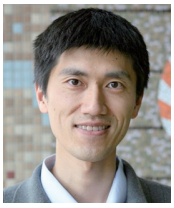
Department Editors: Kevin Rudd and Kevin Skadron

(Re)Designing Data-Centric Data Centers



PARTHASARATHY RANGANATHAN
JICHUAN CHANG
HP Labs

We are entering an exciting era for systems design—one driven by data-centric computing. A recent report from the University of San Diego estimated that, conservatively, enterprise server systems have processed and delivered more than 9 zettabytes of information in 2008 (where 1 zettabyte = 10^{21} bytes);[1] this number is projected to double every two years. Walmart servers, for example, handle more than 1 million customer transactions every hour, feeding databases estimated in several petabytes. High-performance computing systems working with the Large Hadron Collider filter through roughly one petabyte of data per second and still produce 15 petabytes a year after multiple levels of data selection. Each day, Facebook operates on nearly 100 terabytes of user log data and several hundred terabytes of user pictures; similarly, 48 hours of video content is uploaded every minute on YouTube (a sixfold increase from four years ago).[2]

This vast and growing amount of information represents both an opportunity and a challenge. On one hand, the ability to collect and process large volumes of new data can drive scientific breakthroughs, new business process optimizations, and day-to-day improvements in our personal lives. Recent data-centric applications for personalized genome sequencing, real-time trends from business analytics, social-network-based recommendations, and so on illustrate this potential. But on the other hand, this data is also creating a host of new problems. In particular, the growth in data produced is outpacing the improvements in the cost and density of storage technologies. Also, perhaps more importantly, our ability to process the data to extract meaningful, actionable insights is significantly lagging our ability to collect and store data.

Given these challenges and opportunities, it is important to rethink how we design future data-centric systems. At the same time, technology inflections such as the increased adoption of nonvolatile memories, optical communications, multicores, and heterogeneous computing all provide a unique opportunity for an end-to-end redesign of data-centric solutions across both hardware and software. Here, we discuss recent computer architecture and systems research matched with such redesigns, culling out cross-cutting directions across these projects that suggest research opportunities for the broader community.

Rethinking data-centric system architectures

Historically, system architecture designs have been driven by advances in processor designs, with the performance of the I/O subsystem usually only a secondary design consideration. Indeed, most popular analyses of improvements in system architecture typically only track base computing performance (for example, historical flops at http://top500.org). However, as future systems are increasingly used to capture, classify, analyze, manage, and archive large volumes of data, we will need a corresponding rethinking of system architecture focused on data storage and management.

Figure 1a presents an overview of the continuum of different architectural organizations to address data management. At the left is a traditional system design using mechanical disks as the persistent data store and DRAM memory as a caching layer. Further in the continuum, several products (such as EMC, Fusion-IO, HP, Oracle, Seagate, and Texas Memory Systems) expose Flash-based nonvolatile memories as block devices either through Serial Attached SCSI (SAS) and Serial Advanced Technology Attachment (SATA) or PCI Express interfaces; the flash memories are used as disk replacements or disk caches, with appropriate software support (such as Fusion-IO drivers, Oracle ASM, and Facebook Flashcache). Flash can also be combined with disaggregated memory to provide large memory space at low cost.[3] Further out, several research studies have also discussed using nonvolatile memory such as phase-change memory (PCRAM) or memristor as byte-addressable memory devices off

Published by the IEEE Computer Society

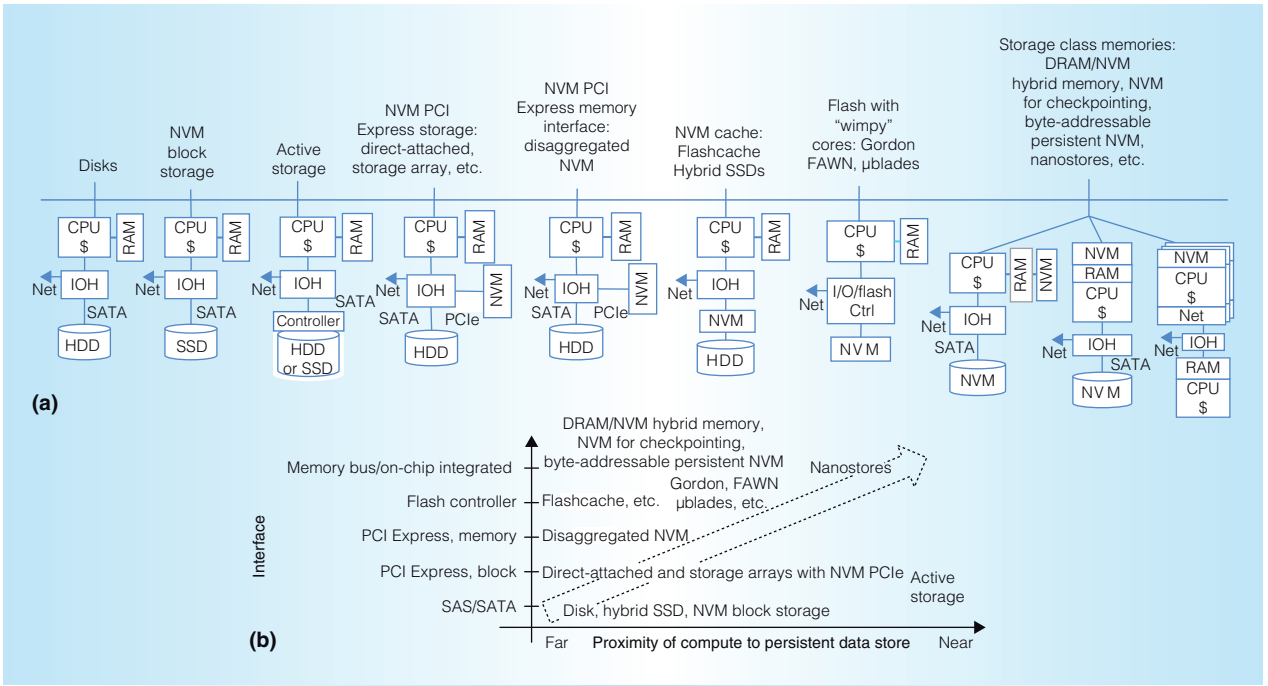


Figure 1. A visual taxonomy of recent system designs for data-centric applications. Overview of system architectures of different designs for data management (a). Classification of different system designs to illustrate trends (b). (PCIe: PCI Express.)

the memory bus or 3D-stacked on the chip.[4-6] Some proposals have used the nonvolatile memory as additional levels of memory caches,[7,8] and others have used the nonvolatile memory as a replacement for the persistent data store,[9] including with collapsed hierarchies in distributed systems.[6] Figure 1b presents an alternate view of these designs, classified along two dimensions—the type of interface to the nonvolatile memory that is exposed (*y*-axis), and the proximity of the computing to the persistent data store (*x*-axis); the arrow highlights the direction of recent shifts. Classifying current approaches in such a view illustrates several interesting trends.

Rethinking memory and storage hierarchy in future system architectures

A key trend, evident from Figure 1, is that persistent data storage is steadily migrating from slow (disk-like) interfaces to faster (memory-like) interfaces with increasing flexibility and performance. Extrapolating these trends, future designs will have computing and persistent data store in a single die. However, there are several open research questions. What are the implications of different nonvolatile memory technologies on traditional memory and storage hierarchies? What are the tradeoffs between the different memory organizations in Figure 1? Are organizations possible that collapse the memory hierarchy and reduce energy overheads of data movement? Can we design heterogeneous or morphable organizations that match specific application characteristics to specific memory technology features (both strengths and weaknesses)? How do we address new resiliency challenges introduced by endurance limits in nonvolatile memory?

Rethinking the balance between the data store, compute, and communication

Several recent proposals (such as µblades,[10] FAWN,[11] and Gordon[12]) combine Flash-based storage with ''wimpy'' lower-power processors for better system balance to improve energy efficiency for cloud workloads. More generally, rethinking the storage and memory hierarchy will create new system bottlenecks, altering the traditional balance between the storage, compute, and communication performance. For example, how do we design new computing structures to better exploit the huge bandwidth enabled by through-silicon-vias in 3D-stacked persistent data stores? What are the appropriate applications for different kinds of compute cores—wimpy and brawny? Do we need to redesign communication provisioning, particularly for large-scale distributed data centers? How should we use energy-efficient high-radix optical communication in future designs?

Moving compute to the action

Another important trend is moving the compute closer to the data. Recent distributed data management frameworks such as MapReduce/Hadoop already operate at large scale by partitioning the data set across individual

nodes and scheduling tasks matched to the data they operate on. With increasing energy costs from excessive and inefficient data movement, colocating compute closer to data within the memory hierarchy might have significant benefits. For example, Micron recently announced its hybrid memory cube technology, which couples a logic layer with 3D-stacked DRAM on the same chip,[13] and the nanostore proposal[6] seeks to colocate compute with the persistent data stores. Such ideas are thematically similar to previous ideas such as Active Storage (more capable disk controllers for offloading and streaming),[14] Intelligent RAM (co-located vector processors with DRAM),[15] or Processor-in-Memory,[16] but with different instantiations in the context of emerging technologies and future distributed architectures. However, several open questions remain. What is the appropriate system organization? Should we be considering a hierarchy of computing elements surrounding the data store, inverting the traditional model of data hierarchies surrounding computation? How do we develop appropriate software models to offload and coordinate computation across the various computational units?

Matching compute to the action

Recent studies have argued that limited power budgets in future processors could lead to ''dark silicon''[17,18]—designs where only some parts of a chip are used at any given point in time—potentially leading to more specialization in future processors (for example, 10×10[19-21]). Such specialization can provide significant energy efficiency advantages. Prior work has examined special-purpose architectures optimized for specific workloads, including use of GPUs, field-programmable gate arrays (FPGAs), and even application-specific integrated circuits (ASICs). More work is needed, however, to understand how these designs apply to broader data-centric workloads. The appropriate system architecture and software model for such heterogeneous architectures is also an open question.

Rethinking software interfaces and algorithms

Rethinking system architecture will require rethinking systems software as well. Specifically, with improvements in hardware performance and balance, software efficiency will become the next bottleneck. For example, researchers have already identified traditional communication stacks' software overheads as key bottlenecks in optimized distributed systems, such as in the Stanford RAMCloud project[22] or in Google distributed clusters.[23] The byte-addressability of emerging nonvolatile memories also opens up possibilities for new persistent data stores with random access semantics, massive aggregated throughput, and energy-efficient access. New optimizations are possible at various levels of the software stack: interfaces, low-level device drivers, data storage systems, and higher-level algorithms.

New interfaces

With nonvolatile memories, system architects can design systems where memory writes are instantly durable, but at the same time, this removes a degree of isolation and security provided by indirection. Two recent proposals, Mnemosyne[24] and NV-heaps,[25] have examined user-level interfaces for safely and efficiently using nonvolatile memory, via durable memory transactions. Additional new interfaces could be beneficial, for example, to explicitly reason about volatility of data, for abstractions to distinguish persistent data such as files from volatile data such as virtual memory, or user-selectable consistency and resilience semantics. Similarly, new software–hardware interfaces can better support other architectural trends such as multicores[26] or GPUs.[27]

New data stores and systems software

Several research studies have examined the redesign of data stores and data structures, such as B-trees,[28] file systems,[29,30,9] key-value stores,[31] and databases.[32] These examples have demonstrated that with careful consideration of the tradeoffs, nonvolatile memory can provide significant performance advantages without compromising persistence guarantees. But more research opportunities remain. For example, can we design new database-join algorithms to better leverage high-radix optical connections? How can we codesign across hardware and software for large in-memory data stores? How can we design future file systems to avoid copying and to leverage persistent data stores? How can such software approaches further take advantage of optimizations such as compute hierarchies or accelerators? Similarly, traditional operating systems' architecture and abstractions were developed in the era of slow disks and limited memories. Improvements to the data path such as with nonvolatile memory could require corresponding systems software redesign including possibly greater embedded management in the hardware to avoid kernel overheads.

Information will be the most valuable resource in the 21st century. Operating on large volumes of diverse data sources to get the right actionable insights at the right time presents new challenges and opportunities for system design. Addressing these opportunities requires a rethinking of future server and data center design—with a *data-centric* focus across both hardware and software. Here, we've presented a brief introduction to some recent research activities in this exciting emerging area, with a specific focus on system architecture and systems software.

There are also other important research challenges that we didn't discuss. Notably, more work is needed in new benchmarks and modeling methodologies for future data-centric data centers. Similarly, significant opportunities exist for applications enabled by new data-centric data center designs: for example, sophisticated, yet cost-effective, insight generation from huge existing volumes

of archival data (''data-at-rest''), or non-traditional ''brain-inspired'' systems that mimic neural algorithms for efficient information processing.[33]

While this area is relatively nascent, these opportunities herald a future data-centric data center that will differ significantly from current designs. In particular, we believe that the distinction between traditional memory and storage hierarchies will be blurred and traditional wisdom on the size and depth of data hierarchies will be revisited. We also believe that computing will be pervasively embedded within the system design colocated with data storage and data communication, and traditional general-purpose server-class processing will be supplemented with additional, more specialized forms of computation. We also anticipate a software stack significantly redesigned to eliminate the inefficiencies in current solutions, with new byte-addressable persistent stores, and new algorithms matched with the advances in the hardware and software architecture. In combination, the advances in technology, hardware architecture, software systems, and higher-level algorithms will enable better, faster, cheaper data-centric computing, which in turn can enable new applications to operate on larger volumes of data to extract better insights and enable greater automation.

This future looks exciting, but our discussion only scratches the surface of what is possible. Overall, we believe that the broad area of data-centric data centers offers a rich opportunity for more innovation from the broader community, and we hope that this column helps fuel additional thinking in this important area.

References

1. J.E. Short, R.E. Bohn, and C. Baru, ''How Much Information 2010: Report on Enterprise Server Information,'' 2011; http://hmi.ucsd.edu/pdf/HMI_2010_EnterpriseReport_Jan_2011.pdf.
2. ''Data, Data Everywhere,'' *The Economist,* 25 Feb. 2010.
3. K.T. Lim et al., ''Disaggregated Memory for Expansion and Sharing in Blade Servers,'' *Proc. 36th Ann. Int'l Symp. Computer Architecture,* ACM Press, 2009, pp. 267-278.
4. M.K. Qureshi, V. Srinivasan, and J.A. Rivers, ''Scalable High Performance Main Memory System using Phase-Change Memory Technology,'' *Proc. 36th Ann. Int'l Symp. Computer Architecture,* ACM Press, 2009, pp. 24-33.
5. B.C. Lee et al., ''Phase Change Technology and the Future of Main Memory,'' *IEEE Micro,* vol. 30, no. 1, 2010, pp. 131-141.
6. P. Ranganathan, ''From Microprocessors to Nanostores: Rethinking Data-Centric Systems,'' *Computer,* vol. 44, no. 1, 2011, pp. 39-48.
7. X. Wu et al., ''Hybrid Cache Architecture with Disparate Memory Technologies,'' *Proc. 36th Ann. Int'l Symp. Computer Architecture,* ACM Press, 2009, pp. 34-45.
8. C.W. Smullen et al., ''Relaxing Non-Volatility for Fast and Energy-Efficient STT-RAM Caches,'' *Proc. 2011 IEEE 17th Int'l Symp. High Performance Computer Architecture,* IEEE Press, 2011, pp. 50-61.
9. J. Condit et al., ''Better I/O Through Byte-Addressable, Persistent Memory,'' *Proc. ACM SIGOPS 22nd Symp. Operating Systems,* ACM Press, 2009, pp. 133-146.
10. K.T. Lim et al., ''Understanding and Designing New Server Architectures for Emerging Warehouse-Computing Environments,'' *Proc. 35th Ann. Int'l Symp. Computer Architecture,* IEEE CS Press, 2008, pp. 315-326.
11. D.G. Andersen et al., ''FAWN: A Fast Array of Wimpy Nodes,'' *Proc. ACM SIGOPS 22nd Symp. Operating Systems Principles,* ACM Press, 2009, pp. 1-14.
12. A.M. Caulfield, L.M. Grupp, and S. Swanson, ''Gordon: Using Flash Memory to Build Fast, Power-Efficient Clusters for Data-Intensive Applications,'' *Proc. 14th Int'l Conf. Architectural Support for Programming Languages and Operating Systems,* ACM Press, 2009, pp. 217-228.
13. J.T. Pawlowski, ''Micron Hybrid Memory Cube (HMC),'' HotChips 23, 2011.
14. E. Riedel, G.A. Gibson, and C. Faloutsos, ''Active Storage for Large-Scale Data Mining and Multimedia,'' *Proc. 24rd Int'l Conf. Very Large Data Bases,* Morgan Kaufmann, 1998, pp. 62-73.
15. D. Patterson et al., ''A Case for Intelligent DRAM: IRAM,'' *IEEE Micro,* vol. 17, no. 2, 1997, pp. 33-44.
16. T. Sunaga et al., ''A Processor in Memory Chip for Massively Parallel Embedded Applications,'' *IEEE J. Solid State Circuits,* Oct. 1996, pp. 1556-1559.
17. H. Esmaeilzadeh et al., ''Dark Silicon and the End of Multicore Scaling,'' *Proc. 38th Ann. Int'l Symp. Computer Architecture,* ACM, 2011, pp. 365-376.
18. N. Hardavellas et al., ''Toward Dark Silicon in Servers,'' *IEEE Micro,* vol. 31, no. 4, 2011, pp. 6-15.
19. A.A. Chien, ''10 × 10: Taming Heterogeneity for General-Purpose Architecture,'' *Proc. 2nd Workshop New Directions in Computer Architecture,* 2011, http://ndca2.saclay.inria.fr/papers/chien.pdf.
20. G. Venkatesh et al., ''Conservation Cores: Reducing the Energy of Mature Computations,'' *Proc. 15th Int'l Conf. Architectural Support for Programming Languages and Operating Systems,* ACM Press, 2010, pp 205-218.
21. V. Govindaraju, C.-H. Ho, and K. Sankaralingam, ''Dynamically Specialized Datapaths for Energy Efficient Computing,'' *Proc. IEEE 17th Int'l Conf. High Performance Computer Architecture,* IEEE Press, 2011, pp. 503-514.
22. J. Ousterhout and P. Agrawal et al., ''The Case for RAMCloud,'' *Comm. ACM,* vol. 54, no. 7, 2011, pp. 121-130.
23. L.A. Barroso, ''Warehouse-Scale Computing: Entering the Teenage Decade,'' Federated Computing Research Conf., 2011.
24. H. Volos, A.J. Tack, and M.M. Swift, ''Mnemosyne: Lightweight Persistent Memory,'' *Proc. 16th Int'l Conf. Architectural Support for Programming*

Languages and Operating Systems, ACM Press, 2011, pp. 91-104.
25. J. Coburn et al., ''NV-Heaps: Making Persistent Objects Fast and Safe with Next-Generation, NonVolatile Memories,'' *Proc. 16th Int'l Conf. Architectural Support for Programming Languages and Operating Systems,* ACM Press, 2011, pp. 105-118.
26. D.A. Holland and M.I. Seltzer, ''Multicore OSes: Looking Forward from 1991, er, 2011,'' *Proc. 13th USENIX Conf. Hot Topics in Operating Systems,* USENIX Assoc., 2011, p. 33.
27. C.J. Rossbach, J. Currey, and Emmett Witchel, ''Operating Systems Must Support GPU Abstractions,'' *Proc. 13th USENIX Conf. Hot Topics in Operating Systems,* USENIX Assoc., 2011, p. 32.
28. S. Chen, P.B. Gibbons, and S. Nath, ''Rethinking Database Algorithms for Phase Change Memory,'' *Proc. 5th Biennial Conf. Innovative Data Systems Research,* 2011; http://www.cidrdb.org/cidr2011/Papers/CIDR11_Paper3.pdf.
29. M. Wu and W. Zwaenepoel, ''eNVy: A NonVolatile, Main Memory Storage System,'' *Proc. 6th Int'l Conf. Architectural Support for Programming Languages and Operating Systems,* ACM Press, 1994, pp. 86-97.
30. P.M. Chen et al., ''The Rio File Cache: Surviving Operating System Crashes,'' *Proc. 7th Int'l Conf. Architectural Support for Programming Languages and Operating Systems,* ACM Press, 1996, pp. 74-83.
31. S. Venkataraman et al., ''Consistent and Durable Data Structures for NonVolatile Byte-Addressable Memory,'' *Proc. 9th USENIX Conf. File and Storage Technologies,* USENIX Assoc., 2011, p. 5.
32. M. Athanassoulis et al., ''Flash in a DBMS: Where and How?'' *IEEE Data Eng. Bull.,* vol. 33, no. 4, 2010, pp. 28-34.
33. G. Snider et al., ''From Synapses to Circuitry: Using Memristive Memory to Explore the Electronic Brain,'' *IEEE Computer,* vol. 44, no. 2, 2011, pp. 21-28.

Parthasarathy Ranganathan is a Fellow at HP Labs. His research interests include system architecture and energy-efficient design. Ranganathan received his PhD in electrical and computer engineering from Rice University. He is also an IEEE Fellow.

Jichuan Chang is a senior research scientist at HP Labs. His research interests include computer system architecture and memory systems. Chang received his PhD in computer sciences from the University of Wisconsin-Madison. He is a senior member of IEEE.

Direct questions or comments about this article to Parthasarathy Ranganathan at partha.ranganathan@hp.com or to Jichuan Chang at jichuan.chang@hp.com.