THE 2014 TOP PICKS IN COMPUTER ARCHITECTURE

Luis Ceze
University of Washington

Karin Strauss
Microsoft Research

It is our pleasure to introduce this special issue featuring the 2014 top picks in computer architecture. Our goal is to bring the reader up to date on the state of the art in computer architecture research. Selected papers have the potential to start new research directions, motivate the industry to pursue new ideas, and show analyses and frameworks that shed light on computer design.

This issue presents 12 articles that describe novel, exciting research directions in areas as diverse as reconfigurable logic, hardware acceleration, design for correctness and verification, nonvolatile memories, high-throughput computing, and programmability enhancing frameworks. For the first time, this edition also recognizes a selection of honorable mention papers (see the "Honorable Mentions" sidebar), work which the selection committee felt deserved broad attention, but which unfortunately could not be included in this issue.

The review process

We received 110 submissions. A 28-member committee (see the sidebar "The Selection Committee") reviewed all submissions. Each paper received at least five reviews from this committee. In cases where one committee chair had a conflict of interest with a submission, the other chair handled the review process. In cases where both had a conflict, the review process was handled by Mary Jane Irwin, who had no submissions and no conflicts with any of these papers. In addition to the Selection Committee reviews, we also sought external reviews for unique cases where we felt specific outside expertise was needed. Papers with high variance in scores were also targeted for additional online discussion and, in some cases, additional reviews. We thank Ms. Irwin, the committee, and the external reviewers for their time and effort toward this valuable service to the computer architecture community.

This year, we also offered a community input option. Authors who, for any reason, did not want to participate in this process were able to opt out. We had a 20 percent opt-out rate. The remaining submissions were posted on a separate site, where anyone with an institutional email address could register and submit reviews. The chairs conveyed only significant positive reviews to the committee via online comments in the website used exclusively by the Selection Committee. Unfortunately, this year we did not have very broad community participation, and the little participation we had (15 reviewers submitting 51 reviews) did not affect the selection of any paper, because many of the reviews were for papers that would have been selected by the committee anyway. However, we received positive feedback from the community that the idea had value in allowing graduate students to practice writing reviews.

We selected 51 top-ranked papers for discussion at the PC meeting in Seattle in January 2015, using a combination of the average overall merit score and the variance of scores for each paper. Committee members with conflicts left the room before papers were discussed. The meeting was conducted in four phases. In the first phase, the committee summarized and discussed all 51 papers, but no

0272-1732/15/$31.00 © 2015 IEEE Published by the IEEE Computer Society

Honorable Mentions

In addition to the 12 Top Picks papers, we selected 12 Honorable Mention papers that the committee believes could be of high interest to the readers, but that space limitations kept us from including in this issue. These papers also cover exciting research directions in areas as diverse as networks on chip, soft error detection and containment, memory virtualization and protection models, cache coherence and memory consistency, new parallel execution models, prefetching, and principled secure hardware design. Table A lists and briefly describes the papers, which we encourage *IEEE Micro* readers to explore.

Table A. The 2014 honorable mentions in computer architecture.

Title of paper	Authors	Conference	Summary
"NoC Architecture for Silicon Interposer Systems"	Natalie Enright Jerger, Ajaykumar Kannan, Zimo Li, and Gabriel H. Loh	MICRO 2014	Future highly integrated systems will use a combination of both silicon-interposer (2.5D) and vertical (3D) stacking technologies to integrate computing chips with multiple stacks of memory. This paper addresses a key research challenge for such systems: how to design an effective interconnection architecture to tie together the multiple silicon chips and effectively exploit the additional routing resources of the silicon interposer.
"Tangle: Route-Oriented Dynamic Voltage Minimization for Variation-Afflicted, Energy-Efficient On-Chip Networks"	Asit Mishra, Jianping Xu, and Josep Torrellas	HPCA 2014	Tangle saves energy in on-chip networks under process variation by reducing the supply voltage guardband. A controller monitors the (corrected) errors of messages as they traverse the network and, based on this, it dynamically decreases or increases the voltage of groups of network routers.
"Avoiding Core's DUE & SDC via Acoustic Wave Detectors and Tailored Error Containment and Recovery"	Gaurang Upasani, Xavier Vera, and Antonio Gonzalez	ISCA 2014	This paper presents a novel architecture based on acoustic wave detectors that can eliminate the SDC and DUE failures by efficiently detecting, containing, and recovering all soft errors. The proposed architecture can protect the unstructured, inherently complex and irregular processor cores, including the latches, combinational logic, and other unprotected elements in the pipeline and the on-chip caches with very little hardware overhead and less than 1 percent performance cost.
"Efficient Memory Virtualization"	Jayneel Gandhi, Arkaprava Basu, Mark D. Hill, and Michael M. Swift	MICRO 2014	Two key trends in computing are evident—the emergence of big-data applications and the prevalence of virtual machines in the era of cloud computing. This paper enables big-data applications to efficiently access large amounts of low-locality data while executing on virtual machines through a novel combination of segmentation and paging.
"The CHERI Capability Model: Revisiting RISC in an Age of Risk"	Jonathan Woodruff, Robert N.M. Watson, David Chisnall, Simon W. Moore, Jonathan Anderson, Brooks Davis, Ben Laurie, Peter G. Neumann, Robert Norton, and Michael Roe	ISCA 2014	This paper presents a capability-based model for memory safety that scales from object-granularity protection to library-scale compartments while allowing legacy binaries to run unmodified, providing an incremental adoption path to increased reliability and security. The paper describes a complete-stack evaluation with an FPGA-based implementation running a modified FreeBSD operating system and LLVM-based C compiler, along with a detailed comparison to existing state-of-the-art memory-safety techniques.

(*Continued*)

Title of paper	Authors	Conference	Summary
"Heterogeneous-Race-Free Memory Models"	Derek R. Hower, Blake A. Hechtman, Bradford M. Beckmann, Benedict R. Gaster, Mark D. Hill, Steven K. Reinhardt, and David A. Wood	ASPLOS 2014	The heterogeneous-race-free (HRF) class of memory consistency models is the first to robustly describe synchronization semantics of heterogeneous memory systems that use coherency domains called scopes. HRF has been immediately influential in the community, and has already been adopted as the memory consistency framework of the industry-wide heterogeneous system architecture (HSA) standard.
"High-Performance Fractal Coherence"	Gwendolyn Voskuilen and T.N. Vijaykumar	ASPLOS 2014	Coherence protocols are notoriously error-prone, but commercial protocols are rarely verified because, to date, verification has not been scalable; and fractal coherence, which is scalably verifiable, degrades performance scalability. This paper proposes the FlatFractal protocol, which achieves both scalable performance comparable to a conventional directory protocol and scalable verifiability.
"ASC: Automatically Scalable Computation"	Amos Waterland, Elaine Angelino, Ryan P. Adams, Jonathan Appavoo, and Margo Seltzer	ASPLOS 2014	This paper presents a radical approach to automatically parallelizing programs. It shows that for some programs it's possible to predict full computational states far in the future, speculatively execute forward from those predictions in parallel with the current execution, and then reuse the computations that turned out to be seeded from correct predictions.
"HELIX-RC: An Architecture-Compiler Co-Design for Automatic Parallelization of Irregular Programs"	Simone Campanoni, Kevin Brownell, Svilen Kanev, Timothy M. Jones, Gu-Yeon Wei, and David Brooks	ISCA 2014	HELIX-RC shows how to accelerate non-numerical programs by automatically extracting parallelism from small loops. This is enabled by a proactive, cache-based, low-latency, broadcast interconnect between cores codesigned with a compiler.
"Sandbox Prefetching: Safe, Run-Time Evaluation of Aggressive Prefetchers"	Seth Pugsley, Zeshan Chishti, Chris Wilkerson, Peng-fei Chuang, Robert Scott, Aamer Jaleel, Shih-Lien Lu, Kingsum Chow, and Rajeev Balasubramonian	HPCA 2014	Sandbox Prefetching introduces a new prefetching paradigm that separates pattern confirmation from prefetch action by evaluating inherently risky prefetching mechanisms in a safe hardware sandbox. The compact sandbox mechanism is used to decide which simple prefetcher to deploy in the real memory hierarchy, and then applies the chosen prefetcher to every subsequent memory access without further analysis or pattern confirmation, achieving high performance using only very simple mechanisms.
"Data-Parallel Finite-State Machines"	Todd Mytkowicz and Madanlal Musuvathi	ASPLOS 2014	This paper describes a new method for breaking dependences in sequential computation and applies them to efficiently parallelize finite-state machines. The key idea is to efficiently enumerate, rather than speculate, all values for the data dependence. This paper demonstrates that using such an enumerative approach, finite-state machine processing can exploit various sources of hardware parallelism, including instruction-level parallelism, multicore, and single-instruction, multiple data.
"Sapper: A Language for Hardware-Level Security Policy Enforcement"	Xun Li, Vineeth Kashyap, Jason Oberg, Mohit Tiwari, Vasanth Ram, Ryan Kastner, Tim Sherwood, Ben Hardekopf, and Fred Chong	ASPLOS 2014	Sapper is a hardware description language that enables the design of hardware with low-overhead, provable information-flow guarantees. The Sapper compiler uses a combination of static analysis, dynamic types, and runtime checks to dramatically reduce overhead from 2× in previous approaches to a few percent in area and power.

decision was made. The goal of this phase was to bring the entire committee up to speed on every paper being discussed and to conclude pending online discussions. The second phase discussed papers for which the outcome of the online discussion was to accept, and a preliminary decision was made to determine whether each of these papers should be a Top Pick or an Honorable Mention. All papers in this category were selected as a preliminary Top Pick (pending discussion in the fourth phase). Next, the committee discussed the remaining papers, voting on whether the paper should be a Top Pick and, if not, voting whether the paper should be an Honorable Mention. The outcome of the second and third phases was 12 Top Picks and 12 Honorable Mentions. The fourth phase addressed whether any of these papers needed to be swapped from Top Pick to Honorable Mention or vice versa to make sure the 12 Top Picks were the most highly regarded 12 papers in the group. No rearrangement was necessary. We congratulate the authors of Top Picks and Honorable Mention papers on this well-deserved accolade.

The selected articles

The selected articles address many of the issues we face today or expect to face in the future. The end of Dennard scaling and the increasing difficulties in further CMOS scaling have led to the need for higher-efficiency systems. Some of the selected articles tackle this issue by proposing datacenter-scale reconfigurable architectures and various special-purpose processors, as well as a framework to quickly evaluate the accelerator design space. A second set has focused on the challenge of multicore memory systems design and verification, tackling both cache coherence and memory consistency. A third set covers increases in throughput and reductions in latency with various microarchitectural techniques. Finally, two articles tackle programmers' difficulties in developing software on emerging hardware by providing frameworks and abstractions to reduce complexity and simplify the process, with special emphasis on dealing with memory nonvolatility and variable uncertainty.

On improving efficiency via reconfigurable logic and specialization, "A Reconfigurable Fabric for Accelerating Large-Scale Datacenter Services" by Andrew Putnam et al. addresses the problem of specialization at the datacenter scale by creating a specialization fabric with a real-world system based on field-programmable gate arrays connected by a custom fast network, and demonstrates its benefits on Web services by accelerating search. The recent trend of using deep neural networks for speech and image recognition motivated "A High-Throughput Neural Network Accelerator" by Tianshi Chen et al. The authors propose a highly optimized accelerator that carefully orchestrates data movement and

couples that with efficient computing engines for neural network evaluation. And given the importance of databases as a workload, "The Q100 Database Processing Unit" by Lisa Wu et al. introduces an accelerator for analytic operations on relational tables and columns. In a less traditional direction, in "Race Logic: Abusing Hardware Race Conditions to Perform Useful Computation," Advait Madhavan et al. propose to use signal races in a clever way to allow operations such as min/max to be made nearly free. To round up the accelerator story, "The Aladdin Approach to Accelerator Design and Modeling" by Yakun Sophia Shao et al. describes a framework that simplifies assessing early designs of new accelerators.

Memory systems are a core piece of computer designs. Given the prevalence of multicore designs and the emergence of new memory technologies, it is no wonder that this issue includes strong work in this space. "Verifying Correct Microarchitectural Enforcement of Memory Consistency Models" by Daniel Lustig et al. proposes to use microarchitecture-level analysis to verify memory consistency models. This lets designers pinpoint cases in which a microarchitecture fails to enforce the memory consistency orderings promised at the architectural level. Following the verification theme, in "PVCoherence: Designing Flat Coherence Protocols for Scalable Verification," Meng Zhang et al. describe a methodology for coherence protocol design that facilitates formal verification of the protocol. Once designed and verified, multicore memory systems must divide shared resources among applications and cores. To do so, "Sharing Incentives and Fair Division for Multiprocessors" by Seyed Majid Zahedi and Benjamin C. Lee redefines fairness and uses economic game theory to manage hardware resource assignments. Giving an accelerator access to memory using virtual addresses greatly simplifies writing code. To this end, "Address Translation for Throughput-Oriented Accelerators" by Bharath Pichai et al. shows how to make it fast by codesigning the memory management unit with the execution unit. And when the memory system can't keep up with processor demand, the new microarchitecture proposed by Arthur Perais and André Seznec in "EOLE: Toward a Practical Implementation of Value Prediction" promises to improve performance through value prediction while limiting implementation complexity by reducing the out-of-order engine aggressiveness.

New ways to reason about properties in computer systems are important for many reasons, such as writing code or designing new systems. Along these lines, in "Memory Persistency: Semantics for Byte-Addressable Nonvolatile Memory Technologies," Steven Pelley et al. propose a model analogous to memory consistency but applied to persistence. Finally, given the pervasiveness of uncertain data from sensors and approximate computing trends, "Uncertain<T>: Abstractions for Uncertain Hardware and Software" by James Bornholt et al. proposes to make uncertainty a first-class citizen in programming languages and discusses its implications on hardware design.

We hope that you enjoy reading these articles, as well as their original conference versions, and we welcome your feedback on this issue.

MICRO

Acknowledgments

We thank Erik Altman and Lieven Eeckhout. As editors in chief, they provided us support and direction on difficult decisions. We thank the Web chair, Benjamin Ransford, for setting up and making sure both websites ran smoothly and glitch-free. We thank Mary Jane Irwin for handling papers with conflicts. Finally, we thank our excellent selection committee, the reviewers who diligently evaluated all papers, and the authors for their submissions, short summaries, and hard work in producing the final versions of their papers for this issue.

Luis Ceze is the Torode Family Associate Professor in the Department of Computer Science and Engineering at the University of Washington. Contact him at luisceze@cs.washington.edu.

Karin Strauss is a researcher at Microsoft Research and affiliate faculty in the Department of Computer Science and Engineering at the University of Washington. Contact her at kstrauss@microsoft.com.