An experience of early initiation to parallelism in the Computing Engineering Degree at the University of Murcia, Spain

Manuel E. Acacio[1], Javier Cuenca[1], Lorenzo Fernández[1], Ricardo Fernández-Pascual[1], Joaquín Cervera[2], Domingo Giménez[2], M. Carmen Garrido[3], Juan A. Sánchez Laguna[3], José Guillén[4], Juan Alejandro Palomino Benito[4], María-Eugenia Requena[4]

[1] Department of Engineering and Technology of Computers

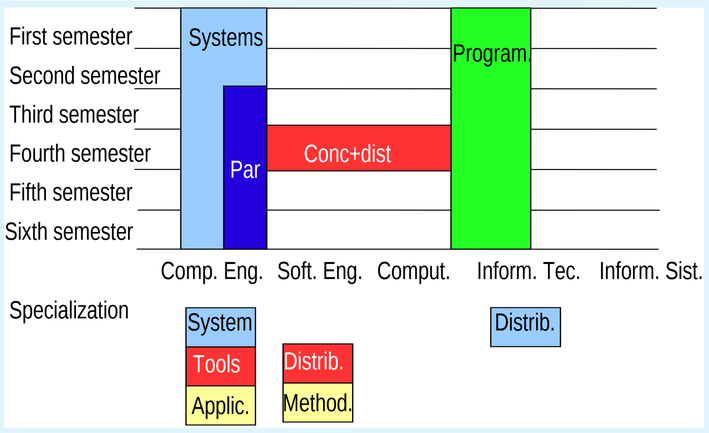
[2] Department of Informatic and Systems

[3] Department of Engineering of Information and Communication

[4] Supercomputing Centre, Scientific Park Foundation of Murcia

Parallelism at the Univ. of Murcia

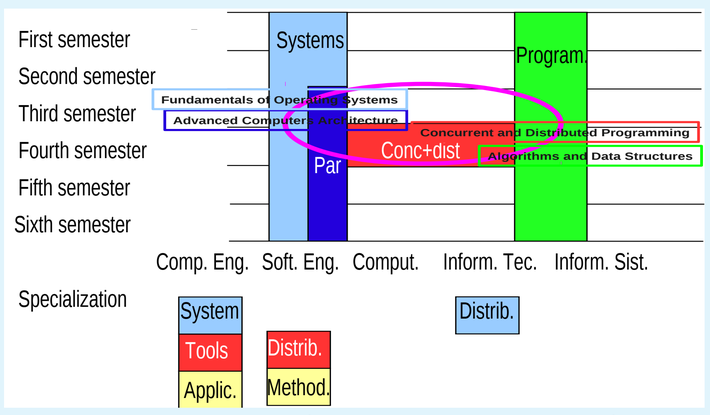
Present situation



- Parallelism in System courses from the third semester.
- Basic concepts of concurrency and distributed computing in a programming course in the fourth semester.
- Algorithmic aspects are not studied in any compulsory course.
- Intensification in parallelism in some specializations, but parallelism is not included in all the specializations.

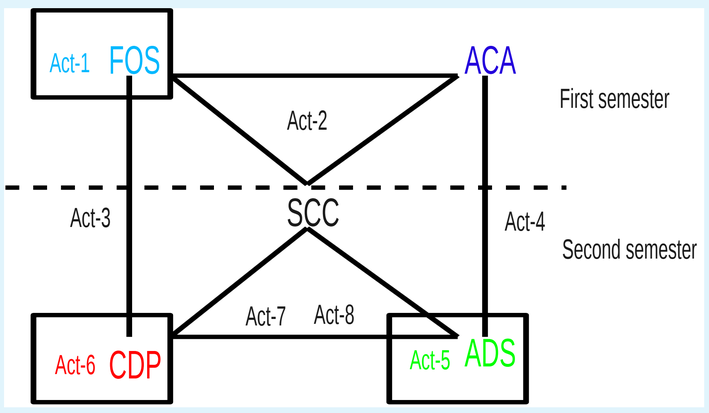
⇒ Computer Science students at the University of Murcia can obtain their degree without having developed and optimized any parallel code.

Early Adopters project



- Four compulsory courses in the second year of the degree. Two courses already included parallelism; two courses include topics of parallelism for the first time.
- Systems and Programming courses, with topics in the four parallelism aspects of the IEEE TCPP curriculum.
- Three departments and a Computing Centre: coordinated treatment of the topics.

Activities:



More information:
`http://www.um.es/earlyadopters`

Topics of the IEEE TCPP curriculum dealt with in the project

			Previous		First semester		Second semester					
Topic or course			ACA	CDP	Act1	Act2	Act3	Act4	Act5	Act6	Act7	Act8
FOS					X	X	X					
ACA			X			X		X				
CDP				X						X	X	X
ADS								X	X		X	X
SCC						X					X	X
ARCHITECTURE												
Architecture classes	0.5	C		X			X			X	X	X
Superscalar	0.5	C	X									
SIMD/Vector	0.5	K	X									
Pipelines	6	C	X									
OoO execution	4	C	X									
Multicore	1	K			X		X					
NUMA (shared memory)	0.5	K			X	X	X					
Cache organization	6	K	X									
Atomicity	2	C		X			X			X	X	X
Impact memory hier. on soft.	3.5	A	X		X			X				
Cycles per instr. Benchmarks	0.5	C	X									
Benchmarks	0.5	C	X									
Spec marks	0.5	C	X									
Peak performance	0.5	K	X									
MIPS/FLOPS	1	C	X					X	X		X	X
Sustained performance	0.5	K	X									
PROGRAMMING												
Shared memory	12	A								X	X	
Distributed memory	12	C								X		X
Client server	0.5	C		X							X	X
Task/thread spawning	2.5	A		X							X	X
SPMD	2	C		X							X	X
Shared memory notations	10	A		X						X	X	
Language extensions	1	K		X								
Libraries	10	A		X			X					
SPMD notations	3	A		X						X	X	
MPI	3	C								X		X
Semantic tasks and threads	5.5	C		X	X						X	X
Synchronization	2	A		X			X			X	X	X
Critical regions	2.5	A		X					X	X	X	
Producer-consumer	1.5	A		X						X	X	X
Monitors	4	A		X								
Deadlocks	0.5	K		X								
Memory models	0.5	K		X								
Scheduling and comp.	2	C	X									
Decomposition strategies	1	K	X									
Loop fusion	0.5	A	X									
Scheduling and mapping	3	C	X		X							
Performance monitoring	2	A			X							
Performance metrics	1.5	C	X						X		X	X
Speed-up	2	C	X						X		X	X
Efficiency	1	C	X						X		X	X
Amdahl's law	1	C	X									
ALGORITHMS												
Asymptotics cost.	0.5	C							X			
Time	0.5	C							X			
Space	0.5	C							X			
Speed-up	0.5	C							X		X	X
Notions from scheduling	1	K		X					X		X	X
Divide and Conquer	0.5	A								X	X	X
Broadcast	0.5	K		X								
Asynchrony	1	K		X								X
Synchronization	1	A		X								
Sorting	0.5	A							X		X	X
Graph search	0.5	K							X		X	X
Specialized computations	1	K						X	X		X	X
CROSS CUTTING												
Why and what is PDC.	1	C		X								
Concurrency	1	C		X								
Non-determinism	1	A		X								
Power	0.5	K	X									
Locality	1.5	C	X		X							
Security in Dist. systems	1	K			X	X						

In red are the new topics and those treated in more depth.

Approximate number of hours devoted in each course to each part of the curriculum (previously treated/included in the project):

	Architecture	Programming	Algorithms	Cross Cutting	TOTAL
FOS	0 / 1	0 / 5	-	0 / 0.5	0 / 6.5
ACA	21 / 1	13 / 0	-	2 / 0.5	36 / 1.5
CDP	1 / 0.5	42 / 13	2.5 / 0	3 / 0	48.5 / 13.5
ADS	0 / 1.5	0 / 13.5	0 / 5.5	-	0 / 20.5
TOTAL	22 / 4	55 / 31.5	2.5 / 5.5	5 / 1	84.5 / 42

Funding

This work is supported by Intel through the IEEE TCPP Early Adopters program, and partially supported by the Conserjería de Educación de la Región de Murcia (Fundación Séneca, 08763/PI/08).