This 4th edition of the BSC Severo Ochoa Doctoral Symposium was held from 2 to 4 of May, 2017 at UPC. Around 90 attendees took part in the event with 45 talks and poster presentations given by PhD students representing all the research departments at BSC. We had five different sessions of talks tackling the topics of: HPC & Novel Computer Architectures, Mathematics, Algorithms & Computational, Programming Models, Performance analysis & Software Tools, Simulations & Modeling and Mathematics, Algorithms & Computational.

The posters were exhibited and presented during four poster sessions that created lively discussion and gave the authors the opportunity to explain their research and results. The directors of BSC strongly supported the creation of the symposium and its organisation and Mateo Valero, Director of BSC, gave the opening speech.

The keynote speaker was Alfonso Valencia, who gave the lecture Personalised Medicine as a Computational Challenge. He is the Director of the Life Sciences’ Department at BSC. In his opinion “facing the enormous biomedical challenges of the future will only be possible with the coordination of the incredible scientific and technical resources of the BSC, in the rich scientific environment of Barcelona, and in combination with both National (INB-ISCIII) and European scientific infrastructures (ELIXIR)”

The BSC researcher Dario Garcia was in charge of the tutorial sessions on “Deep learning”.

The goal of the symposium is twofold: first, to provide a framework to share results of research undertaken by PhD students at BSC; second, to offer training sessions on topics and skills that will be useful to them as
future researchers and professionals. Created within the framework of the Severo Ochoa Programme at BSC, in line with the project’s aims relating to talent development and knowledge sharing, the symposium offers an interactive forum for PhD students, both those who are just beginning their research and those who have progressed far enough to share some results.

Further information about the Doctoral Symposium

Event photos

See full album
Time Predictable Parallel Programming Models

Motivation
- Convergence of RPC and EC
- Embedded systems are increasingly concerned with
  providing higher performance in real-time, anticipating the performance capabilities of current architectures
- Next-generation many-core platforms require the converging need for predictable performance

Parallel Programming Models
- Provide the abstraction level required to exploit the parallel performance opportunities of many-core, while harnessing processor complexities
- OpenMP: widely used in HPC, offers a convenient tasking execution model with interesting similarities with real-time scheduling models

Towards a Predictable OpenMP

OpenMP Tasking Model
- Allows expressing fine-grained and irregular operations with mature support for data dependencies among tasks
- A task is an independent unit of work executed by an available thread/core in the processor. It is possible to monitor data dependencies among tasks.

Sporadic DAG Scheduling Model
- The system is expressed as a set of periodic tasks, each of which is a Directed Acyclic Graph (DAG) with a set of edges representing dependencies between tasks
- Tasks are represented with a Directed Acyclic Graph (DAG)
  - G = (V,E): DAG structure
  - (v1,...,vn): Set of nodes. Each node v_i is characterized by its worst-case execution time (WCET) τ_i
  - E: Set of edges (precedence constraints)
  - P: Period or maximum interarrival time
  - B: Deadline
- Some properties:
  - WCET = \sum_{v_i} τ_i Volume of the DAG
  - B = \max(\tau_i) + \sum_{v_i} τ_i
  - Depth of the longest path (critical path)
  - Number of tasks: \#(v_i)

Scheduling Analysis of OpenMP

Scheduling task: Define as upper bound the worst-case response-time of a (not-off-the-shelf task) and compare with its deadline. If the deadline and the WCET of the task are exceeded, the task is said to miss its deadline.

Scheduling Analysis of a set of OpenMP tasks

Optimization Task Set
- Task scheduling and tasking allocation
- Task dependencies
- Work-conserving scheduler
- Dynamic load balancing

Unified Model
- Tasks can be mapped to available threads during runtime, while their dependencies are fulfilled
- Allows the implementation of work-conserving schedulers
- A task is started if all of its predecessors have been completed

Response Time Analysis of a set of OpenMP tasks

Limited Preemption
- A task possesses a preemption time that is determined by the job's boundaries
- Preemption is only performed at task boundaries