# Ettore Speziale

## Work experience

October 2014 – Present
**GPGPU Compiler Engineer**, *Apple Inc.*, Cupertino, United States of America.
Working in the GPUSW organization, that provides the graphics stack for all Apple products. Responsible of working on the Metal frontend and toolchain, that targets an LLVM-IR based abstract virtual machine – AIR. Main achievements includes:
**Compiler Driver/Frontend/Code Generation**
Productized the Metal compiler after initial release. [Packed] vectors support. Indirect argument buffers. Address spaces support. OpenCL compiler and runtime library for Apple Silicon.
**Standard Library**
Optimized Metal standard library headers – included by default when a Metal program is compiled online – to spedup I/O and parsing time.
**Support Libraries**
Develop a library to read/write AIR metadata. Taught the LLVM object layer about the AIR object format (MetalLib). TableGen-based AIR functions library.
**Toolchain**
Developed a linker and a static archiver targeting the AIR virtual machine. Linker is able to emit both stand-alone executables and dynamic libraries. Taught LLVM binary utils about the MetalLib object format.
**Runtime Support** Function constants support for online specialization of precompiled GPGPU programs. Loader for AIR executables, enabling remapping of Metal/OpenGL/OpenCL binding points – e.g. allows to run OpenGL/OpenCL programs on Apple silicon, that does not natively support as many binding points as non-Apple Macs.
**Atomics**
Extended C++11 memory model to address spaces and execution scopes (thread, simdgroup, threadgroup, device). Implemented standard library support, code genearation, and testing.

April 2013 – September 2014
**OpenCL Toolchain Engineer**, *ARM Ltd.*, Cambridge, United Kingdom.
Working in the Media Processing Division on the OpenCL runtime for the Mali GPGPUs. Core tasks include maintaining release branches, performance evaluation, design, implementation and testing of runtime support for speculative compiler optimizations, and optimization of the GPGPU payload construction process.

June 2011 – October 2011, October 2012 – March 2013
**Research Internship**, *Barcelona Supercomputing Center*, Barcelona, Spain.
Hosted by Programming Models group (Professor Eduard Ayguadé). My assigned task was to extend the Open Source Nanox project: 1) introducing a new back-end suitable for executing OpenCL applications, 2) improving host/device data-transfer protocol, 3) re-engineering of the cluster back-end to enhance the support for distributed applications.

## Teaching activity

January 2012 – June 2012
*Teaching assistant for the* **Code Optimization and Transformation**, *M.Sc. course at Politecnico di Milano*, Milano (MI), Italy.
Compiler middle-end analysis and optimizations. Introduction to LLVM compiler internals. Goal is to teach students how to implement compiler analyses and optimizations using the LLVM framework.

*234 Escuela Avenue #13 – Mountain View, CA 94040 USA*
✉ *speziale.ettore@gmail.com* • *speziale-ettore.github.com*

January 2012 – June 2012
*Teaching assistant for the* **Programming Languages Principles**, *M.Sc. course at Politecnico di Milano*, Milano (MI), Italy.
Introduction to features implemented by mainstream languages, including inheritance, static/dynamic/type polymorphism and operator overloading. Introduction to basic memory management: free lists, pooled allocators, basic garbage collection algorithms. Introduction to parallel programming models: shared memory vs message passing paradigms. The course focuses on C++ as the language of choice.

January 2010 – January 2011
*Teaching assistant for the* **Formal Languages and Compilers**, *M.Sc. course at Politecnico di Milano*, Milano (MI), Italy.
Lectures about theory and practice of languages classification, parsing algorithms, and attribute grammars. Automatic compiler-building tools (flex and bison). Introduction to compiler internal structure: syntax-directed front-end for a C-like language and support for new language constructs.

## Education

January 2010 – December 2012
**Ph.D. studies in Computer Engineering**, *Politecnico di Milano*, Milano (MI), Italy.
Major research topics: compiler/runtime support for explicitly parallel programming models.
Minor research topics: design and development of a Linear Temporal Logic runtime verifier in Haskell.

## Ph.D. thesis

Title — Improving synchronization and data access in parallel programming models

Supervisors — Professor Stefano Crespi Reghizzi and Giovanni Agosta

Description — Automatic parallelization techniques cannot efficiently extract parallelism from a sequential application. For this reason, parallel languages are more attractive. They expose a simplified view of the parallel hardware, in order to ease the programmer writing explicitly parallel applications. Another interesting feature is the possibility to control data distribution in the parallel hardware, either explicitly (e.g. partitioned address space) or implicitly (e.g. exploiting programmer-provided hints to layout data). Moreover, these languages must handle many processing elements, leading to optimizing current synchronization primitives in order to reduce communication overhead.
The Ph.D. work aimed at analyzing inefficiencies related to the usage of parallel computing units, and to optimize them from the runtime perspective. In particular, we analyzed the optimization of reduction computations when performed together with barrier synchronizations. Moreover, we showed how runtime techniques can exploit affinity between data and computations to limit as much as possible the performance penalty hidden in NUMA architectures, both in the OpenMP and MapReduce settings. We then observed how a lightweight JIT compilation approach could enable better exploitation of parallel architectures, and lastly we analyzed the resilience to faults induction of synchronization primitives, a basic building block of all parallel programs.

## Awards

HiPEAC grant supporting the internships at Barcelona Supercomputing Center.

ST Microelectronics scholarship supporting Ph.D. studies.

## Languages

Italian — Native

English — Working knowledge