Ari Rasch

Einsteinstraße 62
48149 Münster, Germany
✉ *a.rasch@uni-muenster.de*

University Education

2016-2024 **Ph.D. studies**, *University of Münster, Germany*, Final grade: **summa cum laude**,

Examiners: Sergei Gorlatch, Herbert Kuchen, Albert Cohen.

Thesis title: *Toward Performance & Portability & Productivity in Parallel Programming: A Holistic Code Generation, Optimization, and Execution Approach for Data-Parallel Computations Targeting Modern Parallel Architectures*

My research focuses on compiler design and programming languages for modern parallel architectures (including GPU and CPU). My overall research goal is to provide *performance*, *portability*, and *productivity* for data-parallel computations, with a particular focus on computations relevant to deep learning (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 for expressing and formally reasoning about data-parallel computations. This project includes the design and specifications of (formal) program representations, at different abstraction levels, e.g., to formally reason about optimizations. MDH generates fully automatically parallel code (e.g., in CUDA for GPU and OpenMP for CPU), based on auto-tuning.

2. *Auto-Tuning Framework (ATF)* – a general-purpose auto-tuning approach for fully automatically optimizing parallel programs, based on numerical search techniques and optimized processes to generating, storing, and exploring the optimization spaces of complex parallel implementations.

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.

2013-2016 **Ph.D. studies: OFERTIE Project**, *University of Münster*, Münster, Germany.
OFERTIE is a project funded by the EC FP7 programme, which aims to use *Software-Defined Networking (SDN)* approaches to improve delivery of an emerging class of distributed applications for the Future Internet known as Real-Time Online Interactive Applications (ROIA)

2013 **Diploma Degree in *Computer Science*, minor in *Mathematics*, (equivalent to combined MSc and BSc)**, *University of Münster*, Germany, **Final grade: 94%**.
Thesis: *A Generic, Observation-Based Notion of Non-Interference (theoretical computer science)*

Internships

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

The goal of this internship was code generation and optimization for deep-learning computations for NVIDIA GPUs via our approaches of *Multi-Dimensional Homomorphisms (MDHs)* and *Auto-Tuning Framework (ATF)*. My work on this project enables fully automatically generating optimized GPU code for DL computations that achieves high performance as compared to state-of-the-art machine-generated and hand-optimized solutions.

Leadership Experience

As the **project lead** of the *MDH+ATF+HCA* research programme, I define its scientific direction and long-term research strategy. The programme involves contributions from **one PhD student** (Richard Schulze) and has produced **11 Master's theses** and **17 Bachelor's theses**, whose topics were aligned with the overarching research agenda and addressed targeted sub-problems of the project. It has also served as the foundation for **15 student project seminars over seven years**, ensuring coherent thematic development and sustained student involvement. Together, these activities demonstrate experience in shaping research structures, coordinating multiple project components, and supporting productive collaboration across different levels of student expertise.

Awards & Achievements

2025 **REACH Thesis Award** from University of Münster for my *PhD Thesis*

IAI Thesis Award Nomination from University of Münster for my *PhD Thesis*

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

Gold-Winner of Student Research Competition at CGO (*ACM/IEEE International Symposium on Code Generation and Optimization*) for our work titled: *"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"*

Fundings & Grants

2025-2028 **Contributions to successful DFG Project Funding (503.813 EUR)**, *Structured Code Generation for Deep-Learning Computations via Multi-Dimensional Homomorphisms (MDH.DL)*.

This project aims to introduce a structured and formally grounded *code generation approach* for Deep Learning (DL) computations, building on the MDH formalism developed during our preceding project PPP-DL."

2022-2025 **Contributions to successful 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 16 travel grants (>20.000$):*

SC ECP Funding for SC'25, SIGPLAN PAC Funding for PLDI'24, 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

2025 **NextSilicon** — *Seminar Series*. Talk title: *Code Generation & Optimization for Data-Parallel Computations via MDH and ATF*

NVIDIA — *Codegen Community Forum*. Talk title: *MDH+ATF: Code Generation & Optimization for Deep-Learning Computations on NVIDIA GPUs*

Roofline (Cologne, Germany) — *Compiler Team*. Talk title: *MDH+ATF: Code Generation & Optimization for Deep-Learning Computations*

2022 **Lorentz Center** (Leiden, 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

2025 [1] R. Schulze, S. Gorlatch, and **Ari Rasch**. "Reduction-Aware Directive-Based Programming via Multi-Dimensional Homomorphisms". In: *International Conference for High Performance Computing, Networking, Storage and Analysis (SC Workshops)* (2025).

[2] M. Hall, C. Oancea, A. C. Elster, **Ari Rasch**, S. Joshi, A. M. Tavakkoli, and R. Schulze. "Scheduling Languages Chronology: A Past, Present, and Future Taxonomy". In: *ACM Transactions on Architecture and Code Optimization (TACO)* (2025), Original research article (no prior conference version).

[3] R. Schulze, S. Gorlatch, and **Ari Rasch**. "pyATF: Constraint-Based Auto-Tuning in Python". In: *ACM SIGPLAN International Conference on Compiler Construction (CC)* (2025).

2024 [4] **Ari Rasch**. "(De/Re)-Composition of Data-Parallel Computations via Multi-Dimensional Homomorphisms". In: *ACM Transactions on Programming Languages and Systems (TOPLAS)* (2024). 73 pages, Original research article presented at PLDI'24 (no prior conference version).

[5] **Ari Rasch**. *Full Version: (De/Re)-Composition of Data-Parallel Computations via Multi-Dimensional Homomorphisms*. 131 pages. 2024. arXiv: `2405.05118 [cs.PL]`.

2023 [6] **Ari 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 [7] **Ari 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 [8] R. Schulze, **Ari 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).

[9] **Ari 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 research article presented at HiPEAC’21 (no prior conference version).

2020 [10] **Ari 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 [11] **Ari 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).

[12] **Ari 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).

[13] **Ari 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).

[14] **Ari 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 [15] **Ari 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).

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

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

[18] **Ari 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).

[19] **Ari 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).

[20] **Ari 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 [21] **Ari 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.

[22] M. Riemenschneider, A. Herbst, **Ari 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 events:

2026 **C4ML** workshop — *Compilers for Machine Learning*, Sydney, Australia

LATHC workshop — *Languages, Architectures, and Tools for Heterogeneous Computing*, Sydney, Australia

2025 **SC** conference — *ACM/IEEE International Conference for High Performance Computing, Networking, Storage, and Analysis*, St. Louis MO, USA

PLDI conference — *ACM SIGPLAN Conference on Programming Language Design and Implementation*, Seoul, South Korea

GTC conference — *GPU Technology Conference*, San Jose CA, USA

CC conference — *ACM SIGPLAN International Conference on Compiler Construction*, Las Vegas NV, USA

C4ML workshop — *Compilers for Machine Learning*, Las Vegas NV, USA

2024 **SC** conference — *ACM/IEEE International Conference for High Performance Computing, Networking, Storage, and Analysis*, Atlanta GA, USA

PLDI conference — *ACM SIGPLAN Conference on Programming Language Design and Implementation*, Copenhagen, Denmark

EuroLLVM meeting — *European LLVM Developers' Meeting*, Vienna, Austria

CGO conference — *ACM/IEEE International Symposium on Code Generation and Optimization*, Edinburgh, Scotland

2023 **SC** conference — *ACM/IEEE International Conference for High Performance Computing, Networking, Storage, and Analysis*, Denver CO, USA

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 **SC** conference — *ACM/IEEE International Conference for High Performance Computing, Networking, Storage, and Analysis*, Dallas TX, USA

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

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

2021 **SC** conference — *ACM/IEEE 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 — *ACM/IEEE 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 — *ACM/IEEE 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 — *ACM/IEEE 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 — *ACM/IEEE 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 — *IEEE International Parallel & Distributed Processing Symposium*, 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 — *ACM/IEEE International Symposium on Code Generation and Optimization*, Washington DC, USA

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

Google Compiler and Programming Language Summit — München, Germany

SC conference — *ACM/IEEE 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 — *ACM 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

Community Service

I have served as an (external) reviewer and program committee member for various *conferences*, *journals*, and research *funding organizations*.

Journals: *ACM Transactions on Programming Languages and Systems (**TOPLAS**); ACM Transactions on Architecture and Code Optimization (**TACO**); ACM Transactions on Design Automation of Electronic Systems (**TODAES**); 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**).*

Conferences: *ACM SIGPLAN International Conference on Compiler Construction (**CC**); International Parallel & Distributed Processing Symposium (**IPDPS**); International Conference for High Performance Computing, Networking, Storage, and Analysis (**SC**) — Research and Student Research Competition (SRC) Poster Reviewer & Best Poster Committee.*

Workshops: *International Workshop on Polyhedral Compilation Techniques (**IMPACT@HiPEAC**); Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (**PMBS@SC**).*

Funding organizations: *German Research Foundation (**DFG**).*

Memberships

ACM Student Member (no.: 1470797); **IEEE** Student Member (no.: 94517237)

Co-Supervised Master's and Bachelor's Students

Master Lars Hunloh: *Design and Implementation of the MDH High-Level Representation in the MLIR Framework*

Jens Hunloh: *Design and Implementation of the MDH Low-Level Representation in the MLIR Framework*

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 Henning Ohlmeyer: *Implementation and Evaluation of a Bias Correction for ATF Chain-of-Trees in the BaCO Tuner*

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 2025 **Student Seminar**, Teaching assistant: *Selected Topics in the Field of Code Generation and Optimization for Deep Learning Computations*

Summer 2022 **Student Seminar**, Teaching assistant: *High-Level Programming Code Generation and Optimization Approaches for Modern Processors*

Summer 2021 **Student Project**, Teaching assistant: *Code Generation and Optimization for Deep-Learning Computations on Modern Processors*

Winter 2020 **Student Project**, Teaching assistant: *Design and Implementation of a CUDA Backend for the Lift Compiler*

Summer 2020 **Student Project**, Teaching assistant: *Code Generation and Optimization for Deep-Learning Computations on Modern Processors*

Course Design and Lecturer: *Introduction to Programming with C and C++*

Winter 2019 **Student Project**, Teaching assistant: *Design and Implementation of CUDA Backend for the Lift Compiler*

Summer 2019 **Student Project**, Teaching assistant: *Parallelizing Numerical Algorithms in C++*

Course, Teaching Assistant: *Parallel Programming: Multi-Core and GPU*

Winter 2018 **Student Project**, Teaching assistant: *Implementing Multi-Dimensional Homomorphisms in Low-Level Programming Models*

Course Design and Lecturer: *Introduction to Programming with JAVA*

Summer 2018 **Student Project**, Teaching assistant: *Automatic Program Optimization via Auto-Tuning and Machine Learning*

Winter 2017 **Student Project**, Teaching assistant: *Evaluating and Programming the AMD Vega Architecture*

Summer 2017 **Course Design and Lecturer**: *Introduction to Programming with C and C++*

Student Project, Teaching assistant: *Automatic Program Optimization for Modern Many-Core Systems*

Winter 2016 **Student Project**, Teaching assistant: *Design and Implementation of a Parallel Pattern Library to Simplify Programming Many-Core Systems*

Summer 2016 **Student Project**, Teaching assistant: *Auto-Tuning Stencil Computations on Modern Multi- and Many-Core Architectures*

Course, Teaching Assistant: *Parallel Programming: Multi-Core and GPU*

Winter 2015 **Course Design and Lecturer**: *Introduction to Programming with JAVA*

Student Project, Teaching assistant: *GPU Realization of the C++ Specification for Parallelism*

Summer 2015 **Student Project**, Teaching assistant: *Skeletons for Exascale*

Course, Teaching Assistant: *Parallel Programming: Multi-Core and GPU*

Winter 2014 **Student Project**, Teaching assistant: *Design and Implementation of Parallel Patterns for Modern Multi- and Many-Core Architectures in OpenCL*