DEPARTMENT: Awards

Iterative Modulo Scheduling

2016 MICRO Test of Time Award Winner

Onur Mutlu
ETH Zurich

Scott Mahlke
University of Michigan

Tom Conte
Georgia Tech

Wen-mei Hwu
University of Illinois

The International Symposium on Microarchitecture (MICRO) recognizes the most influential papers published in past MICRO conferences that have had a significant impact on the field with its annual Test of Time (ToT) award.[1] Testimonials and retrospectives for a large subset of the award-winning papers have appeared in past *IEEE Micro* issues[2] (and you can find more information on procedures, calls for nominations, and past winners at microarch.org/Micro-ToT-Award/).

In 2016, the MICRO ToT award was given to Dr. B. Ramakrishna (Bob) Rau's seminal paper entitled "Iterative Modulo Scheduling: An Algorithm for Software Pipelining Loops," which appeared at MICRO 1994, 22 years before receiving the award.[3] Bob had an immense impact on the MICRO conference over the decades due to his technical contributions and service. Unfortunately, he passed away in 2002 and was neither able to accept the award nor write the retrospective of his work.

We provide the following brief testimonial for this seminal paper to recognize its tremendous impact.

BACKGROUND: SOFTWARE PIPELINING

Back in the late 1980s and 1990s, software pipelining was one of the hottest topics at the MICRO conference, with countless different approaches published each year. Many researchers focused on this problem due to the emergence of very long instruction word (VLIW)[4] multicomputers like Multiflow[5] and Cydra-5.[6] Simple VLIW hardware was used to construct wide-issue processors that were dependent on the compiler to produce efficient instruction schedules that could uncover enough instruction-level parallelism to maximize the hardware's performance. Software pipelining was particularly effective because it focused on program hotspots, namely loops, and it was capable of hiding long arithmetic and memory latencies that were common in these high-performance designs.

Published by the IEEE Computer Society
0272-1732/18/$33.00 ©2018 IEEE

ITERATIVE MODULO SCHEDULING

Iterative modulo scheduling, or IMS, was the culmination of Bob's nearly two decades of work on what was originally called polycyclic scheduling[7] (and for which Bob's original paper in MICRO 1981 received one of the inaugural 2014 MICRO ToT awards[1,2]). After polycyclic scheduling, it was called software pipelining—which is the name that has stuck in the broader community—and finally modulo scheduling. IMS was the perfect combination of technical elegance and engineering excellence, which are the true hallmarks of Bob himself. IMS solves the complex problem of overlapping the execution of multiple iterations of a loop as simply generating the schedule for just a single iteration under a constraint that the schedule will repeat itself in a fixed number of cycles.

This simplification brought sanity to the chaotic world of software pipelining that had traditionally thought in terms of multiple loop iterations. IMS with hardware support for predicated execution eliminated all the code expansion of traditional software pipelining. The pipeline is filled and drained, and then it executes arbitrary numbers of iterations with just a single copy of every instruction in the loop. The scheme also allows optional loop unrolling before and after to address practical needs such as function unit utilization and overlapping register life times without the complexity in traditional multi-iteration methods. The beauty of the approach is the mathematical formulation of starting with bounds on the maximum throughput the loop could achieve, and backing away until a near-optimal solution is found. The code generation schema was extremely systematic, enabling rigorous implementation and testing that allowed it to be part of many commercial compilers. Such rigor is a rare find in the complex world of compiler back-ends and made IMS the *de facto* way to implement software pipelining.

IMS stands the test of time because it is widely in use 20 years after being published and will continue to be for the foreseeable future. It translated software pipelining from a research concept to an engineering reality that could be implemented in production compilers.

TECHNICAL IMPACT

IMS was a central part of the original Cydra-5 compiler, one of the best strengths of the Itanium compilers built by Intel and Hewlett-Packard, and is also standard in any modern VLIW DSP compiler from companies like Texas Instruments and STMicroelectronics. IMS stands the test of time because it is widely in use 20 years after being published and will continue to be for the foreseeable future. It translated software pipelining from a research concept to an engineering reality that could be implemented in production compilers.

BOB RAU

Bob passed away December 10, 2002, but his memory lives on in all of our hearts—not only because of his unparalleled technical excellence but also because he was a true gentleman who treated even the most novice graduate students with kindness and respect. He contributed greatly to the transition of MICRO from a workshop to a full-fledged ACM/IEEE symposium in 1992. He was an integral part of the MICRO community for many years and someone who will never be forgotten.

We also note that Bob wrote two other papers[7,8] on related topics that received the MICRO ToT award in the past.[1,2] We encourage especially the young readers of *IEEE Micro* to enjoy Bob's seminal work by reading his original papers.

REFERENCES

1. O. Mutlu and R. Belgard, “Introducing the MICRO Test of Time Awards: Concept, Process, 2014 Winners, and the Future,” *IEEE Micro*, vol. 35, no. 2, 2015, pp. 85–87; computer.org/csdl/mags/mi/2015/02/mmi2015020085.html.
2. O. Mutlu et al., “The 2014 MICRO Test of Time Award Winners: From 1978 to 1992,” *IEEE Micro*, vol. 36, no. 1, 2016, pp. 60–c3; computer.org/csdl/mags/mi/2016/01/mmi2016010060-abs.html.
3. B.R. Rau, “Iterative modulo scheduling: an algorithm for software pipelining loops,” *Proceedings of the 27th annual international symposium on Microarchitecture*, 1994, pp. 63–74; dl.acm.org/citation.cfm?id=192731.
4. J.A. Fisher, “Very Long Instruction Word architectures and the ELI-512,” *ISCA '83 Proceedings of the 10th annual international symposium on Computer architecture*, 1983, pp. 140–150; dl.acm.org/citation.cfm?id=801649.
5. R.P. Colwell et al., “A VLIW Architecture for a Trace Scheduling Compiler,” *IEEE Transactions on Computers*, vol. 37, no. 8, 1988, pp. 967–979; computer.org/csdl/trans/tc/1988/08/t0967.pdf.
6. B.R. Rau et al., “The Cydra 5 departmental supercomputer: design philosophies, decisions, and trade-offs,” *Computer*, vol. 22, no. 1, 1989, pp. 12–35; ieeexplore.ieee.org/document/19820/.
7. B.R. Rau and C.D. Glaeser, “Some scheduling techniques and an easily schedulable horizontal architecture for high performance scientific computing,” *Proceedings of the 14th annual workshop on Microprogramming*, 1981, pp. 183–198; dl.acm.org/citation.cfm?id=802449.
8. B.R. Rau, M.S. Schlansker, and P.P. Tirumalai, “Code generation schema for modulo scheduled loops,” *Proceedings of the 25th annual international symposium on Microarchitecture*, 1992, pp. 158–169; dl.acm.org/citation.cfm?id=145795.

ABOUT THE AUTHORS

Onur Mutlu is a professor of computer science at ETH Zurich and a faculty member at Carnegie Mellon University. His research interests are in computer architecture, computer systems, and bioinformatics. He has a PhD from the University of Texas at Austin. Contact him at onur.mutlu@inf.ethz.ch.

Scott Mahlke is a professor in the Electrical Engineering and Computer Science department at the University of Michigan, where he directs the Compilers Creating Custom Processors research group. His research interests include application-specific processors and accelerators, compiler optimization, and computer architecture. Mahlke has a PhD from the University of Illinois at Urbana-Champaign. Contact him at mahlke@umich.edu.

Tom Conte is a professor of computer science and of electrical and computer engineering at the Georgia Institute of Technology. His research interests include novel computing architectures, parallel computing, and neuromorphic computing. Conte has a PhD in electrical engineering from the University of Illinois at Urbana-Champaign. Contact him at tom@conte.us.

Wen-mei Hwu is a professor in the Electrical and Computer Engineering department, the chief scientist of the Parallel Computing Institute, and the director of the IMPACT research group at the University of Illinois at Urbana-Champaign. His primary research area is parallel processing. Hwu has a PhD in computer science from the University of California, Berkeley. Contact him at w-hwu@illinois.edu.