15-745 Optimizing Compilers, Spring 2019
Papers for In-Class Presentation and Discussion

Pointer Analysis

- Osbert Bastani, Rahul Sharma, Alex Aiken, and Percy Liang. *"Active learning of points-to specifications"* in *Proceedings of the 39th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '18)*, June 2018.
- Yulei Sui, Peng Di, and Jingling Xue. *"Sparse flow-sensitive pointer analysis for multithreaded programs,"* in *Proceedings of the 2016 International Symposium on Code Generation and Optimization (CGO '16)*, March 2016.
- Vitor Paisante, Maroua Mallei, Leonardo Barbosa, Laure Gonnord, and Fernando Magno Quintao Periera. *"Symbolc range analysis of pointers"* in *Proceedings of the 2016 International Symposium on Code Generation and Optimization (CGO '16)*, March 2016.

Cache Optimizations

- Probir Roy, Shuaiwen Leon Song, Sriram Krishnamoorthy, and Xu Liu. *"Lightweight detection of cache conflicts,"* in *Proceedings of the 2018 International Symposium on Code Generation and Optimization (CGO '18)*, February 2018.
- Jonathan Lifflander and Sriram Krishnamoorthy. *"Cache locality optimization for recursive programs,"* in *Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '17)*, June 2017.
- Changwan Hong, Wenlei Bao, Albert Cohen, Sriram Krishnamoorthy, Louis-Noel Poucher, Fabrice Rastello, J. Ramanujam, and P. Sadayappan. *"Effective padding of multidimensional arrays to avoid cache conflict misses,"* in *Proceedings of the 37th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '16)*, June 2016.

Optimizations for Dynamic Languages (e.g., JavaScript)

- Guilherme Ottoni. *"HHVM JIT: a profile-guided, region-based compiler for PHP and Hack,"* in *Proceedings of the 39th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '18)*, June 2018.
- Thomas Wrthinger, Christian Wimmer, Christian Humer, Andreas W, Lukas Stadler, Chris Seaton, Gilles Duboscq, Doug Simon, and Matthias Grimmer. *"Practical partial evaluation for high-performance dynamic language runtimes,"* in *Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '17)*, June 2017.
- Gem Dot, Alejandro Martinez, and Antonio Gonzalez. *"Removing checks in dynamically typed languages through efficient profiling,"* in *Proceedings of the 2017 International Symposium on Code Generation and Optimization (CGO '17)*, February 2017.

Dynamic Optimization

- David Devecsery, Peter M. Chen, Jason Flinn, and Satish Narayanasamy. *"Optimistic Hybrid Analysis: Accelerating Dynamic Analysis through Predicated Static Analysis,"* in *Proceedings of the 23rd International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '18)*, March 2018.

- David Leopoldseder, Lukas Stadler, Thomas Wrthinger, Josef Eisl, Doug Simon, and Hanspeter Mssenbck. *"Dominance-based duplication simulation (DBDS): code duplication to enable compiler optimizations* in *Proceedings of the 2018 International Symposium on Code Generation and Optimization (CGO '18)*, February 2018.
- Daniele Cono D'Elia and Camil Demetrescu. *"Flexible on-stack replacement in LLVM,"* in *Proceedings of the 2016 International Symposium on Code Generation and Optimization (CGO '16)*, March 2016.

Compiler-Assisted Debugging

- Bozhen Liu, and Jeff Huang. *"D4: fast concurrency debugging with parallel differential analysis,"* in *Proceedings of the 39th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '18)*, June 2018.
- Jake Roemer, Kaan Gen, and Michael D. Bond. *"High-coverage, unbounded sound predictive race detection,"* in *Proceedings of the 39th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '18)*, June 2018.
- Anirudh Santhiar and Aditya Kanade. *"Static deadlock detection for asynchronous C# programs,"* in *Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '17)*, June 2017.

Compiling for GPUs

- Xiang Gong, Zhongliang Chen, Amir Kavyan Ziabari, Rafael Ubal, and David Kaeli. *"TwinKernels: An execution model to improve GPU hardware scheduling at compile time,"* in *Proceedings of the 2017 International Symposium on Code Generation and Optimization (CGO '17)*, February 2017.
- Michal Steuwer, Toomas Remmelg, and Christophe Dubach. *"Lift: A functional data-parallel IR for high-performance GPU code generation,"* in *Proceedings of the 2017 International Symposium on Code Generation and Optimization (CGO '17)*, February 2017.
- Jingyue Wu, Artem Belevich, Eli Bendersky, Mark Heffernan, Chris Leary, Jacques Pienaar, Bjarke Roune, Rob Springer, Xuetian Wang. and Robert Hundt. *"gpucc: An open cource GPGPU compiler,"* in *Proceedings of the 2016 International Symposium on Code Generation and Optimization (CGO '16)*, March 2016.

Compiling for Vectors

- Vasileios Porpodas, Rodrigo C. O. Rocha, and Luis F. W. Góes. *"Look-ahead SLP: Auto-vectorization in the presence of commutative operations,"* in *Proceedings of the 2018 International Symposium on Code Generation and Optimization (CGO '18)*, February 2018.
- Peng Jiang and Gagan Agrawal. *"Conflict-free vectorization of associative irregular applications with recent SIMD architectural advances,"* in *Proceedings of the 2018 International Symposium on Code Generation and Optimization (CGO '18)*, February 2018.
- Linchuan Chen, Peng Jiang, and Gagan Agrawal. *"Exploiting recent SIMD architectural advances for irregular applications,"* in *Proceedings of the 2016 International Symposium on Code Generation and Optimization (CGO '16)*, March 2016.

Inter-procedural Analysis

- Teresa Johnson, Mehdi Amini, and Xianliang David Li. *"ThinLTO: Scalable and incremental LTO,"* in *Proceedings of the 2017 International Symposium on Code Generation and Optimization (CGO '17)*, February 2017.
- Patrick W. Sathyanathan, Wenlei He, and Ten H. Tzen. *"Incremental whole program optimization and compilation,"* in *Proceedings of the 2017 International Symposium on Code Generation and Optimization (CGO '17)*, February 2017.

- Mateus Tymburiba, Rubens E. A. Moreira, and Fernando Magno Quintao Pereira. *"Inference of peak density of indirect branches to detect ROP attacks,"* in *Proceedings of the 2016 International Symposium on Code Generation and Optimization (CGO '16)*, March 2016.

Portability

- William F. Ogilvie, Pavlos Petoumenos, Zheng Wang, and Hugh Leather. *"Minimizing the cost of iterative computation with active learning,"* in *Proceedings of the 2017 International Symposium on Code Generation and Optimization (CGO '17)*, February 2017.
- Saurav Muralidharan, Amit Roy, Mary Hall, Michael Garland, and Piyush Rai. *"Architecture-adaptive code variant tuning,"* in *Proceedings of the 21st International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '16)*, April 2016.
- Toshihiko Koju, Reid Copeland, Motohiro Kawahito, and Moriyoshi Ohara. *"Re-constructing high-level information for language-specific binary re-optimization,"* in *Proceedings of the 2016 International Symposium on Code Generation and Optimization (CGO '16)*, March 2016.

Compilers for AI

- Tianqi Chen, Thierry Moreau, Ziheng Jiang, Lianmin Zheng, Eddie Yan, Haichen Shen, Meghan Cowan et al. *"TVM: An Automated End-to-End Optimizing Compiler for Deep Learning,"* in Proceeding of the 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI '18), October 2018.
- Samyam Rajbhandari, Yuxiong He, Olatunji Ruwase, Michael Carbin, and Trishul Chilimbi. *"Optimizing CNNs on Multicores for Scalability, Performance and Goodput,"* in *Proceedings of the Twenty-Second International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '17)*, April 2017.
- Leonard Truong, Rajkishore Barik, Ehsan Totoni, Hai Liu, Chick Markley, Armando Fox, and Tatiana Shpeisman. *"Latte: a language, compiler, and runtime for elegant and efficient deep neural networks,"* in *Proceedings of the 37th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '16)*, June 2016.

Compiler Support for Intermittent Computing

- Kiwan Maeng, and Brandon Lucia. *"Adaptive dynamic checkpointing for safe efficient intermittent computing,"* in Proceesing of the 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI '18), October 2018.
- Kiwan Maeng, Alexei Colin, and Brandon Lucia. *"Alpaca: intermittent execution without checkpoints,"* in Proceeding of the ACM on Programming Languages (OOPSLA '17), October 2017.
- Joel Van Der Woude, and Matthew Hicks. *"Intermittent computation without hardware support or programmer intervention,"* in Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16), November 2016.