MAP-I doctoral program

UCPP: Multi-core and Many-core 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, since 2005, 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 been yet adapted to the new platforms.

Parallel computing became popular with SMP systems in the 80's, where a small number of processors shared a common memory. This paradigm was later extended with the MPP systems, where large number of computer systems cooperated to run parallel applications over a distributed memory system. 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. Computer clusters at the beginning of this century inherited these features: cluster nodes are typically the SMP systems from the past, and the collection of interconnected nodes follows the distributed memory approach of the early MPP systems. Although cluster computers are very common nowadays, current graduated computer scientists are still lagging competences in the development of efficient applications for these environments. This, alone, justifies a course on Advanced Computing at graduate level.

In the past half-decade, further complexities were added into these systems, aiming to further 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:

- from a single processing element (or core) per device, and associated memory controller and channel, into some or many (thousands?) cores, all competing for the same memory channels to access the external RAM;
- from simple single-threaded cores into units with hardware capabilities to support hundreds or thousands of threads, with context switch times at the clock cycle level;
- from a single data cache level, to multi-level 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;
- from standard CPU architectures (all from the same Hennessy & Patterson school), into alternative and competitive computing accelerators, such as the NVidia GPUs and the Intel many-core x86 based Xeon Phi.

And devices with these mixed set 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 explore available environments and tools to efficiently allocate data and computing load to the available resources, using, for example, OpenACC and StarPU.

Related courses in other institutions:

1. MIT
   6.888 Parallel and Heterogeneous Computer Architecture
   http://courses.csail.mit.edu/6.888/spring13/
   Instructor: Daniel Sanchez, Joel Emer,

2. New York University
   CSCI-GA.3033-012: Multicore Processors: Architecture & Programming
   http://cs.nyu.edu/courses/fall12/CSCI-GA.3033-012/
   Instructor: Mohamed Zahran

3. Stanford University
   CS149 Parallel Computing
   http://scpd.stanford.edu/search/publicCourseSearchDetails.do?method=
   load&filter=false&courseId=8254396
   Instructor: Alex Aiken

4. The University of Utah
   CS6235: Parallel Programming for Many-Core Architectures
   http://www.cs.utah.edu/~mhall/cs6235s12/
   Instructor: Mary Hall

5. University of Missouri
   ECE8270 - Parallel Computer Architecture
   http://nps.missouri.edu/nps_wiki/index.php/ECE8270_-
   _Parallel_Computer_Architecture_%28Spring_2013%29
   Instructor: Michela Becchi

2. Objectives and Learning Outcomes

Programming for multi-core and many-core architectures require 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 course-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.

We identify two sets of interesting component parts that can help to structure course material in **multi-core and many-core computing**. Each set is covered in several lectures that close with a case study.

The work will be challenging and rewarding, enhancing each student's ability to work in this rapidly advancing field and giving participants experience to 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, both the Intel MIC and the CUDA enabled GPUs, including their internal organization and structure;
- be able to develop and evaluate the scalability of programs on multi-core and many-core architectures;
- be familiar with the emerging tools and environments for heterogeneous environments with both multi-core and many-core devices;
- be familiar with parallel program analysis and debugging tools.

3. Proposed Syllabus

1. Parallel architectures
   a. Instruction-Level Parallelism (ILP): approaches on x86 multi/many-core versus GPU-based many-core devices.
   b. Hiding memory latency: memory hierarchy on x86 multi-core and many-core devices *versus* massive multithreading on GPU many-core.
   c. Data locality: effect on processor performance of cache aware solutions.

2. Parallel Computing issues
   a. Programming model: shared memory, false sharing and cache coherence.
   b. Programming model: message passing approaches on multi-node or on computing nodes with accelerator(s).
   c. Computation models, efficiency and performance measures.
   d. Scalability analysis, isoefficiency and isogranularity functions.
   e. Methodology to develop parallel applications: problem partition, data dependencies (communication patterns), synchronization, granularity of parallelization, data locality analysis and scheduling.

3. Tools and environments for multi-core and many-core programming
   a. Programming APIs and languages for multi-core: OpenMP, Chapel, TBB (Intel Thread Building Blocks)
   b. Environments, APIs and languages for multi- and many-core: CUDA, OpenCL, Intel Parallel Studio.

4. Heterogeneous systems
   a. Development and runtime support systems: StarPU, StarSS and QUARK.
   b. Workflow representation of applications.
   c. Scheduling strategies.

4. Teaching Methods and Evaluation

The overall aim is to give students a broad and well-balanced understanding of multi-core and many-core computing 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.

Invited lectures and researchers from Univ. Texas in Austin and from other research centres will provide advanced seminars similar to the ones that have been organized at UMinho in past years, which will also support discussions on the case studies. These activities have the partial support of the UTA cooperation program with Portugal.
The grading will consider the course work and discussions (40-60%) and by a final exam (40-60%).

5. Main Bibliography

[1] J. Hennessy, D. Patterson, "Computer Architecture. A Quantitative Approach", 5th Ed., Morgan Kaufmann, 2012

[2] C. Lin, L. Snyder, "Principles of Parallel Programming", Addison Wesley, 2008

[3] M. Quinn; “Parallel programming in C and OpenMP”, McGraw Hill, 2003

[4] Chapman B., Jost G., Kuck D., Pas R.; “Using OpenMP”, The MIT Press, 2008

[5] D. Kirk, W. Hwu, "Programming Massively Parallel Processors: A Hands-on Approach", 2nd Ed., Morgan Kaufmann, 2013

[6] J. Jeffers , J. Reinders; “Intel Xeon Phi Coprocessor High Performance Programming”, Morgan Kaufmann, 2013

[7] C. Augonnet, S. Thibault, R. Namyst, and P.-A. Wacrenier.; "StarPU: A Unified Platform for Task Scheduling on Heterogeneous Multicore Architectures". Concurrency and Computation: Practice and Experience, Special Issue: Euro-Par 2009, 23:187-198, February 2011.

B. **Instructors' team**

This team is based on 2 faculty members from U. Minho (Alberto Proença) and U. Porto (Jorge Barbosa) that persistently run the course, together with other faculties from US that will cover some particular modules.

At least two professors have already agreed to give some specific courses:

- **Keshav Pingali**, Professor at the Dep. Computer Science and at the Institute for Computational Engineering and Science at Univ. Texas in Austin, W.A."Tex" Moncrief Chair of Grid and Distributed Computing; he is currently the Director of the Focus Theme on Advanced Computing at CoLab@Austin, the cooperation program between Portugal and the Univ. Texas in Austin;
- **Calvin Lin**, also a Professor at the Dep. Computer Science at Univ. Texas in Austin, has been lecturing the Doctoral Program course on Parallel Systems for the past 6 years, whose material he published as a book (Parallel Programming, see the course references). His research interests are compilers, parallel computing, and microarchitecture. Professor Calvin is well appreciated by his lecturing capabilities and has been selected for this year's Jean Holloway Award.

Experts in heterogeneous computing in Portugal, including PG students preparing their thesis, will be invited to present their research work and associated context.

Alberto José Proença (UC coordinator): Professor at Dep. Computer Science, University of Minho, since January 2001. He lead the creation of the research center CCTC presented at FCT, where he is a member. Received the MSc and PhD degrees from University of Manchester, UK, in 1979 and 1982. He is the national Director of the Advanced Computing Program at CoLab since October 2007, under the agreement between Portugal and the Univ. Texas in Austin. Current R&D interests lie on efficient heterogeneous high performance computing, namely on the exploitation of system and devices capabilities to improve the performance and efficiency of advanced scientific computing applications.

Coordination of Projects:

SeARCH: Services and Advanced Research Computing with HTC/HPC clusters
REEQ/443/EEI/2005 (Infrastructures)
Jan'05– Dec'09; Jan'10 – today (only with partners funding)
Initial Budget: €286.6k

CROSS-Fire - Collaborative Resources Online to Support Simulations on Forest Fires: a Grid Platform to Integrate Geo-referenced Web Services for Real-Time Management
GRID/GRI/81795/2006 (FCT)
Aug'07 – Aug'10
Budget: €170k

CICH: Computer Imaging in the Cultural Heritage
Contract with a non-profit US Institution and a project proposal submitted to FCT
Jun'09 – Dec'10
Budget: €20k (proposal: €200k)

Supervision of Ph.D. students:

João Barbosa, "Hardware Aware Scheduling in Heterogeneous Environments", dual PhD degree UT-Computer Science & MAP-I, 2010-2014 (co-advisor).

Selected Publications:

L.P. Santos, V. Coelho, P. Bernardes, A. Proença, "High Fidelity Walkthroughs in Archaeology Sites"; 6th Int. Symp. Virtual Reality, Archaeology and Cultural Heritage (VAST'2005); Italy, November 2005

J. Fernando, J. Sobral, A. Proenca. "JaSkel: "A Java Skeleton-Based Framework for Structured Cluster and Grid Computing", 6th IEEE Int. Symp. Cluster Computing and the Grid (CCGrid'2006), IEEE Computer Society, Singapore, May 2006

Sobral, J., Proença, A.; Enabling JaSkel Skeletons for Clusters and Computational Grids, Proc. IEEE Cluster (Cluster 2007), Austin, Texas, September 2007

Proença, A.J., Barbosa, J., Sobral, J.L.; Imaging Techniques to Simplify the PTM Generation of a Bas-Relief, Proc. 8th Int. Symposium on Virtual Reality, Archaeology and Cultural Heritage, VAST´2007, Brighton, UK, November 2007

Mudge, M., Malzbender, T., Chalmers, A., Scopigno, R., Davis, J., Wang, O., Gunawardane, P., Ashley, M., Doerr, M., Proenca, A. Barbosa, J., Image-Based Empirical Information Acquisition, Scientific Reliability, and Long-Term Digital Preservation for the Natural Sciences and Cultural Heritage. Full Day Tutorial, Eurographics 2008, April 14-18, 2008, Crete, Greece

A. Pina, B. Oliveira, J. Puga, R. Marques, A. Proença, An OGC-WS Framework to Run FireStation on the Grid, Proc. 3rd Iberian Grid Infrastructure, Conf (IBERGRID´09), Valência, Spain, May 2009

A. Pina, B. Oliveira, J. Puga, A. Esteves, A. Proença.; FireStation on the grid - a step further on the adoption of OGC/SDI standards. Proc. Enabling Grids for E-sciencE Conf. (EGEE'09), Barcelona, Spain, September 2009

A. Esteves, M. Caldas, A. Pina, A. Proença.; An OGC/SOS Conformant Client to Manage Geospatial Data on the GRID, Proc. 4th Iberian Grid Infrastructure Conf (IBERGRID´10), Portugal, May 2010

A. Mariano, R. Alves, J. Barbosa, L. P. Santos and Alberto Proenca; A (ir)regularity-aware task scheduler for heterogeneous platforms. Proc. 2nd Int Conf High Perf Computing. Kiev, October 2012

Jorge Manuel Gomes Barbosa: is a lecturer at the Departmento de Engenharia Informática of 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, algorithms for biomedical applications, 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 Computer Science from FEUP, in 2001. From 2003 to present, he is responsible of the course "Parallel Computing" of the MSc course in Informatics at FEUP. He participates in the Cost Action IC0805 "Open Network for High-Performance Computing on Complex Environments".

Supervision of Ph.D. students:

- Hamid Arabnejad, "Optimization of multi-user job scheduling on heterogeneous platforms", MAP-I, 2011.
- Tito Vieira, "Utility Computing on Scientific Cloud Infrastructures", ProDEI, FEUP, 2009.
- Altino Sampaio, "Virtualization Management on a Scientific Cloud Computing Infrastructure", ProDEI, FEUP, 2010.
- Tiago Santos, "Uncoordinated checkpointing with distributed message logging and recovery in modern clusters", ProDEI, FEUP, 2010.

Financed Research Projects:

- Rede Grid de Imagiologia Cerebral, GRID/GRI/81833/2006, Team budget 31560 Euros.
- GERES-med: repositórios em GRID para aplicações médicas, GRID/GRI/81819/2006, Team budget 61560 Euros.
- Re-equipment project: in 2002 collaborated in the proposal presented by Biomedical Engineering Institute to acquire a cluster infrastructure with 72 processors, which was installed in December 2006.

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, in press.

[2] J. Barbosa, Belmiro Moreira, "Dynamic scheduling of a batch of jobs on heterogeneous clusters", Parallel Computing, Elsevier, Vol.37 nº 8, pp.428-438, 2011.

[3] J. Barbosa, J. Tavares, A.J. Padilha, "Optimizing Dense Linear Algebra Algorithms on Heterogeneous Machines", Book Chapter in Algorithms and Tools for Parallel Computing On Heterogeneous Clusters, Nova Science Publisher, N.Y., pp 17-31, ISBN: 1-60021-049-X, 2007

[4] Altino Sampaio, Jorge G. Barbosa, "Dynamic Power- and Failure-Aware Cloud Resources Allocation for Sets of Independent Tasks". In IEEE International Conference on Cloud Engineering (IC2E), San Francisco, 2013

[5] Hamid Arabnejad, Jorge G. Barbosa, "Fairness resource sharing for dynamic workflow scheduling on heterogeneous systems". In IEEE Parallel and Distributed Processing with Applications (ISPA), pp.633-639, 2012

[6] Hamid Arabnejad, Jorge G. Barbosa, "Performance Evaluation of List Based Scheduling on Heterogeneous Systems". In Euro-Par 2011: Parallel Processing Workshops, volume 7155 of Lecture Notes in Computer Science, pages 440–449, 2012

[7] J. Barbosa, Belmiro Moreira, "Dynamic job scheduling on heterogeneous clusters", in International Symposium on Parallel and Distributed Computing, IEEE CS, pp.3-10, 2009

[8] J. Barbosa, M. Monteiro, "A List Scheduling Algorithm for Scheduling Multi-user Jobs on Clusters", in High Performance Computing for Computational Science - VECPAR 2008, 8th International Conference, LNCS 5336, pp.123-136, 2008

Scientific Activities:

- Since 2003, member of the Scientific Committee of the "Workshop on Algorithms, Models and Tools for Parallel Computing on Heterogeneous Networks – Heteropar", integrated on IEEE Cluster conference.
- Member of the Scientific Committee of the IASTED International Conference on Parallel and Distributed Computing and Networks, PDCN.
- Reviewer of scientific papers submitted to the "Vecpar'08 - International Meeting on High Performance Computing for Computational Science".
- Reviewer of IEEE International Symposium on Parallel and Distributed Computing.
- Reviewer of the international journal Parallel Computing - Systems & Applications, Elsevier.
- Reviewer of papers submitted to IEEE Transactions on Parallel and Distributed Systems.
- Reviewer of book chapters submitted to "HPC: Transition towards Exascale Computing", to be published in the series of "Advances in Parallel Computing" by IOS Press.