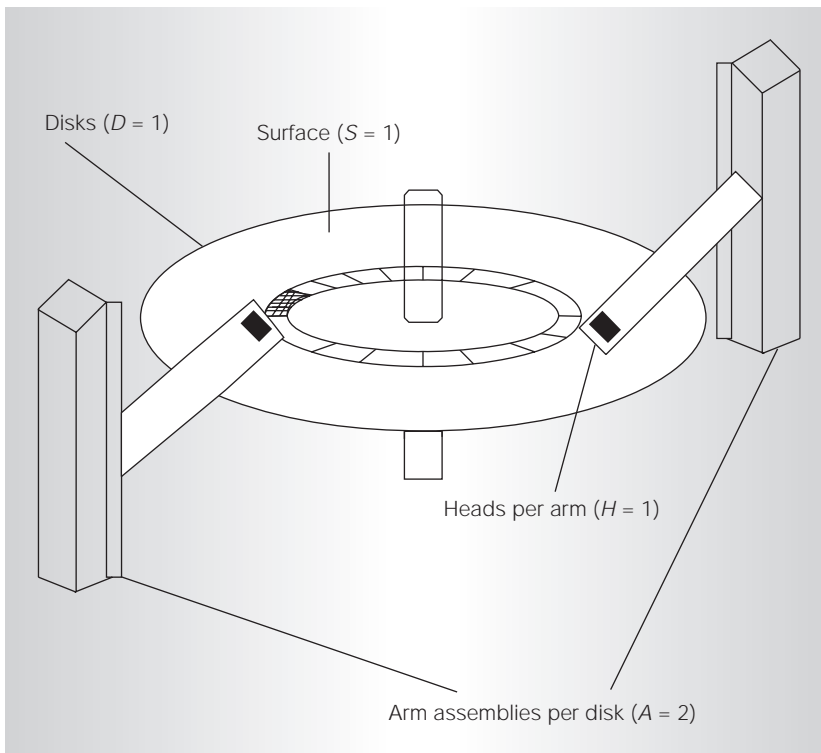


Figure 2. A multiactuator disk drive. The two sets of arms can be used by the disk to exploit parallelism in the I/O request stream.

performing in-depth analyses of various aspects of flash. Recent research has addressed this void through empirical measurement studies of real flash chips.¹⁰

Although flash is currently the solid-state memory used in SSDs, other nonvolatile memory technologies such as phase change memory (PCM) and spin-torque transfer RAM (STTRAM) are also suitable candidates. These technologies offer latency and endurance characteristics that are superior to flash, and they facilitate data accesses at a much finer granularity. However, they too have certain idiosyncrasies. Researchers have already begun exploring the use of these memory technologies in main memory and caches to complement or supplement DRAM and SRAM.^{11,12} In the future, they could potentially be leveraged for building SSDs.

Finally, the presence of a large amount of nonvolatile memory that could serve as both memory and storage

raises interesting questions about the memory hierarchy's design as well as that of the operating system, which is based on historical differences between volatile and expensive (in terms of cost-per-bit) main memory and a nonvolatile and relatively inexpensive disk.

Hard disk drives won't go away, they'll just vanish into the cloud

Although SSDs offer compelling performance and power benefits, they can't replace HDDs in a data center. Although flash's cost-per-gigabyte has been dropping, it's still an order of magnitude more expensive than HDDs. Therefore, HDDs will still be used to meet capacity demands, and data centers will use large disk-based storage systems for the foreseeable future. Hybrid drives that contain both an HDD and solid-state memory can provide SSDs' performance benefits and HDDs' capacity benefits. Alternately, instead of

using hybrid drives, you can build a tiered storage architecture consisting of SSDs as the storage system's front end to provide high performance and HDDs as the back end to provide the requisite capacity. A key issue in designing such tiered systems is determining the optimal system configuration to meet performance, capacity, and fault-tolerance requirements.¹³ Developing policies for migrating data between tiers at runtime based on workload behavior is another important issue.

Energy consumption is another critical problem when using large HDD-based storage systems. As mentioned earlier, disks consume a significant amount of power even when idle. Moreover, it isn't easy to power down the disks in a data center due to the relatively low durations of the idle periods compared to the time it takes to spin a disk down and back up again. Therefore, disks should provide *energy-proportional* behavior¹⁴—that is, they should consume very little power when idle, and the increase in power consumption should be in proportion to the I/O load. Architectural techniques that provide energy-proportional behavior for HDDs include multi-RPM operating modes¹⁵ and multiactuator disks.¹⁶ Multi-RPM drives allow trading off rotational latency and transfer speed for energy consumption. Multiactuator disks provide high performance via intradisk parallelism instead of higher rotational speeds. Figure 2 shows an example of a multiactuator disk. These HDD architectures, used in conjunction with system-level power-management policies,^{17,18} can significantly reduce energy consumption while providing acceptable performance.

Designing storage architectures for emerging data-intensive applications presents several challenges and opportunities. Tackling these problems requires a combination of architectural optimizations to the storage devices and layers of the memory/storage hierarchy as well as hardware/software

techniques to manage the flow of data between the cores and storage. As we move deeper into an era in which data is a first-class citizen in architecture design, optimizing the storage architecture will become more important. This article presents a very brief introduction to this vast and exciting research area.

References

1. J.F. Gantz et al., "The Diverse and Exploding Digital Universe: An Updated Forecast of Worldwide Information Growth Through 2011," IDC whitepaper, Int'l Data Corp., Mar. 2008.
2. F. Hady, "Integrating NAND Flash into the Storage Hierarchy: Research or Product Design?" Workshop on Integrating Solid-State Memory into the Storage Hierarchy (WISH), invited talk, 2009; http://csl.cse.psu.edu/wish2009_hady.pdf.
3. S. Gurumurthi, A. Sivasubramaniam, and V. Natarajan, "Disk Drive Roadmap from the Thermal Perspective: A Case for Dynamic Thermal Management," *Proc. Int'l Symp. Computer Architecture (ISCA 05)*, IEEE CS Press, 2005, pp. 38-49.
4. E. Gal and S. Toledo, "Algorithms and Data Structures for Flash Memories," *ACM Computing Surveys*, vol. 37, no. 2, June 2005, pp. 138-163.
5. N. Agrawal et al., "Design Tradeoffs for SSD Performance," *Proc. Usenix Ann. Technical Conf.*, Usenix Assoc., 2008, pp. 57-70.
6. C. Dirik and B. Jacob, "The Performance of PC Solid-State Disks (SSDs) as a Function of Bandwidth, Concurrency, Device Architecture, and System Organization," *Proc. Int'l Symp. Computer Architecture (ISCA 09)*, IEEE CS Press, 2009, pp. 279-289.
7. A. Gupta, Y. Kim, and B. Urganekar, "DFTL: A Flash Translation Layer Employing Demand-Based Selective Caching of Page-Level Address Mappings," *Proc. Int'l Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS 09)*, ACM Press, 2009, pp. 229-240.

8. D.G. Andersen et al., "FAWN: A Fast Array of Wimpy Nodes," *Proc. Symp. Operating Systems Principles (SOSP 09)*, ACM Press, 2009, to appear.
9. A.M. Caulfield, L. Grupp, and S. Swanson, "Gordon: Using Flash Memory to Build Fast, Power-Efficient Clusters for Data-Intensive Applications," *Proc. Int'l Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS 09)*, ACM Press, 2009, pp. 217-228.
10. L. Grupp et al., "Characterizing Flash Memory: Anomalies, Observations, and Applications," *Proc. Int'l Symp. Microarchitecture (MICRO 09)*, IEEE CS Press, 2009, to appear.
11. B.C. Lee et al., "Architecting Phase Change Memory as a Scalable DRAM Alternative," *Proc. Int'l Symp. Computer Architecture (ISCA 09)*, IEEE Press, 2009, pp. 2-13.
12. M.K. Qureshi, V. Srinivasan, and J.A. Rivers, "Scalable High Performance Main Memory System Using Phase-Change Memory Technology," *Proc. Int'l Symp. Computer Architecture (ISCA 09)*, IEEE CS Press, 2009, pp. 24-33.
13. D. Narayanan et al., "Migrating Server Storage to SSDs: Analysis of Tradeoffs," *Proc. European Conf. Computer Systems (EUROSYS 09)*, ACM Press, 2009, pp. 145-158.
14. L.A. Barroso and U. Hözlze, "The Case for Energy-Proportional Com-

- puting," *Computer*, vol. 40, no. 12, Dec. 2007, pp. 33-37.
15. S. Gurumurthi et al., "DRPM: Dynamic Speed Control for Power Management in Server Class Disks," *Proc. Int'l Symp. Computer Architecture (ISCA 03)*, IEEE CS Press, 2003, pp. 169-179.
16. S. Sankar, S. Gurumurthi, and M.R. Stan, "Intra-Disk Parallelism: An Idea Whose Time Has Come," *Proc. Int'l Symp. Computer Architecture (ISCA 08)*, IEEE CS Press, 2008, pp. 303-314.
17. S. Sankar, S. Gurumurthi, and M.R. Stan, "Sensitivity Based Power Management of Enterprise Storage Systems," *Proc. Int'l Symp. Modeling, Analysis, and Simulation of Computer and Telecomm. Systems (MASCOTS 08)*, IEEE Press, 2008, pp. 1-10.
18. Q. Zhu et al., "Hibernator: Helping Disk Arrays Sleep through the Winter," *Proc. Symp. Operating Systems Principles (SOSP 05)*, ACM Press, 2005, pp. 177-190.

Sudhanva Gurumurthi is an assistant professor of computer science at the University of Virginia. His research interests are in computer architecture, in particular energy-efficient storage systems and silicon reliability. Gurumurthi has a PhD in computer science and engineering from Pennsylvania State University. He is a recipient of the NSF CAREER Award. Contact him at gurumurthi@cs.virginia.edu.

IEEE Intelligent Systems

THE #1 ARTIFICIAL INTELLIGENCE MAGAZINE!

IEEE Intelligent Systems delivers the latest peer-reviewed research on all aspects of artificial intelligence, focusing on practical, fielded applications. Contributors include leading experts in

- Intelligent Agents
- The Semantic Web
- Natural Language Processing
- Robotics
- Machine Learning

Visit us on the Web at www.computer.org/intelligent