BenchCouncil Achievements Evaluation Report

Lieven Eeckhout

Certificate Number: BCAER-000001
Date of issue: November 28, 2023

BenchCouncil: International Open Benchmark Council

Lieven Eeckhout’s Achievement Evaluation Report

Short Bio

Lieven Eeckhout is a full professor in the Department of Electronics and Information Systems at Ghent University, Belgium. He received his PhD degree from Ghent University in 2002. Lieven Eeckhout's research focuses on computer architecture, particularly performance evaluation, modeling, and workload characterization. As of November 24th, 2023, Lieven Eeckhout has 12345 citations on Google Scholar.

Key contributions

Lieven’s novel contributions lie in the workload characterization of metrologies and tools. His main contributions are as follows:

- Sniper Multi-Core Simulation
- Java Performance Analysis
- Performance evaluation and benchmarking

1. Sniper Multi-Core Simulation

Simulators are critical in evaluating and optimizing computer systems, providing researchers and developers with a controlled environment to test and experiment with various configurations and scenarios. One such simulator is the Sniper simulator, an x86 simulator capable of performing timing simulations for multi-program workloads and multi-threaded, shared-memory applications with 10s to 100+ cores. The Sniper simulator is based on the interval core model and the Graphite simulation infrastructure, which enables it to deliver fast and precise simulations.

[Publication]

1) Carlson, Trevor E., Wim Heirman, and Lieven Eeckhout. "Sniper: Exploring the level of abstraction for scalable and accurate parallel multi-core simulation." In Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 1-12. 2011. **Cited by 1072**.
2) Trevor E. Carlson, Wim Heirman, Stijn Eyerman, Ibrahim Hur, Lieven Eeckhout. An Evaluation of High-Level Mechanistic Core Models. ACM Transactions on Architecture and Code Optimization. 11(3): 28:1-28:25. 2014. **Cited by 366.**
3) Genbrugge, Davy, Stijn Eyerman, and Lieven Eeckhout. "Interval simulation: Raising the level of abstraction in architectural simulation." HPCA-16 2010 The Sixteenth International Symposium on High-Performance Computer Architecture. IEEE, 2010. **Cited by 186.**
4) Heirman, Wim, et al. "Power-aware multi-core simulation for early design stage hardware/software co-optimization." Proceedings of the 21st international conference on Parallel architectures and compilation techniques. 2012. **Cited by 54.**

[Influence]

Sniper is a robust open-source tool that can be found on GitHub with 51 stars and 26 forks, and on DockerHub with 211 pulls. It is built on the X86 ISA and offers support for ARM ISA, with initial support for RISC-V. Researchers and developers regularly employ Sniper in their publications, and it has been referenced in several studies. Some influential publications that have cited or used Sniper include:

1) Sanchez, Daniel, and Christos Kozyrakis. "ZSim: Fast and accurate microarchitectural simulation of thousand-core systems." ACM SIGARCH Computer architecture news 41.3 (2013): 475-486. **Cited by 665.**

2) Li, Shuangchen, et al. "Pinatubo: A processing-in-memory Architecture for bulk bitwise operations in emerging non-volatile memories." Proceedings of the 53rd Annual Design Automation Conference. 2016. **Cited by 464.**

3) Aga, Shaizeen, et al. "Compute caches." 2017 IEEE International Symposium on High Performance Computer Architecture (HPCA). IEEE, 2017. **Cited by 355.**

4) Bharathwaj Raghunathan, Yatish Turakhia, Siddharth Garg and Diana Marculescu. Cherry-Picking: Exploiting Process Variations in Dark-Silicon Homogeneous Chip Multi-Processors. IEEE/ACM Design, Automation, and Test in Europe Conference (DATE 2013). **Cited by 132.**

5) Yatish Turakhia, Bharathwaj Raghunathan, Siddharth Garg and Diana Marculescu. HaDeS: Architectural Synthesis for Heterogeneous Dark Silicon Chip Multi-processors. ACM/IEEE Design Automation Conference (DAC 2013). **Cited by 73.**

2. Performance evaluation and benchmarking methods

Performance evaluation and benchmarking are the fundamentals of system design and optimization. The main contributions include program similarity analysis, microarchitecture-independent workload characterization, system-level performance metrics, performance models for superscalar out-of-order processors, and performance prediction.

[Publication]

1) Eyerman, Stijn, and Lieven Eeckhout. "System-level performance metrics for multiprogram workloads." IEEE Micro 28.3 (2008): 42-53. **Cited by 508.**

2) Van Craeynest, Kenzo, et al. "Scheduling heterogeneous multi-cores through performance impact estimation (PIE)." 2012 39th Annual International Symposium on Computer Architecture (ISCA). **Cited by 437.**

3) Hoste, Kenneth, and Lieven Eeckhout. "Microarchitecture- independent workload characterization." IEEE Micro 27.3 (2007): 63-72. **Cited by 248.**

4) Eyerman, Stijn, et al. "A mechanistic performance model for superscalar out-of-order processors." ACM Transactions on Computer Systems (TOCS) 27.2 (2009): 1-37. **Cited by 236.**

5) Lieven Eeckhout, Hans Vandierendonck, Koenraad De Bosschere. Workload Design: Selecting Representative Program-Input Pairs. IEEE PACT 2002: 83-94. **Cited by 154.**

[Influence]

As an expert in performance evaluation and benchmarking, he co-authored the book "Performance Evaluation and Benchmarking," which has been widely referenced in the industry and academia. His outstanding contributions in this field have earned him numerous accolades, including the prestigious 2017 Maurice Wilkes Award for his exceptional work in computer architecture performance analysis and modeling. In addition, his 2007 paper "A Top-Down Approach to Architecting CPI Component Performance Counters" was selected as an IEEE Micro Top Pick for its significant contribution to the field of computer architecture and its relevance to the industry. Some influential publications that have cited his work include:

1) Che, Shuai, et al. "Rodinia: A benchmark suite for heterogeneous computing." 2009 IEEE international symposium on workload characterization (IISWC). IEEE, 2009. **Cited by 3548.**

2) Delimitrou, Christina, and Christos Kozyrakis. "Paragon: QoS-aware scheduling for heterogeneous datacenters." ACM SIGPLAN Notices 48.4 (2013): 77-88. **Cited by 943.**

3) Zhuravlev, Sergey, Sergey Blagodurov, and Alexandra Fedorova. "Addressing shared resource contention in multicore processors via scheduling." ACM Sigplan Notices 45.3 (2010): 129-142. **Cited by 825.**

4) Moscibroda, Thomas, and Onur Mutlu. "A case for bufferless routing in on-chip networks." Proceedings of the 36th annual international symposium on Computer architecture. 2009. **Cited by 553.**

5) Kim, Yoongu, et al. "ATLAS: A scalable and high-performance scheduling algorithm for multiple memory controllers." HPCA-16 2010 The Sixteenth International Symposium on High-Performance Computer Architecture. IEEE, 2010. **Cited by 553.**

3. Java Performance Analysis

Java performance evaluation is a complex process that is influenced by several factors, including the Java application, virtual machine, garbage collector, heap size, and non-determinism at run-time. There are various methodologies available for Java performance evaluation, and they differ in how they report performance. Lieven has proposed a series of works that focus on Java performance analysis, such as Statistically rigorous Java performance evaluation, Analyzing Java programs at the microarchitectural level, Method-level phase behavior in Java workloads, Java performance evaluation through rigorous replay compilation, and customized Java program analysis tools. These works emphasize the importance of rigorous data analysis and provide valuable insights into the performance of Java applications.

[Publication]

1) Georges, Andy, Dries Buytaert, and Lieven Eeckhout. "Statistically rigorous Java performance evaluation." In Proceedings of the 22nd annual ACM SIGPLAN conference on Object-oriented programming systems, languages and applications (OOPSLA '07). **Cited by 786**.

2) Eeckhout, Lieven, Andy Georges, and Koen De Bosschere. "How Java programs interact with virtual machines at the microarchitectural level." In Proceedings of the 18th annual ACM SIGPLAN conference on Object-oriented programming systems, languages and applications (OOPSLA '03). **Cited by 118.**

3) Georges, Andy, et al. "Method-level phase behavior in Java workloads." In Proceedings of the 19th annual ACM SIGPLAN conference on Object-oriented programming systems, languages and applications (OOPSLA '04). **Cited by 92.**

4) Andy Georges, Lieven Eeckhout, Dries Buytaert. Java performance evaluation through rigorous replay compilation. In Proceedings of the 23rd annual ACM SIGPLAN conference on Object-oriented programming systems, languages and applications (OOPSLA '08). **Cited by 65.**

[Influence]

His work in this field has been widely recognized and cited in numerous studies. In fact, it was awarded the 2017 OOPSLA Most Influential Paper Award for its statistically rigorous approach to evaluating Java performance presented at OOPSLA 2007. This recognition highlights the importance of accurate performance evaluation in software development and underscores the impact of the paper on the field of computer science. Some influential publications that have cited his work are as follows:

1) Blackburn, Stephen M., et al. "The DaCapo benchmarks: Java benchmarking development and analysis." Proceedings of the 21st annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications. 2006. **Cited by 1924.**

2) Blackburn, Stephen M., Perry Cheng, and Kathryn S. McKinley. "Myths and realities: The performance impact of garbage collection." ACM SIGMETRICS Performance Evaluation Review 32.1 (2004): 25-36. **Cited by 483.**

3) Mytkowicz, Todd, et al. "Producing wrong data without doing anything obviously wrong!." ACM Sigplan Notices 44.3 (2009): 265-276. **Cited by 433.**

4) Jones, Richard, Antony Hosking, and Eliot Moss. The garbage collection handbook: the art of automatic memory management. CRC Press, 2023. **Cited by 400.**

5) Blackburn, Stephen M., Perry Cheng, and Kathryn S. McKinley. "Myths and realities: The performance impact of garbage collection." ACM SIGMETRICS Performance Evaluation Review 32.1 (2004): 25-36. **Cited by 323.**