Michele Scuttari

michele.scuttari@polimi.it | [LinkedIn](#) | [GitHub](#)

WORK EXPERIENCE

Ongoing PhD   Nov. 2022 – Present
PhD candidate, defending in 2026   *Milano, IT*
- Lead and main developer of the MARCO compiler: https://github.com/marco-compiler/marco
- Designed and implemented MLIR dialects, causalization and scheduling algorithms, a multithreaded runtime dispatch system, and other several optimizations. All the compiler algorithms and data structures have been developed in an array-oriented manner, giving MARCO state-of-the-art scaling capabilities for large-scale models based on differential-algebraic equations.
- The project will be presented at the LLVM Developers' Meeting in October 2025.

LLVM Contributor   August. 2022 – Present
- Upstreamed improvements to the MLIR project
- Accepted contributions: folding of chains of unrealized casts, addressing of the quadratic scaling of the bufferization infrastructure, symbols caching in Arith & Func To LLVM dialect conversion.

Teaching Assistant   Sept. 2022 – Sept. 2025
Politecnico di Milano   *Milano, IT*
- Teaching assistant for the "Informatics for aerospace engineers" course
  - Exercise, laboratory, and tutoring sessions.
  - Topics: C language
- Teaching assistant for the "Formal Languages and Compilers" course
  - Exercise, laboratory, and tutoring sessions.
  - Topics: Grammars, Automatas, Parsing, Hands-on activities on the teaching-oriented compiler ACSE

Research Fellow   Nov. 2021 – Oct. 2022
Politecnico di Milano   *Milano, IT*
- Post-graduation assignee
- Development of the MARCO compiler: full embracement of the MLIR ecosystem

EDUCATION

Politecnico di Milano   Oct. 2021
MSc Computer Science and Engineering   *Milano, IT*
- Grade: 109/110, EQF Level 7, Approximate GPA: 3.75

PUBLICATIONS

"MARCO: An Experimental High-Performance Compiler for Large-Scale Modelica Models"   Dec. 2023
- https://doi.org/10.3384/ecp204
- This paper introduces MARCO, a research compiler aimed at the efficient generation of efficient simulation code from a large-scale Modelica model. MARCO's design goals, requirements, and specifications are discussed in the paper, as well as the software architecture, the current development status, and a future development roadmap. The results of two test cases demonstrate MARCO's capability to handle non-trivial Modelica models with over 10 million equations very efficiently.

"Clever DAE: Compiler Optimizations for Digital Twins at Scale"   **Aug. 2023**

- https://doi.org/10.1145/3587135.3589945
- Modeling and simulation are fundamental activities in engineering to facilitate prototyping, verification and maintenance. Declarative modeling languages allow to simulate physical phenomena by expressing them in terms of Differential and Algebraic Equations (DAE) systems. In this paper, we focus on the problem of generating code for performing the numerical integration of the model equations, and in particular on the overhead introduced by external numerical solver libraries. We propose a novel methodology for minimizing the amount of equations which require to be solved through an external solver library, together with the number of computations that are required to computed the Jacobian matrix of the system. Through a prototype LLVM-based compiler, we demonstrate how this approach achieves a linear speed-up in simulation time with respect to the baseline.

"Efficient memory management for Modelica simulations"   **Jun. 2022**

- https://doi.org/10.4230/OASIcs.PARMA-DITAM.2022.7
- The paper explains how to improve the memory management of Modelica simulations. One aspect regards the reduction of the heap memory usage, which is obtained by modifying functions whose resulting arrays could instead be allocated on the stack by the caller. The other aspect regards the possibility of avoiding garbage collection altogether by performing all memory lifetime tracking statically. Once implemented in a prototype Modelica compiler, these techniques show an improvement of the memory management overhead of over 10 times compared to a garbage collected solution, and an improvement of 56 times compared to the production-grade compiler OpenModelica.

SKILLS & INTERESTS

- **Technologies:** C, C++, Java, MLIR, LLVM
- **Skills:** Software Engineering, High-Level Design and Coordination
- **Interests:** Automotive & Aerospace sectors, Homelabs
- **Languages**: native Italian, fluent English
- **Driving License**: B (Italian)