

Work experience

- April 2013 – Present **OpenCL toolchain engineer**, *ARM Ltd.*, Cambridge, United Kingdom.
Working in the Media Processing Division. My main task is to develop/maintain the OpenCL runtime for the Mali T6xx GPGUs family, in particular the component that interacts with the OpenCL compiler.
- October 2012 – March 2013 **Research internship**, *Barcelona Supercomputing Center*, Barcelona, Spain.
Hosted by Programming Models group (Professor Eduard Ayguadé). My assigned task is to optimize the Open Source project Nanox through focusing on the task-based data-flow scheduler for OpenCL applications.
- January 2010 – December 2012 **Research activity on EU FP7 2PARMA project**.
Within the PARallel PARadigms and Run-time MANagement techniques for Many-core Architectures (www.2parma.eu) European project, my role was to design and develop a prototype OpenCL compiler and runtime system for many-core architectures. Efficient assembly implementation of work-item scheduler.
- June 2011 – October 2011 **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.
- July 2010 **Research visit**, *STMicroelectronics*, Grenoble, France.
Hosted by Advanced System Technologies group (Diego Melpignano). Setup of research collaboration on OpenCL compiler/runtime technologies for many-core architectures based on LLVM/CLANG.
- September 2009 – December 2009 **Research activity on ILDJIT project**, *Politecnico di Milano*, Milano (MI), Italy.
The ILDJIT dynamic compiler, an open source implementation of the Common Language Infrastructure (ISO/IEC 23271) standard, is composed by two main components: a virtual machine and a class library. The virtual machine is mainly developed at Politecnico di Milano, while the class library is shared with the Portable .NET project. My work added to the ILDJIT virtual machine the capability of using as class library a subset of the Mono class library.

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.
- 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.
- December 2011 – January 2012 **Teaching assistant for the Software Compilers**, *M.Sc. course at Università della Svizzera italiana*, Lugano, Switzerland.
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.
- 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.

September 2009 – *Teaching assistant for the **Fundamentals of Computer Science**, M.Sc. course at Politecnico di Milano, Milano (MI), Italy.*
January 2011
Introductory laboratory course on C programming: control structures, data design, and file I/O.

Student supervision

September 2011 – **Marcello Maggioni**, *Detecting Data Access Patterns in OpenMP Parallel Loops*, Co-advisor for M.Sc. thesis.
April 2012
September 2011 – **Daniele Gianola**, *Analysis of Lock/unlock Sequences in the DaCapo Benchmark Suite*, Co-advisor for M.Sc. thesis.
July 2012
September 2010 – **Alberto Magni**, *Design and Implementation of an LLVM-based OpenCL compiler*, Co-advisor for M.Sc. thesis.
April 2011

Education and training

July 2010 **ACACES summer school, HiPEAC**, Terrassa (Barcelona), Spain.
<http://www.hipeac.net/acaces2010/>
Topics: advanced computer architecture and compilation for high-performance and embedded systems; parallel programming models and parallel programs optimization.
January 2010 – **Ph.D. studies in Computer Engineering**, *Politecnico di Milano*, Milano (MI), Italy.
December 2012 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.
Graduation expected on March 2013.
September 2006 – **Master of Science Degree in Computer Engineering**, *Politecnico di Milano*, Milano (MI), Italy.
July 2009 Main topics: design and implementation of static and dynamic (JIT) compilers, computer architecture.
September 2003 – **Bachelor of Science Degree in Computer Engineering**, *Politecnico di Milano*, Milano (MI), Italy.
March 2007 Main topics: fundamentals of software and computer engineering.

Ph.D. thesis

Title	Improving synchronization and data access in parallel programming models
Supervisors	Professor Stefano Crespi Reghizzi and Giovanni Agosta
Description	<p>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.</p> <p>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.</p>

Master thesis

Title	Multithreading support in ILDJIT dynamic compiler
-------	---

Supervisors	Professor Stefano Crespi Reghizzi and Simone Campanoni
Description	<p>The ILDJIT virtual machine is an open source implementation of the Common Language Infrastructure (ISO/IEC 23271) standard, developed at Politecnico di Milano. Thesis main contribution was augmenting ILDJIT in order to execute multi-threaded programs. Multi-threading allows to split an application into threads, that can be run in parallel on architectures that expose some kind of hardware parallelism, such as multi-processor or multi-core machines.</p> <p>The main problem involved into multi-threading support are mapping user-defined threads into operating system provided threads and implementing an efficient communication mechanism between threads.</p> <p>Inside the ILDJIT virtual machine the first problem is addressed by linking each user defined thread to an operating system thread, while the latter is resolved through the implementation of an optimized locking algorithm.</p>

Bachelor thesis

Title	NLFS: progetto di un filesystem basato sui metadati
Supervisors	Professor Pietro Braione and Marco Plebani
Description	<p>File systems usually organize data in a tree structure, in order to both provide a clear environment to the users and to efficiently support data access. However, this kind of organization does not allow classifying data in multiple classes, due to the hierarchical structure of the tree.</p> <p>NLFS is a filesystem that stores data in an unordered set. Each file can be marked with one or more labels. Such labels are organized in indexes, allowing searching files by expressing a query above the labels. With this organization, a file can be classified into multiple topics.</p>

Awards

October 2012 – December 2012	HiPEAC grant supporting the internship at Barcelona Supercomputing Center.
June 2011 – October 2011	HiPEAC grant supporting the internship at Barcelona Supercomputing Center.
January 2010 – December 2012	ST Microelectronics scholarship supporting Ph.D. studies.

Expertises

Operating systems	In-depth knowledge of Unix-like operating systems. Skills ranging from system administration to low-level programming. Experience with the GASNet fast networking library.
Software development methodologies	Experience with agile software development models. Effective user proficiency of tools for project automation, in particular GNU autotools. Effective user proficiency of distributed versioning systems: bazaar, mercurial, git. Consolidated user experience of testing automation tools: xUnit, googletest, LLVM Integrated Tester. Practical experience of administration of continuous integration servers: buildbot.
Programming languages	Detailed knowledge and extensive experience in the use of C language (12 years), C++ (3 years). Good knowledge and experience in the use of Java and C# languages. Working knowledge of Ruby and Python scripting languages. Extensive knowledge of OpenCL, from both the compiler and run-time side, including implementation details.
Compiler framework	Experience with GCC C and Fortran front-end internals. Ability to write simple GCC analysis/transformation passes over GIMPLE tuples. Detailed knowledge of LLVM and CLANG internals. Proficiency with flex, bison, and gperf tool.
Others	Consolidated user experience of \LaTeX language for scientific and technical writing, including advanced macro packages such as TikZ and PGFPlots.
Algorithms and data structures	Algorithms for concurrent and parallel programming, including fast locking/unlocking, transactional memory, and non-blocking data structures.

Open Source software development

- OpenCRun github.com/speziale-ettore/OpenCRun
Designed and developed a LLVM/CLANG based OpenCL compiler and runtime for multi-core i386/amd64 architectures.
- Nanox pm.bsc.es/projects/nanox
Implemented the backed for supporting execution of OpenCL applications. Improved the data-transfer subsystem. Re-engineered the cluster backed for better supporting execution of distributed OpenCL applications.
- ILDJIT ildjit.sourceforge.net
Implemented threading framework, including the thin locking algorithm for user-space fast lock handling. Designed and implemented a small copying garbage collector.

Languages

- Italian **Native**
English **Working knowledge**

References

- Ph.D. advisor Professor Stefano Crespi Reghizzi
Politecnico di Milano,
Dipartimento di Elettronica ed Informazione,
via Ponzio 34/5, 20133, Milano (MI), Italy
Email: crespi@elet.polimi.it
- HiPEAC host Professor Eduard Ayguadé
Barcelona Supercomputing Center,
calle Jordi Girona 31, 08034, Barcelona, Spain
Email: eduard.ayguade@bsc.es

Publications

Andrea Di Biagio, Ettore Speziale, and Giovanni Agosta. Exploiting Thread-data Affinity in OpenMP with Data Access Patterns. In *Proceedings of the 17th international conference on Parallel processing - Volume Part I*, Euro-Par'11, pages 230–241, Berlin, Heidelberg, 2011. Springer-Verlag.

Paolo Roberto Grassi, Mariagiovanna Sami, Ettore Speziale, and Michele Tartara. Analyzing the Sensitivity to Faults of Synchronization Primitives. In *Proceedings of the 2011 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems*, DFT '11, pages 349–355, Washington, DC, USA, 2011. IEEE Computer Society.

Ettore Speziale, Andrea Di Biagio, and Giovanni Agosta. An Optimized Reduction Design to Minimize Atomic Operations in Shared Memory Multiprocessors. In *Proceedings of the 2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and PhD Forum*, IPDPSW '11, pages 1300–1309, Washington, DC, USA, 2011. IEEE Computer Society.

Ettore Speziale and Michele Tartara. A Lightweight Approach to Compiling and Scheduling Highly Dynamic Parallel Programs. In *Proceedings of the Fourth USENIX conference on Hot topics in parallelism*, HotPar'12 (Poster), Berkeley, CA, USA, 2012. USENIX Association.