

Compiler-assisted Correctness Checking and Performance Optimization for HPC

(C3PO'20)

URL: <https://c3po-workshop.github.io>

ISC Topic Area: Programming Models & System Software

ISC Keywords:

Compilers
Heterogeneous Systems
Parallel Applications
Performance Analysis and Optimization
Programming Models & Languages

Topics of interest:

- Automatic parallelization
- Compilation for heterogeneous systems
- Compiler enabled optimization
- Correctness checking of parallel software
- Cross-languages translation
- Modernization of legacy codes
- Source code instrumentation
- Source-to-source analysis/transformations
- Static analysis
- Static/dynamic interaction

Abstract

Practical compiler-enabled programming environments, applied analysis methodologies, and end-to-end toolchains can contribute significantly to performance portability in the exascale era. The practical and applied use of compilation techniques, methods, and technologies, including static analysis and transformation, are imperative to improve the performance, correctness, and scalability of high-performance applications, middleware, and reusable libraries.

This workshop brings together a diverse group of researchers with a shared interest in applying compilation and source-to-source translation methodologies, among others, to enhance explicit parallel programming such as MPI, OpenMP, and hybrid models. These types of compiler technologies can also be applied to heterogeneous programming elements including FPGAs and GPUs in order to deliver higher achievable performance as compared to library-based methods and hand-written approaches taken in isolation.

Original papers will identify and solve challenges in the tradeoffs of scalability, performance, predictability, correctness, productivity, and portability on-node and at massive scale; strong-scaling, weak-scaling, and hybrid-scaling solutions assisted, augmented, and/or enabled by compiler technology are in scope. Topics of interest include but are not limited to: correctness checking of parallel programs, source-to-source translation of legacy MPI codes to improve performance-portability, instrumentation, and massively multipass FPGA compiler optimization strategies.

We recognize that there are standard static analysis technologies (dataflow analysis, polyhedral analysis, etc); this workshop seeks innovative applications of such technologies singly and in combination to derive enhanced utility in parallel programs that are generalizable beyond a single case study or narrow application.

Steering Committee:

General Chair	Peter Pirkelbauer Lawrence Livermore National Laboratory / University of Central Florida pirkelbauer2@llnl.gov +1 925 424 4569
Associate Chair	Emmanuelle Saillard Inria Bordeaux Sud-Ouest emmanuelle.saillard@inria.fr +33 5 24 57 41 24
Publications Chair	Anthony Skjellum University of Tennessee at Chattanooga Tony-Skjellum@utc.edu
Publicity Chair	Martin Ruefenacht University of Edinburgh m.ruefenacht@ed.ac.uk
Track Chairs	Purushotham Bangalore University of Alabama at Birmingham puri@uab.edu Julien Jaeger French Alternative Energies and Atomic Energy Commission (CEA) julien.jaeger@cea.fr Peter Thoman University of Innsbruck peter.thoman@uibk.ac.at

Tentative Program Committee

- Hadia Ahmed (Lawrence Berkeley National Laboratory)
- Ritu Arora (Texas Advanced Computing Center - TACC)
- Protonu Basu (Facebook)
- Ira Baxter (Semantic Design)
- Elisabeth Brunet (Telecom Sud Paris)
- Patrick Carribault (CEA)
- Thomas Fahringer (University of Innsbruck)
- Chunhua Liao (Lawrence Livermore National Laboratory)
- Reed Milewicz (Sandia National Laboratories)
- Christina Peterson (University of Central Florida)
- Amalee Wilson (Stanford University)
- Sara Royuela (BSC)
- Muite Benson (Kichakato Kizito)
- Joachim Protze (RWTH Aachen University)
- Markus Schordan (Lawrence Livermore National Laboratory)

Workshop format

Full day workshop at ISC'20.

- Invited Talk
- Full papers
- Short papers
- Position papers
- Short announcements
- Panel discussion

Timeliness and Importance

This workshop is timely and important because the changing HPC architectures, libraries, and the emergence of hybrid programming models demand novel toolchains that help application reach high utility (e.g., performance, correctness, portability). Programmers working with yesterday's toolchains will not reach high utility without the assistance of practical compilation infrastructures.

The organizers are the right group to implement this workshop because they represent a cross-section of practitioners, compiler-experts, middleware designers, and application developers in HPC.

Target Audience

Researchers with a shared interest in applying compilation and source-to-source translation methodologies to the analysis, transformation, and optimization of HPC applications.

Relevance to and Impact on ISC community

Changing HPC architecture and software stack create enormous challenges for HPC application developers that need to write performance portable code and keep existing applications up to speed. Purely manual solutions are cost prohibitive. Source-to-source translators are poised to address these challenges automatically or with user input semi-automatically.

The state-of-the art tools and techniques presented at this workshop will be of interest to many ISC attendees, in particular tool developers, middleware implementers, and HPC application developers. Together they will discuss current challenges in their respective domains, available technologies, unsolved problems, and solution ideas. The workshop will conclude by stating important research questions and giving directions for future research.

Strategies for advertising and attracting attendees

Members of the organizing and program committees will seek contributions from their regions and institutions.

The organizers will also distribute a call for papers and a call for participation on mailing lists relevant to the HPC and compiler communities, including HPC-announce, as well as lists for Clang/LLVM and ROSE developers.

Paper Submission and Review Process

C3PO solicits two kinds of papers. Full papers present novel and original research and are limited to 12 pages in Springer LNCS format. Short papers report on interesting new ideas, work with promising/early results, or empirical studies and are limited to 6 pages in Springer LNCS format.

The review process is double blind. Each submission will receive at least three reviews.

Papers must be submitted in PDF format on the submission website. The organizers will make informal proceedings available online before the conference. Authors may submit a revised version of the paper after the workshop for printed proceedings. Authors of accepted papers will be required to sign the Springer copyright form. Instructions for preparing papers for the proceedings will be emailed to authors of accepted papers.

Demonstrated Community Interest

Applying source-to-source techniques to HPC application is a very active area of research. A search on Google Scholar for papers published in the last five years using the query *(source-to-source OR transpiler) compiler (HPC OR "high-performance computing")* produces roughly 1500 results. A list with selected papers published in the last five years that cover topics relevant to this workshop is available as appendix.

Selected list of recent (past 5 years) publications within the scope of this workshop

- E. Agullo, O. Aumage, B. Bramas, O. Coulaud and S. Pitoiset, "Bridging the Gap Between OpenMP and Task-Based Runtime Systems for the Fast Multipole Method," in IEEE Transactions on Parallel and Distributed Systems, vol. 28, no. 10, pp. 2794-2807, 1 Oct. 2017.
- F. Wende, "C++ Data Layout Abstractions through Proxy Types," 2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), Rio de Janeiro, Brazil, 2019, pp. 758-767.
- C. Silvano et al., "Autotuning and adaptivity approach for energy efficient Exascale HPC systems: The ANTAREX approach," 2016 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, 2016, pp. 708-713.
- T. Nguyen, P. Cicotti, E. Bylaska, D. Quinlan and S. Baden, "Automatic translation of MPI source into a latency-tolerant, data-driven form," Journal of Parallel and Distributed Computing, Volume 106, 2017, pp. 1-13.
- J. Cabezas et al., "Automatic Parallelization of Kernels in Shared-Memory Multi-GPU Nodes," Proceedings of the 29th ACM on International Conference on Supercomputing (ICS'15), 2015, pp. 3-13.
- L. Kalms, T. Hebbeler and D. Göhringer, "Automatic OpenCL Code Generation from LLVM-IR using Polyhedral Optimization", Proceedings of the 9th Workshop and 7th Workshop on Parallel Programming and RunTime Management Techniques for Manycore Architectures and Design Tools and Architectures for Multicore Embedded Computing Platforms, 2018, pp. 45-50.
- R. Sotomayor, R. Sanchez, R. Garcia Blas, J. Fernandez and J. D. Garcia, "Automatic CPU/GPU Generation of Multi-versioned OpenCL Kernels for C++ Scientific Applications," International Journal of Parallel Programming, vol. 45, no. 2, pp. 262-282, 1 Apr. 2017.
- K. Kofler, J. J. Durillo, P. Gschwandtner and T. Fahringer, "A Region-Aware Multi-Objective Auto-Tuner for Parallel Programs," 2017 46th International Conference on Parallel Processing Workshops (ICPPW), Bristol, 2017, pp. 190-199.
- C. Liao, P.-H. Lin, M. Schordan and I. Karlin, "A Semantics-Driven Approach to Improving DataRaceBench's OpenMP Standard Coverage", In Evolving OpenMP for Evolving Architectures, Springer International Publishing, 2018, pp. 189-202.
- H. Iwashita, M. Nakao, H. Murai and M. Sato, "A Source-to-Source Translation of Coarray Fortran with MPI for High Performance," Proceedings of the International Conference on High Performance Computing in Asia-Pacific Region, 2018, pp. 86-97.
- X. Xiao, S. Hirasawa, H. Takizawa and H. Kobayashi, "An Approach to Customization of Compiler Directives for Application-Specific Code Transformations," 2014 IEEE 8th International Symposium on Embedded Multicore/Manycore SoCs, Aizu-Wakamatsu, 2014, pp. 99-106.
- J. Lidman, S. McKee, D. Quinlan and C. Liao, "An Automated Performance-Aware Approach to Reliability Transformations," European Conference on Parallel Processing, Springer International Publishing, 2014, pp. 523-534.
- S. Hukerikar, K. Teranishi, P. C. Diniz and R. F. Lucas, "An evaluation of lazy fault detection based on Adaptive Redundant Multithreading," 2014 IEEE High Performance Extreme Computing Conference (HPEC), Waltham, MA, 2014, pp. 1-6.
- J. Bosch et al., "Application Acceleration on FPGAs with OmpSs@FPGA," 2018 International Conference on Field-Programmable Technology (FPT), Naha, Okinawa, Japan, 2018, pp. 70-77.

F. Alessi, P. Thoman, G. Georgakoudis, T. Fahringer and D. Nikolopoulos, "Application-Level Energy Awareness for OpenMP", In *OpenMP: Heterogenous Execution and Data Movements*, Springer International Publishing, 2015, pp. 219–232.

C. Shen, X. Tian, D. Khaldi and B. Chapman, "Assessing One-to-One Parallelism Levels Mapping for OpenMP Offloading to GPUs," In *Proceedings of the 8th International Workshop on Programming Models and Applications for Multicores and Manycores (PMAM'17)*, 2017, pp. 68-73.

D. Majeti, K. Meel, R. Barik and V. Sarkar, "Automatic Data Layout Generation and Kernel Mapping for CPU+GPU Architectures", In *Proceedings of the 25th International Conference on Compiler Construction*, 2016, pp. 240–250.

M. Haidl et al., "PACXXv2 + RV: An LLVM-based Portable High-Performance Programming Model," In *Proceedings of the Fourth Workshop on the LLVM Compiler Infrastructure in HPC (LLVM-HPC'17)*, 2017, pp. 7:1–7:12.

J. Carretero et al., "Optimizations to Enhance Sustainability of MPI Applications," In *Proceedings of the 21st European MPI Users' Group Meeting*, 2014, pp. 145:145–145:150.

G. D. Balogh, G. R. Mudalige, I. Z. Reguly, S. F. Antao and C. Bertolli, "OP2-Clang: A Source-to-Source Translator Using Clang/LLVM LibTooling," *2018 IEEE/ACM 5th Workshop on the LLVM Compiler Infrastructure in HPC (LLVM-HPC)*, Dallas, TX, USA, 2018, pp. 59-70.

H. Ahmed, A. Skjellum and P. Pirkelbauer, "Petal Tool for Analyzing and Transforming Legacy MPI Applications", In *Languages and Compilers for Parallel Computing (LCPC'16)*, Springer International Publishing, 2016, pp. 156–170.

H. Ahmed, A. Skjellum, P. Bangalore and P. Pirkelbauer, "Transforming Blocking MPI Collectives to Non-blocking and Persistent Operations," In *Proceedings of the 24th European MPI Users' Group Meeting*, 2017, pp. 3:1–3:11.

T. Teixeira, C. Ancourt, D. Padua, and W. Gropp, William, "Locus: A System and a Language for Program Optimization," *Proceedings of the 2019 IEEE/ACM International Symposium on Code Generation and Optimization*, 2019, 217–228.

J. Kim, Y.-J. Lee, J. Park, Junggho and J. Lee, Jaejin, "Translating OpenMP Device Constructs to OpenCL Using Unnecessary Data Transfer Elimination," In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2016, pp. 51:1–51:12.

G. Stelle, W. Moses, S. Olivier and P. McCormick, "OpenMPIR: Implementing OpenMP Tasks with Tapir," In *Proceedings of the Fourth Workshop on the LLVM Compiler Infrastructure in HPC*, 2017, pp. 3:1–3:12.

P. Huchant, E. Saillard, D. Barthou and P. Carribault, "Multi-Valued Expression Analysis for Collective Checking", In *Euro-Par Conference*, pages 29–43, 2019