Ari Rasch

Einsteinstraße 62
48149 Münster, Germany
☎ *+49 (0) 251 83-32763*
✉ *a.rasch@uni-muenster.de*

University Education

since 2016 **Ph.D. studies**, *University of Münster*, Münster, Germany.

Supervisor: Prof. Sergei Gorlatch

My research focuses on compiler technologies (code generation and optimization) as well as programming language design for multi- and many-core architectures, such as GPU and CPU. My overall research goal is to provide *performance*, *portability*, and *productivity* for data-parallel computations with particular focus on computations relevant for deep learning platforms (e.g., linear algebra routines and stencil computations). To achieve my goal, I am the main designer of a holistic code *generation*, *optimization*, and *execution* approach, consisting of three major sub projects:

1. *Multi-Dimensional Homomorphisms (MDH)* – a novel algebraic formalism toward expressing and formally reasoning about data-parallel computations; in particular, this project includes the formal design and specification of a Domain-Specific Language (DSL) for expressing MDH functions, as well as the design and implementation of a compiler for this DSL – the compiler enables automatically generating code for MDHs (e.g., in CUDA, OpenMP, or OpenCL) that can be automatically optimized (auto-tuned) for state-of-the-art GPUs, CPUs, etc.
2. *Auto-Tuning Framework (ATF)* – a general-purpose auto-tuning approach that automatically optimizes parallel programs, based on numerical search techniques and optimized processes of generating, storing, and exploring the optimization spaces of modern parallel programs
3. *Host Code Abstraction (HCA)* – a high-level programming abstraction that simplifies implementing and optimizing so-called host code which is required in modern programming approaches (e.g., CUDA and OpenCL) to execute parallel code on the devices of distributed, heterogeneous systems.

since 2013 **Research Associate**, *University of Münster*, Münster, Germany.

2013 **Master of Science in computer science**, *University of Münster*, Münster, Germany, *Final grade in computer science (94%)*.

Thesis title: *A Generic, Observation-Based Notion of Non-Interference* (theoretical computer science). *Grade for thesis: excellent*

Internships

03/2021 – 05/2021 **Deep Learning Compiler Engineer Intern (3 months)**, *NVIDIA*, Redmond, WA, USA.

The goal of this internship was code generation and optimization for deep-learning computations on NVIDIA GPUs via our approach of Multi-Dimensional Homomorphisms (MDHs) and auto-tuning technologies provided by our Auto-Tuning Framework (ATF). My work on this project enables fully automatically generating optimized GPU code that achieves better performance than the state-of-the-art machine-generated and hand-optimized solutions.

Leadership Qualities

As project leader of our MDH+ATF+HCA approach, I supervise one PhD student (Richard Schulze) working on this project, and I coordinate student project seminars for more than 7 years (15 in total) – all seminars are/were directly connected to our project. Moreover, I was the main supervisor of 9 master and 16 bachelors theses which all were focused on particular aspects of our project.

Awards & Achievements

2023 *Performance Bonus* from University of Münster for *Extraordinary Achievements in Research and Teaching (2500 EUR)*

2021 *Best Research Poster Finalist* at SC (*International Conference for HPC, Networking, Storage, and Analysis*) for our work titled: *Code Generation & Optimization for Deep-Learning Computations on GPUs via Multi-Dimensional Homomorphisms*

2020 *Gold-Winner* at Student Research Competition of PACT (*International Conference on Parallel Architectures and Compilation Techniques*) for our work titled: *"`md_stencil`: High-Performance Stencil Computations on CPU and GPU via Multi-Dimensional Homomorphisms"*

Gold-Winner at Student Research Competition of CGO (*International Symposium on Code Generation and Optimization*) for our work titled: *"`md_poly`: A Performance-Portable Polyhedral Compiler Based on Multi-Dimensional Homomorphisms"*

2019 *Best Poster Award* at *PUMPS+AI (Programming and Tuning Massively Parallel Systems + Artificial Intelligence)* for our poster: *"Performance, Portability, and Productivity for Data-Parallel Applications on Multi- and Many-Core Architectures"*

2018 *IHK Price* awarded by the German Chamber of Commerce and Industry for supervised master thesis of Richard Schulze titled: *"Design and Implementation of a Performance-Portable BLAS Library Based on Multi-Dimensional Homomorphisms"*

Fundings & Grants

2022-2025 **DFG Project Funding (606.271 EUR)**, *Performance, Portability, and Productivity for Deep-Learning Computations on Multi- and Many-Core Architectures (PPP-DL).*

This project aims at achieving *Performance*, *Portability*, and *Productivity* (PPP) for Deep-Learning (DL) computations on multi- and many-core architectures (GPUs, CPUs, etc) via our approaches of Multi-Dimensional Homomorphisms (MDH) and the Auto-Tuning Framework (ATF).

2021 **HiPEAC Collaboration Grant (5.000 EUR)**, *Productive Parallel Programming via Polyhedral Techniques and Multi-Dimensional Homomorphisms*, University of Edinburgh (Host: Tobias Grosser).

This collaboration aims at combining polyhedral compilation techniques with the code generation approach of Multi-Dimensional Homomorphisms (MDH) to achieve high performance in a user productive way.

Travel Grants *I successfully applied for 14 travel grants (>14.000$):*

ACM's SRC Travel Award for CGO'20, Google grant for Google Compiler and Programming Language Summit 2019, TCHPC for SC'19, PACT'19 Student Travel Grant, SIGPLAN funding for SPLASH'19, ACM's SRC Travel Award for PLDI'19, TCPP grant for IPDPS'19, DOE grant to attend annual PPP meeting 2019, Google grant for Google Compiler and Programming Language Summit 2018, SIGHPC grant for SC'18, COLOC grant for EuroPar'18, HiPEAC Grant Summer School ACACES'18, SIGARCH HPDC'18 Student Travel Grants, TCPP grant for IPDPS'18

(Co-)Organized Events

2022 **Workshop**, *Generic Auto-Tuning Technologies for GPU Applications*, Lorentz Center (Netherlands), Ben van Werkhoven (Netherlands eScience Center), Gabriele Keller (Utrecht University), Jiří Filipovič (Masaryk University), Ari Rasch (University of Münster).

The goal of the workshop was to foster international collaboration among research groups working on auto-tuning technologies, including groups working on high-level programming languages and compilers.

Research Visits

2022 **University of Copenhagen (1 week)**, Copenhagen, Denmark, participants: Mary Hall (University of Utah), Cosmin Oancea (University of Copenhagen – Host), Ari Rasch (University of Münster), Richard Schulze (University of Münster), Denys Shabalin (Google Zurich).

This meeting was focused on discussing and designing programming abstractions for expressing low-level code optimizations.

Invited Talks

2022 Lorentz Center (Netherlands) - *Generic Auto-Tuning Technologies for GPU Applications*. Talk title: *Auto-Tuning Framework (ATF)*

2020 Google - *SIG MLIR Open Design Meeting*. Talk title: *Using MLIR for Multi-Dimensional Homomorphisms*

NVIDIA - *CUDA C++ Compiler Team Meeting*. Talk title: *Multi-Dimensional Homomorphisms and Their Implementation in OpenCL: An Algebraic Approach Toward Performance, Portability, and Productivity for Data-Parallel Computations on Multi- and Many-Core Architectures*

Publications

2023 [1] A. Rasch. "(De/Re)-Composition of Data-Parallel Computations via Multi-Dimensional Homomorphisms". In: *ACM Transactions on Programming Languages and Systems (TOPLAS)* (2023), (under review).

[2] A. Rasch, R. Schulze, D. Shabalin, A. Elster, S. Gorlatch, and M. Hall. "(De/Re)-Compositions Expressed Systematically via MDH-Based Schedules". In: *ACM SIGPLAN International Conference on Compiler Construction (CC)* (2023).

2022 [3] A. Rasch, R. Schulze, and S. Gorlatch. "Expressing Hierarchical Code Optimizations via MDH-Based Schedules". In: *Workshop on Hierarchical Parallelism for Exascale Computing (HiPar)@SC'22* (2022), (WIP paper).

2021 [4] R. Schulze, A. Rasch, and S. Gorlatch. "Code Generation & Optimization for Deep-Learning Computations on GPUs via Multi-Dimensional Homomorphisms". In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC)* (2021), (short paper).

[5] A. Rasch, R. Schulze, M. Steuwer, and S. Gorlatch. "Efficient Auto-Tuning of Parallel Programs With Interdependent Tuning Parameters via Auto-Tuning Framework ATF". In: *ACM Transactions on Architecture and Code Optimization (TACO)* (2021), (original work).

2020 [6] A. Rasch, R. Schulze, and S. Gorlatch. “md_poly: A Performance-Portable Polyhedral Compiler Based on Multi-Dimensional Homomorphisms”. In: *10th International Workshop on Polyhedral Compilation Techniques (IMPACT)* (2020), (WIP paper).

2019 [7] A. Rasch, R. Schulze, and S. Gorlatch. “Generating Portable High-Performance Code via Multi-Dimensional Homomorphisms”. In: *The 28th International Conference on Parallel Architectures and Compilation Techniques (PACT)* (2019).

[8] A. Rasch, J.Bigge, M. Wrodarczyk, R. Schulze, and S. Gorlatch. “dOCAL: high-level distributed programming with OpenCL and CUDA”. In: *The Journal of Supercomputing (JOS)* (2019).

[9] A. Rasch. “Performance, Portability, and Productivity for Data-parallel Applications on Multi- and Many-core Architectures”. In: *Proceedings Companion of the 2019 ACM SIGPLAN International Conference on Systems, Programming, Languages, and Applications: Software for Humanity (SPLASH)* (2019), (short paper).

[10] A. Rasch, R. Schulze, and S. Gorlatch. “Developing High-Performance, Portable OpenCL Code via Multi-Dimensional Homomorphisms”. In: *7th International Workshop on OpenCL (IWOCL)* (2019), (extended abstract).

2018 [11] A. Rasch, R. Schulze, M. Gorus, J. Hiller, S. Bartholomäus, and S. Gorlatch. “High-Performance Probabilistic Record Linkage via Multi-Dimensional Homomorphisms”. In: *The 34th ACM/SIGAPP Symposium On Applied Computing (SAC)* (2018).

[12] A. Rasch and S. Gorlatch. “Multi-Dimensional Homomorphisms and Their Implementation in OpenCL”. In: *International Journal of Parallel Programming (IJPP)* (2018).

[13] A. Rasch and S. Gorlatch. “ATF: A Generic, Directive-Based Auto-Tuning Framework”. In: *Concurrency and Computation: Practice and Experience (CCPE)* (2018).

[14] A. Rasch, M. Wrodarczyk, R. Schulze, and S. Gorlatch. “OCAL: An Abstraction for Host-Code Programming with OpenCL and CUDA”. In: *The 24th IEEE International Conference on Parallel and Distributed Systems (ICPADS)* (2018).

[15] A. Rasch and S. Gorlatch. “ATF: A Generic Auto-Tuning Framework”. In: *The 27th International Symposium on High-Performance Parallel and Distributed Computing (HPDC)* (2018), (short paper).

[16] A. Rasch, R. Schulze, and S. Gorlatch. “Portable Parallel Performance via Multi-Dimensional Homomorphisms”. In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC)* (2018), (short paper).

2017 [17] A. Rasch, M. Haidl, and S. Gorlatch. “ATF: A Generic Auto-Tuning Framework”. In: *The 19th IEEE International Conference on High Performance Computing and Communications (HPCC)*. 2017.

[18] M. Riemenschneider, A. Herbst, A. Rasch, S. Gorlatch, and D. Heider. “eccCL: Parallelized GPU Implementation of Ensemble Classifier Chains”. In: *BMC Bioinformatics* (2017).

Attended Academic Events

I presented our research (in form of talks and/or posters) at different conferences and events:

2023 PLDI conference (at FCRC) - *ACM SIGPLAN Conference on Programming Language Design and Implementation*, Orlando FL, USA

CC conference - *ACM SIGPLAN International Conference on Compiler Construction*, Montreal, Canada

C4ML workshop - *Compilers for Machine Learning*, Montreal, Canada

2022 HiPEAC conference - *European Forum for Experts in Computer Architecture, Programming Models, Compilers and Operating Systems for Embedded and General-Purpose Systems*, Budapest, Hungary

Lorentz Center workshop - *Generic Auto-Tuning Technologies for GPU Applications*, Leiden, Netherlands

2021 SC conference - *International Conference for High Performance Computing, Networking, Storage, and Analysis*, St. Louis MO, USA (remote participation)

HiPEAC conference - *European Forum for Experts in Computer Architecture, Programming Models, Compilers and Operating Systems for Embedded and General-Purpose Systems*, Budapest, Hungary (shifted to online event)

2020 PACT conference - *International Conference on Parallel Architectures and Compilation Techniques*, Atlanta GA, USA (shifted to online event)

ACACES summer school (organized by HiPEAC) - *Sixteenth International Summer School on Advanced Computer Architecture and Compilation for High-Performance and Embedded Systems*, Fiuggi, Italy (shifted to online event)

GTC conference - *NVIDIA GPU Technology Conference*, San Jose CA, USA (shifted to online event)

CGO conference - *International Symposium on Code Generation and Optimization*, San Diego CA, USA

C4ML workshop - *Compilers for Machine Learning*, San Diego CA, USA

IMPACT workshop - *International Workshop on Polyhedral Compilation Techniques*, Bologna, Italy

HiPEAC conference - *European Forum for Experts in Computer Architecture, Programming Models, Compilers and Operating Systems for Embedded and General-Purpose Systems*, Bologna, Italy

2019 SC conference - *International Conference for High Performance Computing, Networking, Storage, and Analysis*, Denver CO, USA

Intel HPC Developer conference, Denver CO, USA

Google Compiler and Programming Language Summit - München, Germany

SPLASH conference - *ACM SIGPLAN conference on Systems, Programming, Languages, and Applications: Software for Humanity.*, Athens, Greece

PACT conference - *International Conference on Parallel Architectures and Compilation Techniques*, Seattle WA, USA

PRACE course - *Deep Learning and GPU programming using OpenACC*, Stuttgart, Germany

PLDI conference (at FCRC) - *ACM SIGPLAN Conference on Programming Language Design and Implementation*, Phoenix AZ, USA

IPDPS conference - *International Symposium on Code Generation and Optimization*, Rio De Janeiro, Brazil

IWOCL workshop - *International Workshop on OpenCL, SYCL and SPIR-V*, Boston MA, USA

SAC conference - *ACM/SIGAPP Symposium On Applied Computing*, Limassol, Cyprus

DOE PPP meeting - *DOE Performance, Portability and Productivity Annual Meeting*, Denver CO, USA

CGO conference - *International Symposium on Code Generation and Optimization*, Washington DC, USA

2018 ICAPDS conference - *International Conference on Parallel and Distributed Systems*, Sentosa, Singapore

Google Compiler and Programming Language Summit - München, Germany

SC conference - *International Conference for High Performance Computing, Networking, Storage, and Analysis*, Dallas TX, USA

ACACES summer school (organized by HiPEAC) - *Fourteenth International Summer School on Advanced Computer Architecture and Compilation for High-Performance and Embedded Systems*, Fiuggi, Italy

HPDC conference - *International Symposium on High-Performance Parallel and Distributed Computing*, Tempe AZ, USA

PRACE course - *Performance portability for GPU application using high-level programming approaches*, Paris, France

IPDPS conference - *IEEE International Parallel & Distributed Processing Symposium*, Vancouver, Canada

Euro-Par conference - *International European Conference on Parallel and Distributed Computing*, Turin, Italy

2017 HPCC conference- *IEEE International Conference on High Performance Computing and Communications* , Bangkok, Thailand

2016 HLPP conference - *International Symposium on High-Level Parallel Programming and Applications*, Münster, Germany

2015 PRACE Course - *Advanced C++ with Focus on Software Engineering*, Stuttgart, Germany

PRACE Course - *Node-Level Performance Engineering*, Stuttgart, Germany

Service to Community

I have been active as an external reviewer for various conferences, journals, and research funding organizations including: *ACM Transactions on Architecture and Code Optimization (TACO)*, *The International Journal of Parallel Programming (IJPP)*, *Concurrency and Computation: Practice and Experience (CCPE)*, *Parallel Processing Letters (PPL)*, *Journal of Parallel and Distributed Computing (JPDC) Transactions on Cloud Computing (TCC)*, *International Parallel & Distributed Processing Symposium (IPDPS)*, and *German Research Foundation (DFG)*.

Memberships

ACM Student Member (no.: 1470797)

IEEE Student Member (no.: 94517237)

Supervised Master and Undergraduate Students

Master

Arne Wilp: *Accelerating Neural Networks using the MDH Approach*

Lukas Rosendahl: *Evaluating the MDH Approach using Benchmark Suites Parboil and Rodinia*

Sebastian Kock: *Evaluating the MDH Approach for Multi-Device Systems*

Richard Schulze: *Design and Implementation of a Performance-Portable BLAS Library via Multi-Dimensional Homomorphisms*

Timo Hoth: *Generating High-Performance Code for FFTs via Multi-Dimensional Homomorphisms*

Markus Damerau: *Implementing a Generic Auto-Tuning Framework for OpenCL*

Martin Wrodarczyk: *ECC Classification via Support Vector Machines and Multilayer Perceptron*

Alexander Herbst: *Parallelization of Ensemble Classifier Chains for GPUs*

Jan Hiller: *Probabilistic Record Linkage on GPUs*

Bachelor

Dominique Bönninghof: *Design and Implementation of a Directive-Based Code Generation Approach for Multi-Dimensional Homomorphisms*

Gabriel Borrelli: *Visualizing Multi-Layered, Multi-Dimensional Parallel Computations for Modern Processors Based on the MDH Approach*

Luis Wetzel: *Evaluating Multi-Dimensional Homomorphisms via Ensemble Classifier Chains*

Lars Hunloh: *Evaluating the MDH Approach using CUTLASS and PPCG*

Waldemar Gorus: *pyATF: Auto-Tuning Interdependent Tuning Parameters in Python*

Karl Heimes: *Design and Implementation of a Visualization Tool for MDH computations*

Luke Thienemann: *Visualizing Multi-Layered, Multi-Dimensional Parallel Computations via a Multi-Transparent Cube-Based Approach*

Moritz Tätweiler: *Evaluating the MDH Approach Based on Frameworks Kokkos, Raja, Occa, and SYCL*

Fabian Kip: *A Multi-Device OpenCL Implementation for Matrix-Vector Multiplication on Heterogeneous Systems*

Julian Bigge: *Extending the OCAL Library for Clusters*

Jan Abbing: *Evaluating GPU Caches using Matrix Multiplication*

Mirco Witte: *Design and Implementation of an Interoperability API for OpenCL and CUDA*

Michael Gomulak: *Design and Implementation of an OpenCL-to-CUDA Translator*

Felix Krull: *Evaluating the SkelCL Library using Discrete Cosine Transform and Histograms*

Kevin Gehling: *Implementing Fast Fourier Transformation in SkelCL*

Fabian Hall: *Design and Implementation of an OpenCL Compatibility API for SkelCL*

Teaching

Summer 2022 Supervisor of a student seminar: *High-Level Programming Code Generation and Optimization Approaches for Modern Processors*

Summer 2021 Supervisor of a student project: *Code Generation and Optimization for Deep-Learning Computations on Modern Processors*

Winter 2020 Supervisor of a student project: *Design and Implementation of a CUDA Backend for the Lift Compiler*

Summer 2020 Supervisor of a student project: *Code Generation and Optimization for Deep-Learning Computations on Modern Processors*

Course design and lecturer: *Introduction to Programming with C and C++*

Winter 2019 Supervisor of a student project: *Design and Implementation of CUDA Backend for the Lift Compiler*

Summer 2019 Supervisor of a student project: *Parallelizing Numerical Algorithms in C++*

Teaching assistant for the course: *Parallel Programming: Multi-Core and GPU*

Winter 2018 Supervisor of a student project: *Implementing Multi-Dimensional Homomorphisms in Low-Level Programming Models*

Course design and lecturer: *Introduction to Programming with JAVA*

Summer 2018 Supervisor of a student project: *Automatic Program Optimization via Auto-Tuning and Machine Learning*

Winter 2017 Supervisor of a student project: *Evaluating and Programming the AMD Vega Architecture*

Summer 2017 Course design and lecturer: *Introduction to Programming with C and C++*

Supervisor of a student project: *Automatic Program Optimization for Modern Many-Core Systems*

Winter 2016 Supervisor of a student project: *Design and Implementation of a Parallel Pattern Library to Simplify Programming Many-Core Systems*

Summer 2016 Supervisor of a student project: *Auto-Tuning Stencil Computations on Modern Multi- and Many-Core Architectures*

Teaching assistant for the course: *Parallel Programming: Multi-Core and GPU*

Winter 2015 Course design and lecturer: *Introduction to Programming with JAVA*

Supervisor of a student project: *GPU Realization of the C++ Specification for Parallelism*

Summer 2015 Supervisor of a student project: *Skeletons for Exascale*

Teaching assistant for the course: *Parallel Programming: Multi-Core and GPU*

Winter 2014 Supervisor of a student project: *Design and Implementation of Parallel Patterns for Modern Multi- and Many-Core Architectures in OpenCL*