From Microprocessors to Nanostores: Rethinking Data-Centric Systems

Parthasarathy Ranganathan, *HP Labs*

The confluence of emerging technologies and new data-centric workloads offers a unique opportunity to rethink traditional system architectures and memory hierarchies in future designs.

What will future computing systems look like?

We are entering an exciting era for systems design. Historically, the first computer to achieve terascale computing (10^{12}, or one trillion operations per second) was demonstrated in the late 1990s. In the 2000s, the first petascale computer was demonstrated with a thousand-times better performance. Extrapolating these trends, we can expect the first exascale computer (with one million trillion operations per second) to appear around the end of this next decade.

In addition to continued advances in performance, we are also seeing tremendous improvements in power, sustainability, manageability, reliability, and scalability. Power management, in particular, is now a first-class design consideration. Recently, system designs have gone beyond optimizing operational energy consumption to examining the total life-cycle energy consumption of systems for improved environmental sustainability. Similarly, in addition to introducing an exciting new model for delivering computing, the emergence of cloud computing has enabled significant advances in scalability as well as innovations in the software stack.

Looking further out, emerging technologies such as photonics, nonvolatile memory, 3D stacking, and new data-centric workloads offer compelling new opportunities. The confluence of these trends motivates a rethinking of the basic systems' building blocks of the future and a likely new design approach called *nanostores* that focus on data-centric workloads and hardware-software codesign for upcoming technologies.

THE DATA EXPLOSION

The amount of data being created is exploding, growing significantly faster than Moore's law. For example, the amount of online data indexed by Google is estimated to have increased from 5 exabytes (one exabyte = 1 million trillion bytes) in 2002 to 280 exabytes in 2009[1]—a 56-fold increase in seven years. In contrast, an equivalent Moore's law growth in computing for the corresponding time would deliver only a 16-fold increase.

This data growth is not limited to the Internet alone, but is pervasive across all markets. In the enterprise space, the size of the largest data warehouse has been increasing at a cumulative annual growth rate of 173 percent[2]—again, significantly more than Moore's law.

New kinds of data

Some common categories for data growth include those pertaining to bringing traditionally offline data online and

0018-9162/11/$26.00 © 2011 IEEE Published by the IEEE Computer Society

to new digital media creation, including webpages, personal images, scanned records, audio files, government databases, digitized movies, personal videos, satellite images, scientific databases, census data, and scanned books. A recent estimate indicates that 24 hours of video are uploaded on YouTube every minute. At HD rates of 2-5 Mbps, that is close to 45-75 terabytes of data per day. Given that only about 5 percent of the world's data is currently digitized,[3] growth in this data category is likely to continue for several more years.

More recently, large-scale sensor deployment has contributed to the explosion in data growth. Developments in nanoscale sensors have enabled tracking multiple dimensions—including vibration, tilt, rotation, airflow, light, temperature, chemical signals, humidity, pressure, and location—to collect real-time data sampled at very fine granularities. These advances have motivated researchers to discuss the notion of developing a "central nervous system for the earth (CeNSE)"[4] with intriguing sample applications of rich sensor networks in areas including retail sales, defense, traffic, seismic and oil explorations, weather and climate modeling, and wildlife tracking. This vision will lead to data creation and analysis significantly beyond anything we have seen so far.

Looking ahead, it's clear that we're only at the beginning of an even more fundamental shift in what we do with data.

The pervasive use of mobile devices by a large part of the world's population, and the ability to gather and disseminate information through these devices, contributes to additional real-time rich data creation. For example, at the time of Michael Jackson's death in June 2009, Twitter estimated about 5,000 tweets per minute, and AT&T estimated about 65,000 texts per second. Currently, over a 90-day period, 20 percent of Internet search queries are typically "new data."[1]

Significantly, this large-scale growth in data is happening in combination with a rich diversity in the type of data being created. In addition to the diversity in media types—text, audio, video, images, and so on—there is also significant diversity in how the data is organized: structured (accessible through databases), unstructured (accessed as a collection of files), or semistructured (for example, XML or e-mail).

New kinds of data processing

This growth in data is leading to a corresponding growth in data-centric applications that operate in diverse ways: capturing, classifying, analyzing, processing, archiving, and so on. Examples include Web search, recommendation systems, decision support, online gaming, sorting, compression, sensor networks, ad hoc queries, cubing, media transcoding and streaming, photo processing, social network analysis, personalization, summarization, index building, song recognition, aggregation, Web mashups, data mining, and encryption. Figure 1 presents a taxonomy of data-centric workloads that summarizes this space.

Compared to traditional enterprise workloads such as online transaction processing and Web services, emerging data-centric workloads change many assumptions about system design. These workloads typically operate at larger scale (hundreds of thousands of servers) and on more diverse data (structured, unstructured, rich media) with I/O-intensive, often random, data access patterns and limited locality. In addition, these workloads are characterized by innovations in the software stack targeted at increased scalability and commodity hardware such as Google's MapReduce and BigTable.

Looking ahead, it's clear that we're only at the beginning of an even more fundamental shift in what we do with data. As an illustrative example, consider what happens when we search for an address on the Web.

In the past, this request would be sent to a back-end webserver that would respond with the image of a map showing the address's location. However, in recent years, more sophisticated data analysis has been added to the response to this query. For example, along with just accessing the map database, the query could potentially access *multiple data sources*—for example, satellite imagery, prior images from other users, webpages associated with location information, overlays of transit maps, and so on. Beyond just static images, *dynamic* data sources can be brought into play—such as providing live traffic or real-time weather information, current Twitter feeds, or live news or video. *Synthetic* data such as images from user-provided 3D models of buildings or outputs from trend analyzers and visualizers also can be superimposed on the map.

Adding personalization and contextual responses to the mix introduces another layer of data processing complexity. For example, different data can be presented to the user based on the last two searches prior to this search, or on the user's prior behavior when doing the same search, or on locational information (for example, if the current location matches the location where the user searched previously).

Social networks and recommendation systems add yet another layer of data processing complexity. Examples include on-map visualization of individuals' locations drawn from social networks, inferred preferences, and prescriptive recommendations based on social trends. Advertisements and, more generally, business monetization

of search, adds another layer of data processing in terms of accessing more data sources and more sophisticated algorithms for user preference and content relevance.

In many cases, all this data processing comes with fairly small latency requirements for response, even requiring real-time responses in some scenarios.

This scenario shows how, from simple Web search and content serving, online data processing is evolving to allow more complex meaning extraction across multiple data repositories, and more sophisticated cross-correlations, including more complex I/O movement. In a continuum of data processing operations including upload/ingress; download/egress; search (tree traversal); read, modify, write; pattern matching; aggregation; correlation/join; index building; cubing; classification; prediction; and social network analysis, recent trends show a strong movement toward operations with more complex data movement patterns.[5]

Response time	Real-time	Real-time or interactive responses required
	Background	Response time is not critical for user needs
Access pattern	Random	Unpredictable access to regions of datastore
	Sequential	Sequential access of data chunks
	Permutation	Data is redistributed across the system
Working set	All	The entire dataset is accessed
	Partial	Only a subset of data is accessed
Data type	Structured	Metadata/schema/type are used for data records
	Unstructured	No explicit data structure, for example, text/binary files
	Rich media	Audio/video and image data with inherent structures and specific processing algorithms
Read vs. write	Read heavy	Data reads are significant for processing
	Write heavy	Data writes are significant for processing
Processing complexity	High	Complex processing of data is required per data item; examples: video transcoding, classification, prediction
	Medium	Simpler processing is required per data item; examples: pattern matching, search, encryption
	Low	Dominated by data access with low compute ratio; examples: sort, upload, download, filtering, and aggregation

Figure 1. Data-centric workload taxonomy.

Similar trends can be seen in enterprise data management across the information→insight→outcome life cycle. There is an increasing emphasis on real-time feeds of business information, often across multiple formal or ad hoc data repositories, reduced latencies between events and decisions, and sophisticated combinations of parallel analytics, business intelligence, and search and extraction operations. Jim Gray alluded to similar trends in scientific computing when discussing a new era in which scientific phenomena are understood through large-scale data analysis.[6] Such trends can also be seen in other important workloads of the future, with applications like computational journalism, urban planning, natural-language processing, smart grids, crowdsourcing, and defense applications. The common traits in all these future workloads are an emphasis on complex cross-correlations across multiple data repositories and new data analysis/compute assumptions.

Together, this growing complexity and dynamism in extraction of meaning from data, combined with the large-scale diversity in the amount of data generated, represent an interesting inflection point in the future data-centric era. The "Implications of Data-Centric Workloads for System Architectures" sidebar provides additional information about this trend for system designs.

IT'S A NEW WORLD—AN INFLECTION POINT IN TECHNOLOGY

Concurrently, recent trends point to several potential technology disruptions on the horizon.

On the compute side, recent microprocessors have favored multicore designs emphasizing multiple simpler cores for greater throughput. This is well matched with the large-scale distributed parallelism in data-centric workloads. Operating cores at near-threshold voltage has been shown to significantly improve energy efficiency.[7] Similarly, recent advances in networking show a strong growth in bandwidth for communication between different compute elements at various system design levels.

However, the most important technology changes pertinent to data-centric computing relate to the advances in and adoption of nonvolatile memory. Flash memories have been widely adopted in popular consumer systems—for example, Apple's iPhone—and are gaining adoption in the enterprise market—for example, Fusion-io.

Figure 2 shows the trends in costs for these technologies relative to traditional hard disks and DRAM memories. Emerging nonvolatile memories have been demonstrated to have properties superior to flash memories, most notably phase-change memory (PCM)[8] and, more recently, memristors.[9] Trends suggest that future nonvolatile

IMPLICATIONS OF DATA-CENTRIC WORKLOADS FOR SYSTEM ARCHITECTURES

An important trend in the emergence of data-centric workloads has been the introduction of complex analysis at immense scale, closely coupled with the growth of large-scale Internet Web services. Traditional data-centric workloads like Web serving and online transaction processing are being superseded by workloads like real-time multimedia streaming and conversion; history-based recommendation systems; searches of text, images, and even videos; and deep analysis of unstructured data—for example, Google Squared.

From a system architecture viewpoint, a common characteristic of these workloads is their general implementation on highly distributed systems, and that they adopt approaches that scale by partitioning data across individual nodes. Both the total amount of data involved in a single task and the number of distributed compute nodes required to process the data reflect their large scale. Additionally, these workloads are I/O intensive, often with random access patterns to small-size objects over large datasets.

Many of these applications operate on larger fractions of data in memory. According to a recent report, the amount of DRAM used in Facebook for nonimage data is approximately 75 percent of the total data size.[1] While this trend partly reflects the low latency requirements and the limited locality due to complex linkages between data for the Facebook workload, similar trends for larger memory capacities can be seen for memcached servers and TPC-H benchmark winners over the past decade. Similarly, search algorithms such as the one from Google have evolved to store their search indices entirely in DRAM. These trends motivate a rethinking of the balance between memory and disk-based storage in traditional designs.

Interestingly, datasets and the need to operate on larger fractions of the data in-memory continue to increase, there will likely be an inflection point at which conventional system architectures based on faster and more powerful processors and ever deeper memory hierarchies are not likely to work from an energy perspective (Figure A). Indeed, a recent exascale report identifies the amount of energy consumed in transporting data across different levels as a key limiting factor.[2] Complex power-hungry processors also are sometimes a mismatch with data-intensive workloads, leading to further energy inefficiencies.

Recent data-centric workloads have been characterized by numerous commercially deployed innovations in the software stack—for example, Google's BigTable and MapReduce, Amazon's Dynamo, Yahoo's PNUTS, Microsoft's Dryad, Facebook's Memcached, and LinkedIn's Voldemort. Indeed, according to a recent presentation, the software stack behind the very successful Google search engine was significantly rearchitected four times in the past seven years to achieve better performance at increased scale.[3]

The growing importance of this class of workloads, their focus on large-scale distributed systems with ever-increasing memory use, the potential inadequacy of existing architectural approaches, and the relative openness to software-level innovations in the emerging workloads offer an opportunity for a corresponding clean-slate architecture design targeted at data-centric computing.

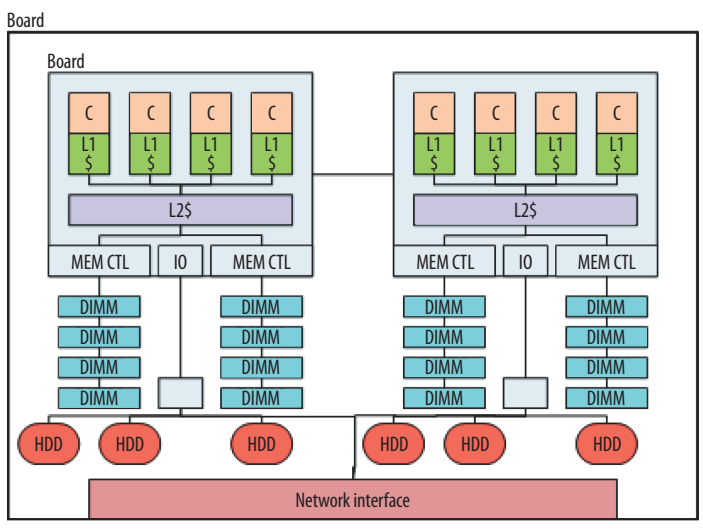

Figure A. Changing workload trends motivate a rethinking of the traditional designs with deep hierarchies.

References

1. J. Ousterhout et al., "The Case for RAM Clouds: Scalable High-Performance Storage Entirely in DRAM," *ACM SIGOPS Operating Systems Rev.*, vol. 43, no. 4, 2009, pp 92-105.
2. P. Kogge ed., "ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems," 2008; http://www.science.doe.gov/ascr/Research/CS/DARPA%20exascale%20-%20hardware%20(2008).pdf.
3. J. Dean, "Challenges in Building Large-Scale Information Retrieval Systems," keynote talk, *Proc. 2nd Ann. ACM Conf. Web Search and Data Mining* (WSDM 09), ACM Press, 2009; http://wsdm2009.org/proceedings.php.

memories can be viable DRAM replacements, achieving competitive speeds at lower power consumption, with nonvolatility properties similar to disks but without the power overhead. Additionally, recent studies have identified a slowing of DRAM growth due to scaling challenges for charge-based memories.[10,11] The adoption of NVRAM as a DRAM replacement can potentially be accelerated due to such limitations in scaling DRAM.

Density and endurance have been traditional limitations of NVRAM technologies, but recent trends suggest that these limitations can be addressed. Multilevel designs can achieve increased density, potentially allowing multiple layers per die.[12] At a single chip level, 3D die stacking using through-silicon vias for interdie communication can further increase density. Such 3D stacking also has the additional advantage of closely integrating the processor

and memory for higher bandwidth and lower power (due to short-length low-capacitance wires). Structures like wire bonding in system-in-package or package-on-package 3D stacking are already integrated into products currently on the market, such as mobile systems, while more sophisticated 3D-stacking solutions have been demonstrated in the lab.

In terms of endurance, compared to flash memories, PCMs and memristors offer significantly better functionality—10^7-10^8 writes per cell compared to the 10^5 writes per cell for flash. Optimizations at the technology, circuit, and systems levels have been shown to further address endurance issues, and more improvements are likely as the technologies mature and gain widespread adoption.[11,13]

More details about emerging nonvolatile memories can be found in several recent overviews and tutorials [14,15]—for example, HotChips 2010 (www.hotchips.org).

These trends suggest that technologies like PCM and memristors, especially when viewed in the context of advances like 3D die stacking, multicores, and improved networking, can induce more fundamental architectural change for data-intensive computing than traditional approaches that use them as solid-state disks or as another intermediate level in the memory hierarchy.

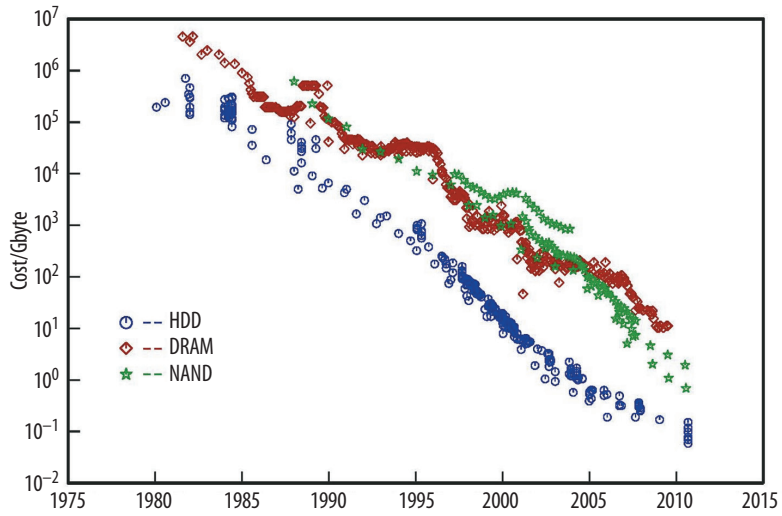

Figure 2. Nonvolatile memory cost trends. These trends suggest that future nonvolatile memories can be viable DRAM replacements.

NANOSTORES: A NEW SYSTEM ARCHITECTURE BUILDING BLOCK?

The confluence of these various trends—future large-scale distributed data-centric workloads with I/O-intensive behavior, innovations in the software stack, and the emergence of new nonvolatile memories potentially timed with the end of scaling for DRAM—offers a unique opportunity to rethink traditional system architectures and memory hierarchies in future designs.

Nanostores offer one such intuitive, and potentially advantageous, way to leverage this confluence of application and technology trends. We coined the term nanostores as a duality of microprocessors to reflect the evolution to nanotechnology and the emphasis on data instead of compute. The key property of nanostores is the colocation of processors with nonvolatile storage, eliminating many intervening levels of the storage hierarchy. All data is stored in a single-level nonvolatile memory datastore that replaces traditional disk and DRAM layers—disk use is relegated to archival backups.

For example, a single nanostore chip consists of multiple 3D-stacked layers of dense silicon nonvolatile memories such as PCMs or memristors, with a top layer of power-efficient compute cores. Through-silicon vias are used to provide wide, low-energy datapaths between the processors and the datastores. Each nanostore can act as a full-fledged system with a network interface. Individual such nanostores are networked through onboard connectors to form a large-scale distributed system or cluster akin to current large-scale clusters for data-centric computing. The system can support different network topologies, including traditional fat trees or recent proposals like HyperX.[16]

In terms of physical organization, multiple nanostore chips are organized into small daughter boards (microblades) that, in turn, plug into traditional blade server boards. Given the heat dissipation characteristics of the design, we also can envision newer packaging technologies for the broader solution. Figure 3 illustrates an example dematerialized datacenter design[17] in which the individual blade servers connect to an optical backplane "spine" with optimized airflow and packaging density.

Power and thermal issues are important concerns with 3D stacking and limit the amount of compute that a nanostore can include. Figure 4 illustrates how additional, more powerful, compute elements can be added to create a "hierarchy of computes" that back up the on-chip computation in the nanostore. This also enables repurposing the design so that nanostores act more like current systems—with powerful compute elements and deep hierarchies to data—if needed for applications such as legacy workloads.

There is a wide range of possible implementations for this high-level organization. There are numerous design choices in terms of the provisioning, organization, and balance of the compute, storage, and network per nanostore

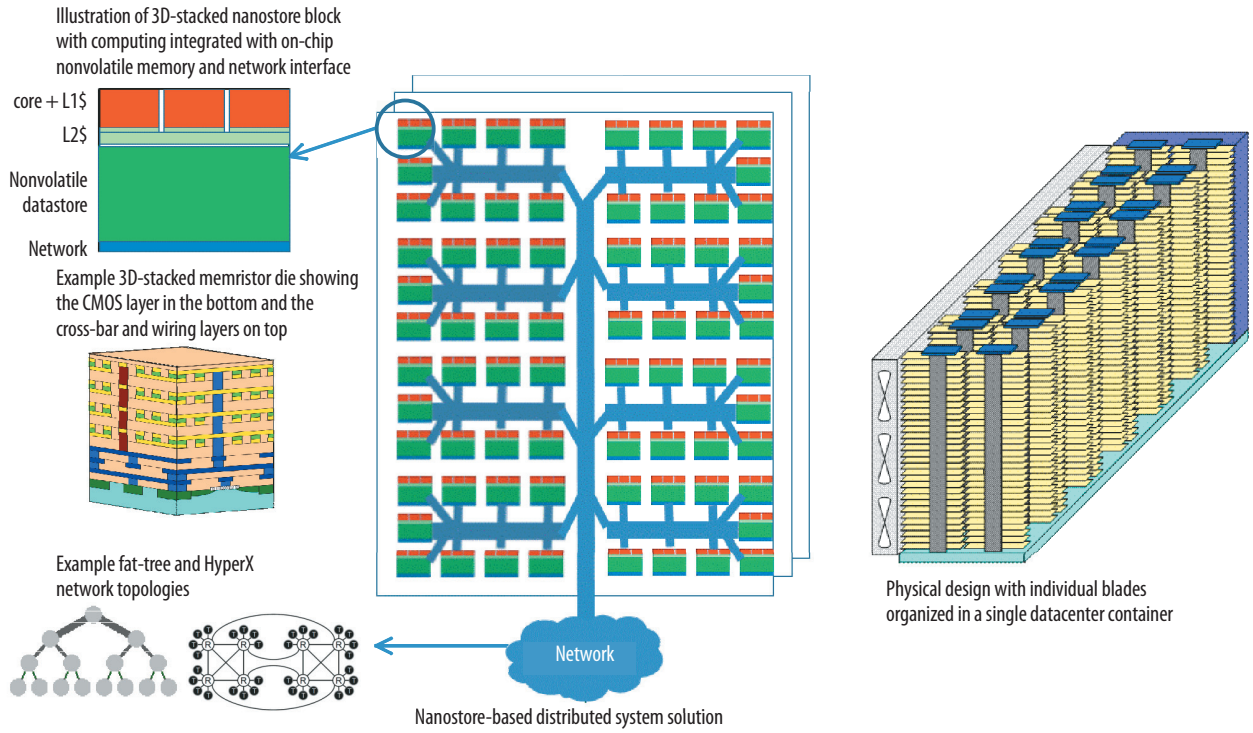

Figure 3. Nanostores colocate processors and nonvolatile memory on the same chip and connect to one another to form a larger cluster for data-centric workloads.

as well as the sharing model across the individual nodes and the network topology, including potential differences between the on-chip, onboard, and cross-cluster networks. Constraints on the design choices include technology- and circuit-level parameters such as the size of the die and the yield, the number of 3D-stacked or intradie (random) layers, as well as packaging constraints such as the power/thermal budget per chip or board.

Similarly, the software models vary depending on the specific architecture built with nanostores. A pure nanostore-to-nanostore architecture is well matched with a large-scale distributed shared-nothing system abstraction, similar to current data-centric workloads. Each nanostore can be viewed as a complete independent system executing the software stack needed to implement a data-parallel execution environment like MapReduce. Alternate designs that include multiple levels of compute will require more sophisticated software models. For example, we could consider software models similar to those under consideration for future desktop systems for coordination across general-purpose and graphics-processing units.

BENEFITS AND CHALLENGES

While similar to some of the principles used in designs such as ActiveStorage,[18] IRAM,[19] and RAMCloud,[20] the nanostore design is unique in its colocation of power-efficient computing with a single-level nonvolatile datastore and a large-scale distributed design well matched with data-centric workloads. This combination provides several benefits.

The single-level datastore enables improved performance due to faster data access in terms of latency and bandwidth. Flattening the memory hierarchy and the increased energy efficiency of NVRAM over disk and DRAM also improve energy efficiency. The large-scale distributed design facilitates higher performance from increased parallelism and higher overall data/network bandwidth. This design also improves energy efficiency by partitioning the system into smaller elements that can leverage more power-efficient components such as simpler cores. Beyond operational energy, this design also has the potential to reduce the embedded energy[17] in the system, which can lead to more sustainable designs.

An illustrative implementation provides a better estimate of these benefits. In this example, the nanostore die size is 100 mm^2, similar to the cost-effective design point for memories.

Let's assume cores in the compute layer are based on low-voltage power-efficient microarchitectures with simple SRAM cache hierarchies. Different organizations are possible for the compute layer—in the number of cores (1 to 128), clock frequency (100 MHz to 2 GHz), issue width and pipeline depth (2-way simple to 4-way deep), and L2 cache size (512 Kbytes or 1 Mbyte per core); the limiting

factor will be the power density at the socket (currently 32 watts/cm^2). For our projected timeframe, we expect 3D stacking to provide significant bandwidth (up to 32 Gbytes per second in the PicoServer design) between the processor and stacked memory, and 80 Gbps (2 x 40-Gbps NICs) networking bandwidth per system (in an equivalent traditional architecture). Assuming 25-nm technology, 8 layers of 3D, and intra-die stacking, a single node that groups nine nanostores to provide 8 + 1 redundancy can provide one-half to one terabyte of nonvolatile memory (depending on assumptions around single-level or multilevel cells) with teraops of local compute colocated with the storage (assuming simple low-power processors) and about 256 Gbytes of aggregate datastore bandwidth.

The latencies to access the data are expected to be competitive with DRAM (within about a factor of two), but at much lower energy consumption (current estimates vary from 2 to 10 picojoules/bit compared to 24 picojoules/bit for DRAM).[9] Compared to traditional disks or existing flash memories, this configuration provides several orders-of-magnitude better latencies and energy efficiency.

These numbers demonstrate the potential for better performance at improved energy efficiency with these designs. This improvement stems from more energy-efficient memory technologies, compute colocation leading to lower energy consumption in the hierarchy, and more energy-efficient balanced designs. While these are peak numbers, we have also experimented with simulation numbers for common data-centric kernels that address the key dimensions of the taxonomy discussed in Figure 1. Our results indicate significant improvements in performance and energy efficiency. For I/O-bound workloads, this can often be a few orders of magnitude higher performance at close to an order of magnitude better energy efficiency.[21] At the same time, achieving this potential presents numerous challenges.

Scalability

Given the smaller capacities of per-socket storage, the number of individual elements in the system increases dramatically. This can potentially increase the stress on the networking subsystem in terms of bandwidth contention (particularly for all-to-all communication), topological complexity and port count, and power.

Endurance

Based on current estimations of expected densities and endurance, in theory, storage wearout can occur in two years for PCMs or 11 years for memristors. However, in practice not all applications sustain rates at that level, and the average across the application is much lower, leading to much longer lifetimes across the array. Wear-leveling schemes must still be used to spread writes across the entire memory to prevent early failure of hot data blocks.

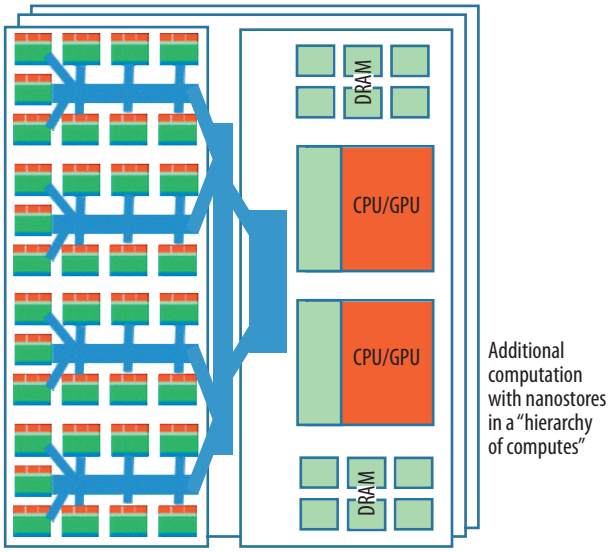

Figure 4. Adding more powerful compute elements can create a hierarchy of computes that backs up on-chip computation in the nanostore.

Assuming a previously proposed approach—start-gap wear leveling—at an efficiency of 90 percent of optimal wear-leveling (shown to be realistic for OLTP/database workloads)[13] and using the memory write bandwidths from our simulations, we estimate per-socket lifetimes of 7-18 years for our benchmarks on the PCM-based design. Nevertheless, techniques that carefully manage wearout warrant further study.

Cost

As Figure 2 shows, current flash memories are about an order of magnitude more costly on a dollar-per-byte basis compared to disk. NVRAM has the potential to lower these costs by allowing more aggressive stacking and simpler fabrication processes. The improved energy efficiency of nanostores can also further lower total costs of ownership. Based on these observations, we expect the nanostore design to be competitive in costs compared to traditional designs, but this needs to be validated with further study.

Design choices

Several interesting design questions remain to be answered. How well do nanostore designs perform compared to aggressive extrapolations of existing approaches? Are the expected benefits significant enough to warrant the change? How do the benefits change across the range of data-centric workloads? How do the benefits break down? Do we need to rethink the balance of compute, data, and network for this new architecture? What are the implications of specific design choices and technol-

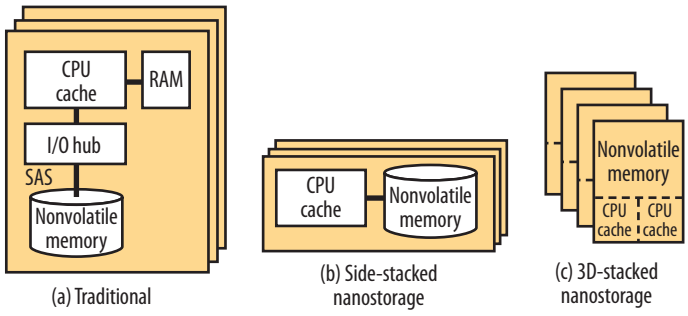

Figure 5. Three different designs offer tradeoffs for data-centric workloads: (a) traditional design, (b) nanostorage side-stacked design, and (c) nanostorage 3D-stacked design.

ogy extrapolations? In particular, what is the sensitivity to the network bandwidth assumptions and packaging limitations?

There is significant potential, and need for, additional architectural innovation in this space.

Software and systems design

Software scalability is an important issue. From 1998 to 2009, Google's infrastructure is reported to have scaled performance (queries processed per day) by 1,000 times while scaling the infrastructure by 1,000 times.[22] While large-scale deployments of data-centric workloads have been demonstrated, the sizing of the system will have to carefully consider latency requirements—for example, response time for a search request. Similarly, current software stacks include decades of development with assumptions of seek limitations of traditional magnetic disks. While some recent studies have revisited such assumptions—for example, byte-persistent file systems[23]—there is a potential, and need, for additional software innovation around datastores for nonvolatile memory-based storage systems.

Modeling and benchmarking

Any new architecture redesign comes with associated challenges in modeling and benchmarking.[24] Our focus on the combination of multiple future technologies for emerging workloads poses several challenges in the choice of benchmarks, technology parameters, and baseline systems appropriate to this longer timeframe.

To evaluate alternate designs and their tradeoffs, we need to study large-scale clusters running distributed workloads operating on large volumes of data. We also need to examine tradeoffs at the full system level, including computing, networking, memory, and storage layers. Conventional architecture simulators not only lack the ability to cope with this level of system scale, but also the modeling means for storage and network subsystems at a distributed systems level. There is also a combinatorial explosion in the design space from various assumptions at the fine-grained and coarse-grained architectural levels, as well as the choice of technology and workload parameters.

An appropriate evaluation methodology is required to systematically reason about this large design space. Similarly, a key need is the availability of a representative set of the emerging distributed workloads that drive these data-centric markets. New approaches are needed in this space as well.[5]

MATCHING NANOSTORES TO DATA-CENTRIC WORKLOADS

The benefits of colocating compute close to nonvolatile datastores can be achieved with different designs, each with different tradeoffs for specific data-centric workloads. As Figure 5 shows, these tradeoffs are best illustrated by comparing three designs—a traditional design with DRAM memory and solid-state disks, a nanostore 3D-stacked design (similar to our discussions so far), and an alternate nanostore side-stacked design that colocates the compute with the nonvolatile datastore, but separately off the memory bus (with the nonvolatile store replacing traditional disks as before).

From a data-centric workload point of view, a good way to reason across these designs is to consider the amount of raw compute processing capacity that can be applied per unit data, at both global and local levels, and the bottlenecks in the hardware and software that limit the ability to use this compute capacity.

The traditional design is likely to work well for compute-heavy workloads with small data bandwidth per cycle—for example, video transcoding—or workloads in which the hot and cold working set sizes are orders of magnitude apart—for example, image archiving. Workloads that require additional bandwidth for the underlying data and can leverage data-partitioned parallelism—for example, MapReduce workloads, sorts, clickstreams, and log analysis—can benefit from the nanostore side-stacked and nanostore 3D-stacked designs.

Rewriting the software to leverage the improved datastore latencies can provide additional benefits—until the network becomes a bottleneck. For parallel workloads that can be rewritten to use fine granularity with limited cross-cluster communication (filtering, aggregation, textual search, and so on), the nanostore 3D-stacked design is likely to work best—until the compute becomes a bottleneck. More work is needed in software for effective parallelization, but the cost and energy advantages may prove these measures to be worthwhile.

The trends toward growing pressure for improved bandwidth and latency in data-centric workloads, ongoing progress in parallelizing software, and improvements in

local interconnection networks support using a nanostore design for future systems, but hybrid designs also may emerge.

GREEN CLOUDS AND BLACK SWANS

It has been said that the essence of science is cumulative.[25] The emergence of new software and computing models, including cloud computing and the increased emphasis on data and data-centric applications, combined with exciting advances in technology—such as 3D-stacked nonvolatile memories, optics, and more power-efficient computation—provide one such opportunity for cumulative benefits. Such an opportunity represents a potential *black swan event*—a high-impact, infrequent event that in hindsight is very predictable—that presages future system architecture designs.[26]

One trend that is both logical and disruptive is colocating computing closer to the data, which will in turn lead to new building blocks for future systems. As stand-alone building blocks, a large number of individual nanostores can communicate over emerging optical interconnects and support large-scale distributed data-centric workloads. The key aspects of this approach are large-scale distributed parallelism and balanced energy-efficient compute in close proximity to the data. Together, these features allow nanostores to potentially achieve significantly higher performance at lower energy.

While such designs are promising, they are by no means inevitable, and several important design questions and challenges still remain to be addressed. Nanostores enable a rich architectural space that includes heterogeneous designs and integrated optics. There are also interesting opportunities for software optimizations including new interfaces and management of persistent datastores.

The improvements in performance, energy efficiency, and density in future system architectures will likely enable new applications across multiple larger, diverse data sources; the corresponding hardware-software codesign also provides rich opportunities for future research.

This article is intended to fuel the discussion that is needed in the broader community to start examining new, more disruptive, alternate architectures for future data-centric systems.

Acknowledgments

This article greatly benefited from the input and support of several people, notably John Sontag, Jichuan Chang, Mehul Shah, David Roberts, Trevor Mudge, Niraj Tolia, Steven Pelley, Kevin Lim, Greg Astfalk, and Norm Jouppi.

References

1. M. Mayer, "The Physics of Data," Xerox PARC Forum Distinguished Lecture, 2009; www.parc.com/event/936/innovation-at-google.html.
2. R. Winter, "Why Are Data Warehouses Growing So Fast?" 2008; www.b-eye-network.com/view/7188.
3. P. Lyman and H.R. Varian, "How Much Information?" 2003; www2.sims.berkeley.edu/research/projects/how-much-info-2003.
4. R.S. Williams, "A Central Nervous System for the Earth," *Harvard Business Rev.*, vol. 87, no. 2, 2009, p. 39.
5. M. Shah et al., "Data Dwarfs: Motivating a Coverage Set for Future Large Data Center Workloads," *Proc. Workshop Architectural Concerns in Large Datacenters* (ACLD 10), 2010; sites.google.com/site/acldisca2010.
6. J. Gray, "E-Science: The Next Decade Will Be Exciting," 2006; http://research.microsoft.com/en-us/um/people/gray/talks/ETH_E_Science.ppt.
7. B. Zhai et al., "Energy-Efficient Near-Threshold Chip Multi-Processing," *Proc. Int'l Symp. Low Power Electronics and Design* (ISLPED 07), IEEE Press, 2007, pp. 32-37.
8. C. Lam, "Cell Design Considerations for Phase Change Memory as a Universal Memory," *Proc. Int'l Symp. VLSI Technology, Systems, and Applications* (VLSI-TSA 08), IEEE Press, 2008, pp. 132-133.
9. D.B. Stukov et al., "The Missing Memristor Found," *Nature*, vol. 453, 2008, pp. 80-83.
10. ITRS Roadmap, www.itrs.net/links/2009ITRS/Home2009.htm.
11. B.C. Lee et al., "Architecting Phase Change Memory as a Scalable DRAM Alternative," *Proc. 36th Ann. Int'l Symp. Computer Architecture* (ISCA 09), ACM Press, 2009, pp. 2-13.
12. D. Lewis and H-H. Lee, "Architectural Evaluation of 3D Stacked RRAM Caches," *Proc. IEEE 3D System Integration Conf.*, IEEE Press, 2009, pp. 1-4.
13. K. Moinuddin et al., "Scalable High-Performance Main Memory System Using Phase-Change Memory Technology," *Proc. 36th Ann. Int'l Symp. Computer Architecture* (ISCA 09), ACM Press, 2009, pp. 24-33.
14. R. Bez, U. Russo, and A. Redaelli, "Nonvolatile Memory Technologies: An Overview," *Proc. Workshop Technology Architecture Interaction: Emerging Technologies and Their Impact on Computer Architecture*, 2010, pp.44-65.
15. N. Jouppi and Y. Xie, tutorial, "Emerging Technologies and Their Impact on System Design," *Proc. 15th Int'l Conf. Architectural Support for Programming Languages and Operating Systems* (ASPLOS 10), 2010; www.cse.psu.edu/~yuanxie/ASPLOS10-tutorial.html.
16. J. Ahn et al., "HyperX: Topology, Routing, and Packaging of Efficient Large-Scale Networks," *Proc. Conf. High-Performance Computing Networking, Storage and Analysis* (SC 09), ACM Press, 2009, pp. 1-11.
17. J. Meza et al., "Lifecycle-Based Data Center Design," *Proc. ASME Int'l Mechanical Eng. Congress & Exposition* (IMECE 10), Am. Soc. Mechanical Engineers, 2010; www.asmeconferences.org/congress2010.
18. E. Riedel et al., "Active Disks for Large-Scale Data Processing," *Computer*, June 2001, pp. 68-74.
19. D. Patterson et al., "A Case for Intelligent DRAM: IRAM," *IEEE Micro*, vol. 17, no. 2, 1997, pp. 33-44.

20. J. Ousterhout et al., "The Case for RAMClouds: Scalable High-Performance Storage Entirely in DRAM," *ACM SIGOPS Operating Systems Rev.*, vol. 43, no. 4, 2009, pp. 92-105.
21. J. Chang et al., "Is Storage Hierarchy Dead? Co-located Compute-Storage NVRAM-Based Architectures for Data-Centric Workloads," tech. report HPL-2010-114, HP Labs, 2010.
22. J. Dean, "Challenges in Building Large-Scale Information Retrieval Systems," keynote talk, *Proc. 2nd Ann. ACM Conf. Web Search and Data Mining* (WSDM 09), ACM Press, 2009; http://wsdm2009.org/proceedings.php.
23. J. Condit et al., "Better I/O through Byte-Addressable, Persistent Memory," *Proc. ACM SIGOPS 22nd Ann. Symp. Operating Systems Principles* (SOSP 09), ACM Press, 2009, pp. 133-146.
24. B. Dally, "Moving the Needle: Effective Computer Architecture Research in Academe and Industry," keynote talk, *Proc. 37th Int'l Symp. Computer Architecture* (ISCA 10), ACM Press, 2010; www.hipeac.net/node/2903.
25. R. Hamming, "You and Your Research," 1986; www.cs.virginia.edu/~robins/YouAndYourResearch.pdf.
26. P. Ranganathan, "Green Clouds, Red Walls, and Black Swans," keynote presentation, *Proc. 7th IEEE Int'l Conf. Autonomic Computing* (ICAC 10), IEEE Press, 2010; www.cis.fiu.edu/conferences/icac2010.

Parthasarathy Ranganathan *is a distinguished technologist at HP Labs, where he is the principal investigator for the exascale datacenter project. His research interests are in systems architecture and energy efficiency. Partha received a PhD in electrical and computer engineering from Rice University. Contact him at partha.ranganathan@hp.com.*