

Accelerating Database Workloads by Software-Hardware-System Co-design

Rajesh R. Bordawekar
IBM T. J. Watson Research Center

Mohammad Sadoghi
IBM T. J. Watson Research Center

Abstract—The key objective of this tutorial is to provide a broad, yet an in-depth survey of the emerging field of co-designing software, hardware, and systems components for accelerating enterprise data management workloads. The overall goal of this tutorial is two-fold. First, we provide a concise system-level characterization of different types of data management technologies, namely, the relational and NoSQL databases and data stream management systems from the perspective of analytical workloads. Using the characterization, we discuss opportunities for accelerating key data management workloads using software and hardware approaches. Second, we dive deeper into the hardware acceleration opportunities using Graphics Processing Units (GPUs) and Field-Programmable Gate Arrays (FPGAs) for the query execution pipeline. Furthermore, we explore other hardware acceleration mechanisms such as single-instruction multiple-data (SIMD) that enables short-vector data parallelism.

I. TUTORIAL OBJECTIVES

The core part of the tutorial is to sketch a roadmap for database hardware-acceleration based on a comprehensive discussion and classification of existing work that not only highlights strengths and novelties of related work, but also critically identifies the limitations for their wide-adoption in practice and open problems from both industry and academic perspectives.

This tutorial features four parts: (1) overview of system-level characterization of database systems with the special attention to areas with high potential for hardware acceleration; (2) overview of the SIMD, GPU, and FPGA architectures and programming models, e.g., GPGPU programming using the Nvidia ecosystem and FPGA programming using the Hardware Description Language (HDL) and its synthesis/deployment process; (3) acceleration of key database kernels using SIMD, GPUs, and FPGAs (guided by existing research); and (4) discussion of the chief limitations and open problems for deploying hardware-acceleration to large-scale data management systems in practice.

For the GPU portion, we overview the GPU architecture and outline the data parallel programming models such as CUDA and OpenCL. Subsequently, we focus specifically on Nvidia's GPUs and describe key features of the CUDA toolkit through code snippets and examples for database kernel operations such as hashing, sorting, and joins. Similarity, for the FPGA portion, we overview the FPGA architecture and the HDL hardware programming model, which demands a different way of thinking about the problem, a major divergence from the traditional software development mindset. In particular, we describe the FPGA synthesis process and the

pros and cons of reconfigurability and reprogrammability of FPGAs through examples and code snippets. This material serve as perquisite, before providing an in-depth explanation and analysis of hardware acceleration for key database kernels and operations (e.g., sorting, joins, filtering, and compression).

II. DETAILED TUTORIAL OUTLINE

System-level characterization of database systems: We begin the tutorial by highlighting key database technologies that go beyond traditional relational systems and the acceleration challenges imposed by today's architectural limitations. In the data processing space, we discuss row- and column-oriented query execution pipelines, data stream computation, and frequent pattern matching/mining. We also highlight the key architectural challenges such as Moore's law physical limitation, memory wall and Von Neumann's bottleneck, large and complex control units, and power consumption.

Hardware programming ecosystem: We provide a brief introduction to data-parallel execution using SIMD instructions, general-purpose computing using GPUs (GPGPUs) and FPGAs, and HDL programming. In this phase, we will cover the basics of SIMD, GPU, and FPGA architectures and programming models. We also describe the entire pipeline and life-cycle for synthesis and re-programmability of FPGAs and discuss the key differences between designing software threading model and hardware logical flow. In particular, our discussion is focused on the Nvidia ecosystem for GPUs. More specifically, we describe how to use the CUDA toolkit for programming GPUs. For FPGAs, we will focus on the Xilinx ecosystem and toolkit and how to design a logical flow based on a high-level hardware description language, such as Verilog, VHDL, or SystemVerilog. These languages enable to specify the circuit behaviorally using traditional C/C++-style constructs for condition, loop, and data types.

After providing an architectural overview of GPGPU and FPGA, e.g., heterogeneous system design, memory subsystem, and compute architecture, we discuss the evolution and life-cycle of these different hardware acceleration paradigms. For GPGPUs, we highlight programming evolution from *sh* to *OpenCL* and *CUDA* while focusing on single-instruction multiple-thread (SIMT) model and the hybrid off-loading in GPU programming. For FPGAs, we discuss programming life-cycle starting from Verilog to synthesis and re-programming of FPGAs (Xilinx toolkit). We further explain the re-programmability using lookup tables (LUTs), the routing architecture and interconnect, utilization of block memory (FPGA on-chip memory) organization and coupling, and comparison of data parallel vs. pipeline programming model. For

SIMD, we present how to enable short-vector data parallelism using Intel x86 and Power architectures.

Lastly, we explore advance topics including communication optimization by focusing on the impact of PCI connectivity, the host-device memory sharing, the multi-GPU/FPGAs programming and communications, and the power consumption and other energy-related issues. We also outline a set of performance optimization methodologies including thread mapping vs. custom logic circuit on hardware strategies, exploiting (distributed) shared memory and shared virtual space, understanding off- and on-chip memory (processor and data coupling), and improving GPUs and FPGAs device utilization.

Accelerating data management workloads using GPUs/FPGAs: The core of the tutorial will be dedicated to surveying existing work and discussing the key opportunities for exploiting GPUs and FPGAs for relational analytical workloads and streaming applications. We will discuss key database kernels such as hashing, sorting, join, and OLAP operations such as aggregation. For these kernels, we analyze their implications on system design, e.g., how does cost-based query optimization affect the hybrid system design, what are the impacts on the data layout strategies (row vs. column organizations), or how to identify off-loadable database kernels for hardware acceleration.

Additionally, we discuss the role of hardware acceleration for data streams. In particular, we highlight major works in the area of SQL query execution using sliding window semantics, regular expression pattern matching using finite-state machines, complex-event processing using high-dimensional indexing and Rete network, and SQL query compiler design and multi-query optimization opportunities. Lastly, we explore advance topics such as (1) best-effort computation including pre-filtering approaches for dividing computation in the co-processor model; (2) supporting online changes to query/data workloads on hardware (i.e., trade-offs between query performance and flexibility) such as modification to existing queries, addition/removal of queries, schema changes, online query topology re-construction for multi-query optimization; and (3) co-processor execution model by combining CPUs, GPUs, and FPGAs as either primary data storage units or within clustered systems.

Open problems and practical limitation of hardware accelerations: We conclude the tutorial with final remarks on benefits and challenges of employing modern hardware accelerators in practice. We offer our views and discuss open problems, namely, closing the gap between the software flexibility and the performant, but inflexible, hardware solutions, from the perspectives of both industry and academic research. In particular, we raise and explore the following questions. (1) How to achieve line-rate data processing (what level of parallelism can be attained)? (2) How to overcome the hardware inflexibility and development cost challenges? (3) How/Where to place hardware accelerators in query execution pipelines in practice? (4) What are the power and energy consumption benefits of hardware acceleration?

ACKNOWLEDGMENTS

We wish to thank Hans-Arno Jacobsen for many insightful discussions and invaluable feedback in the earlier stages of this work.

REFERENCES

- [1] IBM Corporation. IBM Netezza 1000 data warehouse appliance. 2009.
- [2] Intel data direct I/O technology (Intel DDIO): A primer. Technical report, Intel Corporation, 2012.
- [3] J. Auerbach, D. F. Bacon, P. Cheng, and R. Rabbah. Lime: a java-compatible and synthesizable language for heterogeneous architectures. In *Proceedings of the ACM international conference on Object oriented programming systems languages and applications*, OOPSLA '10, pages 89–108, Reno/Tahoe, Nevada, USA, 2010. ACM.
- [4] J. a. Bispo, I. Sourdis, J. a. M. P. Cardoso, and S. Vassiliadis. Synthesis of regular expressions targeting FPGAs: current status and open issues. In *Proceedings of the 3rd international conference on Reconfigurable computing: architectures, tools and applications*, ARC'07, pages 179–190, Mangaratiba, Brazil, 2007. Springer-Verlag.
- [5] S. Breßand G. Saake, Why it is time for a HyPE: A hybrid query processing engine for efficient GPU coprocessing in DBMS, *Proc. VLDB Endow.*, vol. 6, no. 12, pp. 1398–1403, Aug. 2013.
- [6] R. Bordawekar, B. Blainey and R. Puri. *Analyzing Analytics*. Synthesis Lectures on Computer Architecture. Morgan & Claypool Publishers, November 2015.
- [7] J. Chhugani, A. D. Nguyen, V. W. Lee, W. Macy, M. Hagog, Y.-K. Chen, A. Baransi, S. Kumar, and P. Dubey, Efficient implementation of sorting on multi-core SIMD CPU architecture, *Proc. VLDB Endow.*, vol. 1, no. 2, pp. 1313–1324, Aug. 2008.
- [8] R. Fang, B. He, M. Lu, K. Yang, N. K. Govindaraju, Q. Luo, and P. V. Sander, GPUQP: Query co-processing using graphics processors, in *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '07. New York, NY, USA: ACM, 2007, pp. 1061–1063.
- [9] W. Fang, B. He, and Q. Luo, Database compression on graphics processors, *Proc. VLDB Endow.*, vol. 3, no. 1-2, pp. 670–680, Sep. 2010.
- [10] J. Frigo, M. Gokhale, and D. Lavenier. Evaluation of the streams-C C-to-FPGA compiler: an applications perspective. In *Proceedings of the 2001 ACM/SIGDA ninth international symposium on Field programmable gate arrays*, FPGA'01, pages 134–140, Monterey, California, USA, 2001. ACM.
- [11] B. Gedik, R. R. Bordawekar, and P. S. Yu. CellSort: high performance sorting on the cell processor. In *Proceedings of the 33rd international conference on Very large data bases*, VLDB'07, pages 1286–1297, Vienna, Austria, 2007. VLDB Endowment.
- [12] M. B. Gokhale, J. M. Stone, J. Arnold, and M. Kalinowski. Stream-oriented FPGA computing in the streams-C high level language. FCCM'00, pages 49–, Washington, DC, USA, 2000. IEEE Computer Society.
- [13] N. Govindaraju, J. Gray, R. Kumar, and D. Manocha, GPUSort: High performance graphics co-processor sorting for large database management, in *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '06. New York, NY, USA: ACM, 2006, pp. 325–336.
- [14] A. Hagiescu, W.-F. Wong, D. F. Bacon, and R. Rabbah. A computing origami: Folding streams in FPGAs. In *Proceedings of the 46th Annual Design Automation Conference*, DAC '09, pages 282–287, New York, NY, USA, 2009. ACM.
- [15] X. Han, G. Kim, G. Lipovski, and R. Chen. An optical centralized shared-bus architecture demonstrator for microprocessor-to-memory interconnects. *IEEE STQE'03*, 2003.
- [16] J. L. Hennessy and D. A. Patterson. *Computer Architecture, Fourth Edition: A Quantitative Approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2006.
- [17] M. C. Herbordt, Y. Gu, T. VanCourt, J. Model, B. Sukhwani, and M. Chiu. Computing models for FPGA-based accelerators. *Computing in Science and Engg.*, 10(6):35–45, Nov. 2008.

- [18] M. C. Herbordt, T. VanCourt, Y. Gu, B. Sukhwani, A. Conti, J. Model, and D. DiSabello. Achieving high performance with FPGA-based computing. *Computer*, 40(3):50–57, Mar. 2007.
- [19] H. Inoue, T. Moriyama, H. Komatsu, and T. Nakatani, AA-Sort: A new parallel sorting algorithm for multi-core SIMD processors, in *Proceedings of the 16th International Conference on Parallel Architecture and Compilation Techniques*, ser. PACT '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 189–198.
- [20] T. Kaldewey, G. Lohman, R. Mueller, and P. Volk, GPU join processing revisited, in *Proceedings of the Eighth International Workshop on Data Management on New Hardware*, ser. DaMoN '12. New York, NY, USA: ACM, 2012, pp. 55–62.
- [21] D. Koch and J. Torresen, FPGASort: A high performance sorting architecture exploiting run-time reconfiguration on FPGAs for large problem sorting, in *Proceedings of the 19th ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, ser. FPGA '11. New York, NY, USA: ACM, 2011, pp. 45–54.
- [22] J. W. Lockwood, A. Gupte, N. Mehta, M. Blott, T. English, and K. A. Vissers. A low-latency library in FPGA hardware for high-frequency trading (HFT). In *IEEE 20th Annual Symposium on High-Performance Interconnects, HOTI 2012, Santa Clara, CA, USA, August 22-24, 2012*, pages 9–16. IEEE Computer Society, 2012.
- [23] A. d. C. Lucas, S. Heithecker, and R. Ernst. FlexWAFE - a high-end real-time stream processing library for FPGAs. In *Proceedings of the 44th annual Design Automation Conference, DAC'07*, pages 916–921, San Diego, California, 2007. ACM.
- [24] I. Mandal. A low-power content-addressable memory (CAM) using pipelined search scheme. In *Proceedings of the International Conference and Workshop on Emerging Trends in Technology, ICWET '10*, pages 853–858, Mumbai, Maharashtra, India, 2010. ACM.
- [25] A. Margara and G. Cugola. High performance content-based matching using GPUs. In *Proceedings of the 5th ACM international conference on Distributed event-based system, DEBS'11*, pages 183–194. ACM, 2011.
- [26] A. Mitra, M. R. Vieira, P. Bakalov, V. J. Tsotras, and W. A. Najjar. Boosting XML filtering through a scalable FPGA-based architecture. In *Fourth Biennial Conference on Innovative Data Systems Research, CIDR'09, Asilomar, CA, USA, 2009*. www.cidrdb.org.
- [27] G. W. Morris, D. B. Thomas, and W. Luk. FPGA accelerated low-latency market data feed processing. In *Proceedings of the 2009 17th IEEE Symposium on High Performance Interconnects, HOTI '09*, pages 83–89, Washington, DC, USA, 2009. IEEE Computer Society.
- [28] R. Moussalli, R. J. Halstead, M. Salloum, W. A. Najjar, and V. J. Tsotras, Efficient XML path filtering using GPUs, in *International Workshop on Accelerating Data Management Systems Using Modern Processor and Storage Architectures - ADMS 2011, Seattle, WA, USA, September 2, 2011.*, 2011, pp. 9–18.
- [29] R. Moussalli, M. Salloum, W. A. Najjar, and V. J. Tsotras, Massively parallel XML twig filtering using dynamic programming on fpgas, in *Proceedings of the 27th International Conference on Data Engineering, ICDE 2011, April 11-16, 2011, Hannover, Germany*, 2011, pp. 948–959.
- [30] R. Moussalli, M. Salloum, W. Najjar, and V. Tsotras, Accelerating XML query matching through custom stack generation on FPGAs, in *High Performance Embedded Architectures and Compilers*, ser. Lecture Notes in Computer Science, Y. Patt, P. Foglia, E. Duesterwald, P. Faraboschi, and X. Martorell, Eds. Springer Berlin Heidelberg, 2010, vol. 5952, pp. 141–155.
- [31] R. Moussalli, M. Salloum, R. Halstead, W. Najjar, and V. J. Tsotras, A study on parallelizing XML path filtering using accelerators, *ACM Trans. Embed. Comput. Syst.*, vol. 13, no. 4, pp. 93:1–93:28, Mar. 2014.
- [32] R. Mueller, J. Teubner, and G. Alonso. Data processing on FPGAs. *Proceedings of the VLDB Endowment*, 2(1):910–921, Aug. 2009.
- [33] R. Mueller, J. Teubner, and G. Alonso. Streams on wires: a query compiler for FPGAs. *Proceedings of the VLDB Endowment*, 2(1):229–240, Aug. 2009.
- [34] R. Mueller and J. Teubner, FPGAs: A new point in the database design space, in *Proceedings of the 13th International Conference on Extending Database Technology*, ser. EDBT '10. New York, NY, USA: ACM, 2010, pp. 721–723.
- [35] R. Mueller, J. Teubner, and G. Alonso, Sorting networks on FPGAs, *VLDB J.*, vol. 21, no. 1, pp. 1–23, 2012.
- [36] R. Mueller, J. Teubner, and G. Alonso. Glacier: a query-to-hardware compiler. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, SIGMOD '10, pages 1159–1162, Indianapolis, Indiana, USA, 2010. ACM.
- [37] M. Najafi, M. Sadoghi, and H.-A. Jacobsen. Flexible query processor on FPGAs. *Proc. VLDB Endow.*, 6(12):1310–1313, Aug. 2013.
- [38] M. Najafi, M. Sadoghi, and H. Jacobsen, Configurable hardware-based streaming architecture using online programmable-blocks, in *31st IEEE International Conference on Data Engineering, ICDE 2015, Seoul, South Korea, April 13-17, 2015*, IEEE, 2015, pp. 819–830.
- [39] M. Najafi, M. Sadoghi, and H.-A. Jacobsen, The FQP vision: Flexible query processing on a reconfigurable computing fabric, *SIGMOD Record*, vol. 44, no. 2, pp. 5–10, Aug. 2015.
- [40] Y. Oge, T. Miyoshi, H. Kawashima, and T. Yoshinaga, A fast handshake join implementation on fpga with adaptive merging network, in *Proceedings of the 25th International Conference on Scientific and Statistical Database Management*, ser. SSDBM. New York, NY, USA: ACM, 2013, pp. 44:1–44:4.
- [41] K. Pagiamtzis and A. Sheikholeslami. Content-addressable memory (CAM) circuits and architectures: A tutorial and survey. *IEEE JOURNAL OF SOLID-STATE CIRCUITS*, 41(3):712–727, 2006.
- [42] O. Polychroniou and K. A. Ross, High throughput heavy hitter aggregation for modern SIMD processors, in *Proceedings of the Ninth International Workshop on Data Management on New Hardware, DaMoN 1013, New York, NY, USA, June 24, 2013*, 2013, p. 6.
- [43] O. Polychroniou and K. A. Ross, Vectorized bloom filters for advanced SIMD processors, in *Tenth International Workshop on Data Management on New Hardware, DaMoN 2014, Snowbird, UT, USA, June 23, 2014*, 2014, pp. 6:1–6:6.
- [44] O. Polychroniou, A. Raghavan, and K. A. Ross, Rethinking SIMD vectorization for in-memory databases, in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, Victoria, Australia, May 31 - June 4, 2015*, 2015, pp. 1493–1508.
- [45] E. Protalinski. NEC develops memory that stores data without using power. *Techspot*, June 2011.
- [46] P. Roy, J. Teubner, and R. Gemulla, Low-latency handshake join, *PVLDB*, vol. 7, no. 9, pp. 709–720, 2014.
- [47] M. Sadoghi, M. Labrecque, H. Singh, W. Shum, and H.-A. Jacobsen, Efficient event processing through reconfigurable hardware for algorithmic trading, *Proc. VLDB Endow.*, vol. 3, no. 1-2, pp. 1525–1528, Sep. 2010.
- [48] M. Sadoghi, H. Singh, and H.-A. Jacobsen. fpga-TopSS: line-speed event processing on FPGAs. In *Proceedings of the 5th ACM international conference on Distributed event-based system, DEBS '11*, pages 373–374, New York, New York, USA, 2011. ACM.
- [49] M. Sadoghi, H. Singh, and H.-A. Jacobsen. Towards highly parallel event processing through reconfigurable hardware. In *Proceedings of the Seventh International Workshop on Data Management on New Hardware, DaMoN '11*, pages 27–32, Athens, Greece, 2011. ACM.
- [50] M. Sadoghi, R. Javed, N. Tarafdar, H. Singh, R. Palaniappan, and H.-A. Jacobsen, Multi-query stream processing on FPGAs, in *Proceedings of the 2012 IEEE 28th International Conference on Data Engineering*, Washington, DC, USA: IEEE Computer Society, 2012, pp. 1229–1232.
- [51] M. Sadoghi, An efficient, extensible, hardware-aware indexing kernel, Ph.D. Dissertation, University of Toronto, 2013.
- [52] M. Sadoghi and H. Jacobsen, Adaptive parallel compressed event matching, in *IEEE 30th International Conference on Data Engineering, Chicago, ICDE 2014, IL, USA, March 31 - April 4, 2014*, IEEE, 2014, pp. 364–375.
- [53] B. Salami, O. Arcas-Abella, and N. Sönmez, HATCH: hash table caching in hardware for efficient relational join on FPGA, in *23rd IEEE Annual International Symposium on Field-Programmable Custom Computing Machines, FCCM 2015, Vancouver, BC, Canada, May 2-6, 2015*, 2015, p. 163.
- [54] N. Satish, C. Kim, J. Chhugani, A. D. Nguyen, V. W. Lee, D. Kim, and P. Dubey, Fast sort on CPUs and GPUs: A case for bandwidth oblivious SIMD sort, in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '10. New York, NY, USA: ACM, 2010, pp. 351–362.

- [55] S. Shukla, R. M. Rabbah, and M. Vorbach. FPGA-based combined architecture for stream categorization and intrusion detection. In *MEMOCODE'10*, pages 77–80. IEEE Computer Society, 2010.
- [56] R. Sidhu and V. K. Prasanna. Fast regular expression matching using FPGAs. In *Proceedings of the the 9th Annual IEEE Symposium on Field-Programmable Custom Computing Machines, FCCM '01*, pages 227–238, Washington, DC, USA, 2001. IEEE Computer Society.
- [57] B. Sukhwani, H. Min, M. Thoennes, P. Dube, B. Iyer, B. Brezzo, D. Dillenberger, and S. Asaad. Database analytics acceleration using FPGAs, in *Proceedings of the 21st International Conference on Parallel Architectures and Compilation Techniques*, ser. PACT '12. New York, NY, USA: ACM, 2012, p. 411–420.
- [58] E. A. Sitaridi and K. A. Ross. Ameliorating memory contention of OLAP operators on GPU processors, in *Proceedings of the Eighth International Workshop on Data Management on New Hardware, DaMoN 2012, Scottsdale, AZ, USA, May 21, 2012*, 2012, pp. 39–47.
- [59] E. A. Sitaridi and K. A. Ross. Optimizing select conditions on GPUs, in *Proceedings of the Ninth International Workshop on Data Management on New Hardware, DaMoN 1013, New York, NY, USA, June 24, 2013*, 2013, p. 4.
- [60] B. Sukhwani, H. Min, M. Thoennes, P. Dube, B. Iyer, B. Brezzo, D. Dillenberger, and S. Asaad. Database analytics acceleration using FPGAs. In *Proceedings of the 21st International Conference on Parallel Architectures and Compilation Techniques*, PACT '12, pages 411–420, New York, NY, USA, 2012. ACM.
- [61] J. Teubner and R. Mueller. How soccer players would do stream joins. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, SIGMOD '11, pages 625–636, Athens, Greece, 2011. ACM.
- [62] J. Teubner, R. Müller, and G. Alonso. FPGA acceleration for the frequent item problem. In *Proceedings of the 26th International Conference on Data Engineering, ICDE'10*, pages 669–680, Long Beach, California, USA, 2010. IEEE.
- [63] J. Teubner, L. Woods, and C. Nie. Skeleton automata for FPGAs: reconfiguring without reconstructing, in *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2012, Scottsdale, AZ, USA, May 20-24, 2012*, 2012, pp. 229–240.
- [64] J. Teubner and L. Woods. *Data Processing on FPGAs*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2013.
- [65] A. Tumeo, O. Villa, and D. Sciuto. Efficient pattern matching on GPUs for intrusion detection systems. In *Proceedings of the 7th ACM international conference on Computing frontiers*, CF '10, pages 87–88, Bertinoro, Italy, 2010. ACM.
- [66] J. Turley. Intel's stellarton mixes CPU and FPGA. 2010.
- [67] T. Willhalm, N. Popovici, Y. Boshmaf, H. Plattner, A. Zeier, and J. Schaffner. SIMD-scan: Ultra fast in-memory table scan using on-chip vector processing units, *Proc. VLDB Endow.*, vol. 2, no. 1, pp. 385–394, Aug. 2009.
- [68] L. Woods, J. Teubner, and G. Alonso. Complex event detection at wire speed with FPGAs. *Proceedings of the VLDB Endowment*, 3(1-2):660–669, Sept. 2010.
- [69] L. Woods, J. Teubner, and G. Alonso. Less watts, more performance: An intelligent storage engine for data appliances, in *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '13. New York, NY, USA: ACM, 2013, pp. 1073–1076.
- [70] L. Woods, Z. István, and G. Alonso. Ibex - an intelligent storage engine with support for advanced SQL off-loading, *PVLDB*, vol. 7, no. 11, pp. 963–974, 2014.
- [71] H. Wu, G. Diamos, T. Sheard, M. Aref, S. Baxter, M. Garland, and S. Yalamanchili. Red fox: An execution environment for relational query processing on GPUs. In *Proceedings of Annual IEEE/ACM International Symposium on Code Generation and Optimization, CGO '14*, pages 44:44–44:54, New York, NY, USA, 2014. ACM.
- [72] L. Wu, L. Andrea, P. Timothy K., M. A. Kim, and K. A. Ross. Q100: The Architecture and Design of a Database Processing Unit. *Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS '14*, pages 255–268, Salt Lake City, Utah, USA, March 2014. ACM.
- [73] L. Wu, A. Lottarini, T. K. Paine, M. A. Kim, and K. A. Ross. The Q100 database processing unit, *IEEE Micro*, vol. 35, no. 3, pp. 34–46, 2015.
- [74] J. Yang and J. Goodman. Symmetric key cryptography on modern graphics hardware. In *Proceedings of the Advances in Cryptology 13th International Conference on Theory and Application of Cryptology and Information Security, ASIACRYPT'07*, pages 249–264, Berlin, Heidelberg, 2007. Springer-Verlag.
- [75] S. Zhang, J. He, B. He, and M. Lu. OmniDB: Towards portable and efficient query processing on parallel CPU/GPU architectures, *Proc. VLDB Endow.*, vol. 6, no. 12, pp. 1374–1377, Aug. 2013.
- [76] K. Zhang, K. Wang, Y. Yuan, L. Guo, R. Lee, and X. Zhang. Mega-KV: A case for GPUs to maximize the throughput of in-memory key-value stores, *Proc. VLDB Endow.*, vol. 8, no. 11, pp. 1226–1237, Jul. 2015.

BIOGRAPHIES

Dr. Rajesh Bordawekar is a research staff member at the IBM T. J. Watson Research Center at Yorktown Heights, NY. Rajesh Bordawekar received his M.S and Ph.D. in Computer Engineering in 1993 and 1996, respectively. He was a post-doctoral fellow at the Center of Advanced Computing Research, California Institute of Technology, where he evaluated out-of-core algorithms on supercomputers. At IBM Research, Rajesh has worked on various projects which include Java Runtime and Compiler, XPath Processing and Parallelization, and Transactional Memory. Rajesh has published extensively in scientific journals and conferences. He has over 40 publications and 14 issued patents. For a past few years, he has been working on optimizing a variety of business analytics and data management problems on multi-core processors. He has presented a tutorial on *Analyzing Analytics for Parallelism* at PPOPP, ASPLOS, and ISCA conferences. Rajesh has been co-organizing an annual workshop co-located with the VLDB conferences on Accelerating Data Management Workloads (ADMS) (www.adms-conf.org). Rajesh served on the program committees of VLDB 2013 (Research) and VLDB 2014 (Research and Industrial tracks). His work on analyzing analytics workloads was recently published as a Synthesis Lecture in Computer Architecture (November 2015).

Dr. Mohammad Sadoghi joined IBM T.J. Watson Research Center after receiving his Ph.D. from the Computer Science department at the University of Toronto in 2013. Dr. Sadoghi's research spans all facets of massive-scale data management that now demand a careful re-examination in light of Big Data, an explosion that is partly fueled by the Internet of Things (IoT). Yet his ultimate vision lies in fundamentally rethinking the foundation of relational databases for future hardware and computing platforms, i.e., cloud, by re-imagining the query, transaction, and storage models in order to sustain the unprecedented scale of data proliferation and heterogeneity observed in the Big Data era. Dr. Sadoghi has over 30 publications in the leading database conferences and journals (e.g., SIGMOD, VLDB, ICDE, TODS, and TKDE) and has filed 23 U.S. patents. His SIGMOD'11 paper was awarded the EPTS Innovative Principles Award, and his EDBT'11 paper was selected as one of the best EDBT papers in 2011. Currently, he is the publicity co-chair of ACM DEBS; has served in the program committee of SIGMOD, VLDB, ICDE, EDBT, IJCAI, ICDCS, ECOOP, ICSOC, DEBS, and ADMS; and has been invited reviewers for TKDE, TSC, IS, and AAAI.