TCPP Curriculum in Parallel Programming courses of the Degree in Computer Science at the University of Murcia, Spain

Javier Cuenca, Domingo Giménez

Courses involved, courses of the 4th year of the Degree in Computer Science: Methodology of Parallel Programming, Manycore Architecture Programming.

Abstract

The proposal analyses the inclusion of parallel programming topics from the TCPP Curriculum in two parallel programming courses. The courses are in the 4th year of the Degree in Computer Science at the University of Murcia, Spain. In a previous Early Adopters project, we analyzed the inclusion of parallelism topics in courses on Parallel Programming, Computer Architecture and Operating Systems in the second year of the Degree, with which the students were introduced to different aspects of parallelism at an early stage, so they have some basic knowledge of parallelism, and in this proposal we center on upper-level parallel programming topics.

I. General lines of the proposal

Computer Science students at the University of Murcia follow a course on Concurrent and Distributed Programming in the second year of the Degree, and courses on Operating Systems and Computer Architecture (including some concepts of parallel architecture) in the second and third years. An experience about how to include parallel programming topics of the TCPP Curriculum in these courses and in Algorithms and Data Structure courses was analyzed in an Early Adopters project four years ago [1]. This experience served to indicate which topics are appropriate for inclusion in these courses, given the knowledge of our students, the study plan and other courses where more advanced parallelism topics can be included. The parallelism topics studied are most of them related to systems and architecture, some of them deal with concurrent programming, and only a few are about parallel algorithms.

Our proposal is centered in two courses in the fourth year of the Degree which the students interested in programming technology and in parallelism can follow. The course on Methodology of Parallel Programming (MPP) is in the first semester, and is followed by the students in the Software Engineering specialization and, optionally, by the students in Computer Engineering. In the second semester there is an optional course on Manycore Architecture Programming (MAP), which can be only followed by students in the Computer Engineering specialization.

The courses are given by professors from two departments, the MPP course by a professor of the Department of Languages and Systems ("Departamento de Lenguajes y Sistemas"), and the MAP course by a professor of the Department of Computer Engineering ("Departamento de Ingeniería y Tecnología de Computadores"). The two professors are members of the same research group, and they collaborate in researching and teaching activities [2], [3], and with this project they will increase teaching coordination between the courses.

The following sections treat the situation of parallelism in the Computer Science studies at the University of Murcia and, after that, the general structure, proposed actions and expected outcomes of the project.

II. Parallelism in the Computer Science Degree at the University of Murcia

Computer Science studies in Spain have been recently reorganized. They have changed from a structure with three different degrees organized in three or five years to a single degree (Degree on Computer Science, "Grado en Ingeniería Informática") of four years. In the previous organization there were two lower degrees (three years) on Management ("Ingeniería Técnica en Informática de Gestión") and Systems ("Ingeniería Técnica en Informática de Sistemas") and one upper degree of five years ("Ingeniería Informática"). In this new degree, specialization is achieved in the fourth year with five different profiles (specializations or intensifications) which comprise clusters of elective courses. Currently, the "Facultad de Informática" (Computer Science School) of the University of Murcia offers five intensifications: Computer Engineering, Software Engineering, Computing, Information Technology and Information Systems. The academic year 2013-2014 has seem for the first time students in the fourth year of the new Degree in Compute Science at the University of Murcia, so now is a good moment to work on how parallel programming should be included in this Degree.

On the other hand, due to the advances in technology, parallel computing is becoming very popular. At present, not only clusters or supercomputers are parallel systems, but also the standard computational systems. Laptops and desktops are multicore and include GPUs that can be programmed to take advantage of parallelism. This caused the initiative of the IEEE Technical Committee on Parallel Processing to define the core topics of parallelism which should be included in undergraduate curricula [4], and there are some proposals and experiences on the implementation of parallelism courses [5], [6].

A poster analyzing the situation of the teaching of parallel computing in the Spanish university system and proposing some actions to include parallelism in core and specialized courses was presented in the EduPar11 workshop [7]. The situation of parallelism teaching on some Spanish universities was analyzed, and this guided the introduction of topics of the TCPP Curriculum into initial courses of the Computer Degree at the University of Murcia [8]. After that first experience with general topics at low level courses, in this project we propose some actions for the inclusion of parallel programming concepts at a higher level.

The University of Murcia is a medium-size university in Spanish terms, with approximately 30000 students. It is a generalist university, with scientific, health, humanist and social studies, but with only two engineering degrees (in Chemistry and Computer Science). Most of the engineering studies in the Region of Murcia are at the Polytechnic University of Cartagena. The proposed project is set for the Computer Science studies at the University of Murcia, and it is specific for the Degree in Computer Science for the year 2014-2015, but the experience will guide the content of the courses in the project for following years, and may be valid for other parallel programming courses.

There are approximately 1200 students at the Computer Science School at the University of Murcia. Each year there are between 25 and 30 students in the Software Engineering specialization and between 5 and 10 in Computer Engineering. So, our proposal targets about 30 or 40 students, which represents approximately half of the students in the last year of the Degree. We think this situation is far from desirable where all the students in the Degree were introduced to parallelism concepts of architecture and systems and only in passing to parallel programming or algorithms. Our proposal can not change this situation, it only deepen the knowledge of students interested in parallel programming.

III. Structure of the project

The two courses involved in the project are:

- **Methodology of Parallel Programming (MPP)**. The course studies parallel programming paradigms, shared-memory, message-passing and hybrid programming, methodology of parallel programming, analysis and design of parallel algorithms, with examples of parallel schemes of divide-and-conquer, dynamic programming, backtracking and branch-and-bound, and of parallel algorithms schemes as, for example, pipeline, master-slave and tasks' pool.
- **Manycore Architecture Programming (MAP)**. The architecture of manycore systems is analyzed, including GPU cards and Intel MIC. The basics of programming for these systems are studied, together with programming environments: TBB, OpenMP, CUDA and OpenCL.

The topics of the TCPP curriculum treated in each course are summarized in Table I, where `K` represents knowledge on the topic and `C` indicates the topic is core in the course, and the students are capable of working with problems related with this topic at the end of the course.

As mentioned, a small number of students take the courses, and the teaching is personalized and focuses on the work of the students. They do various studies and practicals:

- In MPP:
  – Presentation about some alternative parallelism techniques or tools not included in the syllabus of the course.
  – OpenMP and MPI practicals with some basic problems. Tools from the Spanish Parallel Programming Contest are used [9]. The students work with some problems (year 2013-2014 with problems A, B and D from the 2013 contest), and send their solutions to the computational system, which evaluates them automatically and in real time. So the lecturer has access to the solutions (valid and non valid) provided by the students and detects each student's errors and he can correct them in individual tutorials.
  – Presentations on sequential and parallel algorithmic techniques to solve some challenging problem. In the year 2013-2014 it was an execution time-energy consumption bi-objective problem. Each student selects some method (backtracking, branch-and-bound, greedy, genetic, particle swarm, tabu...) for application to the problem, and analyzes the possible application to their selected method of the parallel programming methodologies and algorithmic schemes in the syllabus of the course.
  – Finally, the students work on the development of parallel versions of their selected method for this problem (for shared memory with OpenMP, for message-passing with MPI, and with MPI+OpenMP hybrid parallelism), analyze them theoretically and experimentally in a heterogeneous system with three nodes of 24+6+6 cores.
- In MAP:
  – Presentation about some alternative manycore architectures not included in the syllabus of the course.
  – CUDA practicals with some basic problems. The students have to complete and modify some basic CUDA programs, in order to get along with improving the data transfers between the host and the

TCPP	MPP	MAP
ARCHITECTURE		
Architecture classes	K	K
Streams (e.g. GPU)		C
MIMD	K	K
Multithreading	K	
Multicore	K	C
Heterogeneous (e.g. cell)	K	C
Shared vs. distributed memory	C	C
SMP	K	C
NUMA	K	K
Message Passing	C	K
Cache organization	K	K
Impact memory hier. on soft.	C	C
Performance Metrics	K	C
PROGRAMMING		
Parallel Programming Paradigms	C	C
Shared memory	C	C
Distributed memory	C	C
Client server	K	
Hybrid	K	C
Task/thread spawning	C	
SPMD	C	C
Data Parallel	C	C
Parallel loop	C	C
Language extensions	C	K
Compiler Directives pragmas	C	K
Libraries	C	K
SPMD Notations	C	K
MPI	C	
CUDA/OpenCL		C
Task and Threads	C	K
Synchronization	C	C
Critical regions	C	C
Producer-consumer	C	K
Deadlocks	C	K
Memory models	K	K
Scheduling and computation	C	C
Decomposition strategies	C	C
Scheduling and mapping	K	C
Data Distribution	K	C
Performance monitoring	K	C
Performance metrics	C	K
Speed-up	C	K
Efficiency	C	K
Amdahl's law	K	
Isoefficiency	C	
ALGORITHMS		
Asymptotics cost	C	
Time	C	K
Space	C	
Speedup	C	K
Cost	C	K
Divide and Conquer	C	
Algorithmic Problems Broadcast, Synchronization, Sorting	C	
Matrix computations	K	
Termination detection	C	
CROSS CUTTING		
Why and what is PDC	K	K
Power	K	
Cluster, Cloud, Grid	K	

TABLE I

Topics of the TCPP curriculum treated in the two courses in the project.

device, the data locality in the different GPU memory levels and the GPU card occupancy.

– Advanced hybrid practical: Tools from the Spanish Parallel Programming Contest are used [9]. The students work with some problems of the previous year Contest that must be solved with an hybrid parallelism approach using OpenMP plus CUDA in order to take the most of a heterogeneous system constituted of a multicore CPU and a manycore GPU.

The coordination between MPP and MAP courses consists on:

- The global plan and the educational skills of the two courses are explained and discussed with the students in each of them.
- The theoretical and the practical materials of the two courses is prepared coordinately.
- In several phases of each course, future possible works in conjunction with the other course are discussed.
- In MPP a general vision of the different parallel architecture is shown. From this point, it is possible to expose a deeper description of manycore architectures in MAP.
- In MPP different parallel paradigms are exposed. In the same way, a discussion about how to combine them is carried out. This will be the basis to face the designing and programming of a hybrid application in MAP.
- The two courses use the Spanish Parallel Programming Contest environment for developing, debugging and executing in some basic problems, which are in same cases common to both courses. In this way, both lecturers can access to the student solutions, detects possible mistakes and if necessary to set an appointment with the student to orientate his/her work.

The principal outcomes of the project will be:

- The reinforcement of the parallel computing concepts to Computer Science students, from two different but coordinated courses, by presenting students to practical aspects of parallel applications and systems.
- The combined vision of the two course can propitiate the desirable link between essential features for a computer scientist: algorithms, computer architecture and high performance programming.
- At the same time, the courses are a solid start point for students who decide to initialize a postgraduate researching career in the supercomputing field.
- Some material (documents, examples, practicals and tools), which could be used in future experiences at the University of Murcia or in other centers. As an example, the record table if the Spanish Parallel Programming Contest.

IV. Evaluation

Typically the teaching material is provided via web, and we follow active methodologies where the student is the main actor and responsible for her own learning. The instructors provide problems and projects, and the students work on solutions and prove their skills through them. The oral defenses and presentations play an important role. As told before, there is a motivational feedback among the problems derived from the courses and the Parallel Programming Contest. This active and individual methodology propitiates a close knowledge of the lecturers of the perception the students have of the courses and the different topics in them treated. Even so, at the end of course 2014-2015, the action will be evaluated with a test for the students about their vision about parallel programming. For each course, students will be evaluated as how much they have learned about each of the topics treated, and students following the two courses will be questioned about the advantages (or disadvantages) of a coordinated organization.

References

[1] Web page of the Early Adopter project at the University of Murcia, `http://www.um.es/earlyadopters/`, 2011.

[2] Web page of the Scientific Computing and Parallel Programming Group at the University of Murcia, `http://luna.inf.um.es/grupo_investigacion/`.

[3] Gregorio Bernabé, Javier Cuenca, Luis-Pedro García, Domingo Giménez, and Sergio Rivas-Gomez, "A high performance computing course guided by the lu factorization," in *ICCS*, 2014.

[4] IEEE Technical Committee on Parallel Processing, Curriculum on Parallel and Distributed Computing, `http://www.cs.gsu.edu/~tcpp/curriculum/index.php`.

[5] David J. Meder, Victor Pankratius, and Walter F. Tichy, `http://www.multicore-systems.org/separs/downloads/GI-WG-SurveyParallelismCurricula.pdf`, 2008.

[6] Ronal Muresano, Dolores Rexachs, and Emilio Luque, "Learning parallel programming: a challenge for university students," in *ICCS*, 2010, number 1, pp. 875–883.

[7] Francisco Almeida, Domingo Giménez, José-Miguel Mantas, and Antonio M. Vidal, "On the integration of the Curriculum Initiative on Parallel and Distributed Computing in the Spanish university system," in *First NSF/TCPP Workshop on Parallel and Distributed Computing Education (EduPar-11)*, 2011.

[8] Manuel E. Acacio, Javier Cuenca, Lorenzo Fernández, Ricardo Fernández Pascual, Joaquín Cervera, Domingo Giménez, M. Carmen Garrido, Juan A. Sánchez-Laguna, José Guillén, Juan Alejandro Palomino Benito, and María-Eugenia Requena, "An experience of early initiation to parallelism in the Computing Engineering Degree at the University of Murcia, Spain," in *IPDPS Workshops*, 2012, pp. 1289–1294.

[9] Spanish Parallel Programming Contests, `http://cpp.fpcmur.es`.