

MAP-I doctoral program

Advanced Parallel Computing

1. Theme, motivation and context

The architecture of efficient computing systems successfully relied for too long on advances at the microelectronics level (smaller gates and faster clock frequencies), the internal organization of the central processing element (with instruction level parallelism, including pipelining and vector computing facilities) and a balanced memory hierarchy (to hide the growing latencies to access data). All added parallelism was hidden from the programmer. However, to take advantage of further performance improvements programmers need to explicit parallelism in their code and data, or in library functions if they have not yet been adapted to the new platforms.

Parallel computing became a trend, nowadays, due to multicore processors, but the development of parallel applications using both paradigms, sharing memory among a tiny number of processes and message passing across a very large number of computing nodes, was already common practice in HPC systems at the end of last century. More recently, GPUs have become crucial in the field of machine learning due to their ability to handle large-scale computations efficiently. They significantly speed up the training of machine learning models, especially deep learning networks that involve large matrices and complex computations. Although parallel computers are very common and accessible today, currently graduated computer scientists are still lagging competences in the development of efficient applications for these environments.

Current systems exhibit many complexities, aiming to explore their performance. Sustainability issues also pushed the need for more efficient approaches to use these systems, which require a deeper knowledge of the hardware features that significantly impact the performance of code execution.

Novel computer architecture approaches have evolved in several fronts:

- devices with some or many cores, all competing for the same memory channels to access the external RAM;
- devices with hardware capabilities to support hundreds or thousands of threads, with context switch times at the clock cycle level;
- multi-level cache and a mix of private and shared cache memories, where data locality and issues such as false sharing have strong impact on performance;
- from a single memory controller per device, to multiple and shared controller per computing node, stressing again data locality on NUMA architectures;
- devices with a mix of architectures such as GPUs, TPUs and NPUs.

And devices with these mixed sets of architecture features are populating not only laptops and desktops, but also other electronics gadgets, including smartphones.

To train the current generation of computer graduates to acquire the required competences to address the development of efficient code for the current and future generations of computing systems, it is our understanding that they must acquire the following skills and competences:

- to understand how the architecture features of advanced computing systems impact the efficiency of the application under development, both at the conventional CPU design and the newer many-threaded vector computing units, typified by the novel GPU generations;
- to master the foundations of concurrency, latency, coherence, contention, speedup and so forth;
- to understand that developing applications for these new systems goes far beyond the simple concepts of parallel algorithms, namely (i) to structure data and manage their placement to complement the code parallelism and (ii) to distribute code and data among the available resources, either homogeneous or heterogeneous (computing nodes with accelerators);
- to get acquainted to programming languages and environments to develop, test and evaluate the quality and efficiency of the running code, on distinct computing environments; these should include programming with multiple threads, with message passing protocols and with CUDA based accelerators;
- to get acquainted with typical AI workload;
- to get acquainted with parallel algorithms common in many real-world applications.

2. Objectives and Learning Outcomes

Programming for multi-core and many-core architectures requires two different programming models mainly due to the memory organization. Current multi-core CPU-based devices have on-chip shared LL-cache, which supports fine- and coarse-grain shared memory parallelism. On the other hand, many-core devices usually have hundreds of simple computing cores organized in segments where memory can only be shared inside each segment. This programming model is closer to the data parallel model.

The work will be challenging and rewarding, enhancing each student's ability to work in this rapidly advancing field and giving participants experience in multi-core and many-core programming. Students will find themselves to:

- be familiar with the architecture of multi-core devices, including the multi-layer memory hierarchy and the inter-core communication facilities;
- be familiar with the architecture of many-core devices, CUDA enabled GPUs, including their internal organization and structure;
- be able to develop and evaluate the scalability of programs on shared and distributed memory architectures;
- be able to identify emergent technologies and contextualize their contributions within the larger advanced computing ecosystem;
- be familiar with parallel programming development, analysis and debugging.

3. Proposed Syllabus

- Multicore architectures:
 - SIMD and MIMD paradigms, vector processing and the memory hierarchy.
 - ARM and x86.
 - Specialized architectures (GPUs, NPUs and TPUs).
- Homogeneous and heterogeneous servers:
 - Organization of hardware accelerators.
 - NUMA, PCI-E and CXL interconnections.
 - Resource-disaggregated computing architectures.
- Supercomputers and trends in advanced computing:
 - Energy and sustainability concerns on supercomputing architectures.
 - Overview of leading large-scale supercomputers and future trends in advanced computing.
- Introduction and foundations on parallel programming:
 - Parallel programming, concurrency and parallelism, Flynn taxonomy.
 - Foster's programming methodology.
 - Major parallel programming models and paradigms.
- Programming for shared memory architectures:
 - Processes.
 - Shared memory segments and shared memory through file mapping.
 - Spinlocks, semaphores.
- Programming for distributed memory architectures:
 - MPI specification, explicit message passing, communication protocols.
 - Derived types and data packing.
 - Collective communication, communicators, topologies.
- Parallel algorithms:
 - Sorting.
 - Graph.
 - Algebra.

4. Teaching Methods and Evaluation

The overall aim is to give students a broad and well-balanced understanding of parallel computing for the diversity of devices currently available, that will serve well as a foundation for more specific work or research. By "computing", we include the issues related to "programming" and the evaluation of the resultant code & data structures in terms of efficiency and scalability. The course will include:

- lectures and discussions on the principles, technologies, experience and exploitation of these architectures;
- analysis and discussions of specific science papers and topics;
- course homework to apply and develop expertise in the course content.

The grading will consider the course work and discussions.

5. Main Bibliography

- **Parallel Programming in C with MPI and OpenMP**
Michael J. Quinn. McGraw-Hill, 2023
- **Parallel Programming for Multicore and Cluster Systems, 3rd Edition**
Thomas Rauber and Gudula Rünger, 2023
- **Programming Massively Parallel Processors: A Hands-on Approach**
D. Kirk, W. Hwu, 2nd Ed., Morgan Kaufmann, 2013
- **Computer Architecture: a Quantitative Approach**
David Patterson and John Hennessy, Morgan Kaufmann, 2017

B. Instructors' team

This team is based on 3 faculty members from U. Minho (André Pereira) and U. Porto (Jorge Barbosa and Miguel Areias).

André Martins Pereira: is an Assistant Professor at the University of Minho and an Integrated Researcher in High-Performance Computing at INESC TEC. He specialised in performance and sustainability issues on heterogeneous servers, focusing on scheduling and hardware-aware optimisations for scientific and AI-based workloads. He developed collaboration mechanisms with academia and industry, focusing on coding best practices and technological consulting for HPC systems. Previously, he was an Assistant Researcher at INESC TEC, working at the Minho Advanced Computing Center on the operation of Deucalion supercomputer. He holds a PhD in Informatics from the MAP Joint Doctoral Program, with research in automatic parallelization and scheduling for various computing architectures.

Financed Projects:

- EPICURE, European Project under the EuroHPC Joint Undertaking. 10M € funding.
- exaSIMPLE, European Project financed under the Inno4Scale innovation studies. 160k € funding.
- SustainableHPC, Fundo de Apoio à Inovação and Fundo de Eficiência Energética. 7.3M € funding.
- EUMaster4HPC, European Project under the EuroHPC Joint Undertaking. 7M € funding.

Publications:

- [1] Carlos Silva, Ricardo Vilaça, André Pereira, Ricardo Bessa, “A review on the decarbonization of high-performance computing centers”. *Renewable and Sustainable Energy Reviews* 189, Part B (2024).
- [2] Reascos, L., Carneiro, F., Pereira, A., Filipe Castro, N., & Mendes Ribeiro, R., “Berry: A code for the differentiation of Bloch wavefunctions from DFT calculations”. *Computer Physics Communications*, 295 (2024).

- [3] André Pereira, António Onofre, and Alberto Proença, “HEP-Frame: an Efficient Tool for Big Data Applications at the LHC”. *European Physical Journal Plus* 138 (2023).
- [4] André Pereira and Alberto Proença, “HEP-Frame: Improving the Efficiency of Pipelined Data Transformation & Filtering for Scientific Analyses”. *Computer Physics Communications*, Volume 263 (2021).
- [5] André Pereira, António Onofre, and Alberto Proença, “Tuning Pipelined Scientific Data Analyses for Efficient Multicore Execution”. In *Proceedings of the International Conference on High Performance Computing & Simulation*, pp. 751-758 (2016).

Jorge Manuel Gomes Barbosa: is Associate Professor at the Departamento de Engenharia Informática da Faculdade de Engenharia da Universidade do Porto (FEUP), and a research member of LIACC (Laboratório de Inteligência Artificial e Ciência de Computadores), being the research activities related to parallel computing, cloud computing, scheduling and performance modelling. He obtained his BSc degree in Computer Science from FEUP, in 1992, and his MSc degree in Digital System from the University of Manchester Institute of Science and Technology, in 1993, and the PhD in Electrical and Computer Engineering from FEUP, in 2001. From 2003 to present, he is responsible for the course “Parallel Computing” of the MSc course in Informatics at FEUP. He has (co-) authored over 60 scientific publications (including journal/conference papers and book chapters) on subjects related to heterogeneous computing, resource management, parallel computing and energy-aware scheduling. He is member of the Editorial Board of Elsevier Journal “Simulation Modelling Practice and Theory”.

Supervision of Ph.D. students:

- Hamid Arabnejad, "Optimization of multi-user job scheduling on heterogeneous platforms", MAP-I, 2016.
- Altino Sampaio, “Virtualization Management on a Scientific Cloud Computing Infrastructure”, ProDEI, FEUP, 2015.
- Daniel C. Moura, "Three-Dimensional Biplanar Reconstruction of the Scoliotic Spine for Standard Clinical Setup", ProDEI, January 2011.

Financed Projects:

- INTEGRA - DevelopINg Tailored comprEhensive services for younG migRAnTs, Project number 610243-EPP-1-2019-1-EL-EPPKA2-CBHE-JP, Budget Global: 655,520 €, FEUP: 76.485 €.
- PRACE 6th Implementation Phase Project – PRACE-6IP, Budget PT: 365 K€.
- ANTAREX - AutoTuning and Adaptivity appRoach for Energy efficient eXascale HPC systems. H2020-FET-HPC-ANTAREX-671623, total budget 3 M€.

Selected Publications:

- [1] Hamid Arabnejad, Jorge G. Barbosa, "List Scheduling Algorithm for Heterogeneous Systems by an Optimistic Cost Table", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 25, n. 3, pp.682-694, March 2014.
- [2] Hamid Arabnejad, Jorge G. Barbosa, "A Budget Constrained Scheduling Algorithm for Workflow Applications", *Journal of Grid Computing*, Vol. 12, no. 4, pp. 665-679, 2014.

- [3] Jorge G. Barbosa, Belmiro Moreira, "Dynamic scheduling of a batch of parallel task jobs on heterogeneous clusters", *Parallel Computing*, Elsevier, Vol.37 n° 8, pp.428-438, 2011.
- [4] Nishant Saurabh, Shajulin Benedict, Jorge G. Barbosa, Radu Prodan, "Expelliarmus: Semantic-centric virtual machine image management in IaaS Clouds", *Journal of Parallel and Distributed Computing*, 2020, Vol. 146, 107-121
- [5] Hamid Arabnejad, João Bispo, João M.P. Cardoso, Jorge G. Barbosa, "Source-to-source Compilation Targeting OpenMP-based Automatic Parallelization of C Applications", *The Journal of Supercomputing*, 76(9), pp. 6753-6785, 2020
- [6] Hamid Arabnejad, Jorge G. Barbosa, "Multi-QoS constrained and Profit-aware scheduling approach for concurrent workflows on heterogeneous systems", *Future Generation Computer Systems*, Elsevier, doi: 10.1016/j.future.2016.10.003, vol. 68, pp. 211-221, 2017
- [7] Hamid Arabnejad, Jorge G. Barbosa, Radu Prodan, "Low-Time Complexity Budget-Deadline Constrained Workflow Scheduling on heterogeneous resources", *Future Generation Computer Systems*, Elsevier, doi: 10.1016/j.future.2015.07.021, vol. 55, pp. 26-40, 2016

Miguel Areias is an Assistant Professor at the Department of Computer Science, Faculty of Sciences, University of Porto, Portugal and a senior researcher at the CRACS & INESC TEC research unit. He received his MSc and PhD degrees in Computer Science from the University of Porto in 2010 and 2015. He is the first author of more than twenty papers in international conferences and journals. His main research topics lie on parallelism and concurrency, in particular, on the design and implementation of concurrent algorithms, data structures and systems.

Coordination of Projects:

- EuroCC2 - National Competence Centres in the framework of EuroHPC
01-2023/12-2025
Budget: 72k

Selected Publications:

- [1] Yet Another Lock-Free Atom Table Design for Scalable Symbol Management in Prolog
Pedro Moreno, Miguel Areias, Ricardo Rocha and Vítor Santos Costa. *International Journal of Parallel Programming (IJPP)*, Springer, March 2024.
- [2] On the Correctness of a Lock-Free Compression-based Elastic Mechanism for a Hash Trie Design
Miguel Areias and Ricardo Rocha, *Computing*, Springer, May, 2022.
- [3] On the Implementation of Memory Reclamation Methods in a Lock-Free Hash Trie Design
Pedro Moreno, Miguel Areias and Ricardo Rocha. *Journal of Parallel and Distributed Computing (JPDC)*, Elsevier, May 2021.
- [4] On the Correctness and Efficiency of a Novel Lock-Free Hash Trie Map Design
Miguel Areias and Ricardo Rocha. *Journal of Parallel and Distributed Computing (JPDC)*, Elsevier. April 2021.

[5] A Compression-Based Design for Higher Throughput in a Lock-Free Hash Map
Pedro Moreno, Miguel Areias and Ricardo Rocha. International European Conference on Parallel and Distributed Computing (Euro-Par 2020), Springer, LNCS, Warsaw, Poland, August 2020.

[6] Table Space Designs For Implicit and Explicit Concurrent Tabled Evaluation
Miguel Areias and Ricardo Rocha. Journal of Theory and Practice of Logic Programming (TPLP), Cambridge University Press. July 2018.

[7] Multi-Dimensional Lock-Free Arrays for Multithreaded Mode-Directed Tabling in Prolog
Miguel Areias and Ricardo Rocha. Concurrency and Computation: Practice and Experience (CCPE), Wiley. March 2018.