

[BSC hosts its 4th Doctoral Symposium](#)



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.

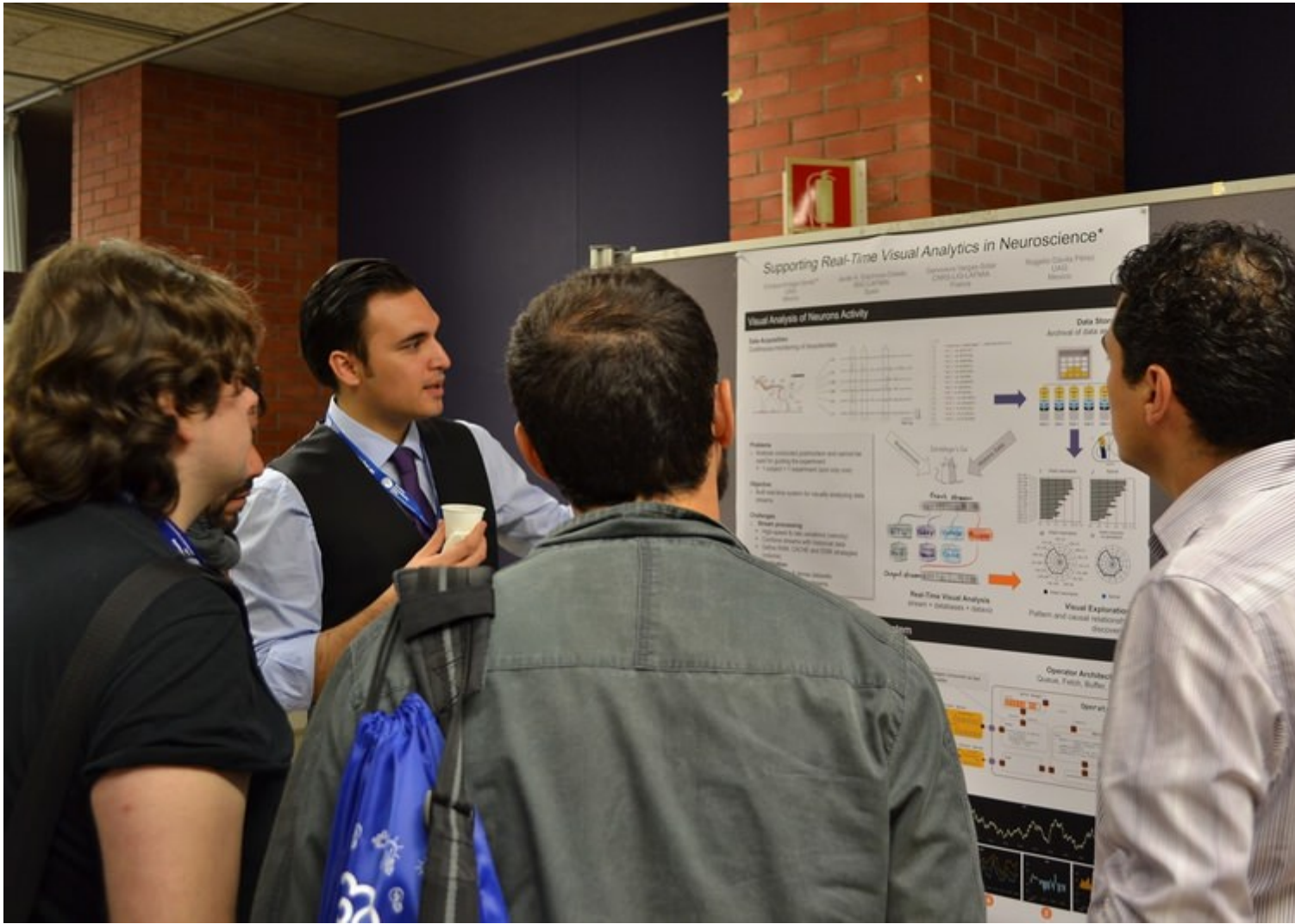
[Further information about the Doctoral Symposium](#)

Event photos

[See full album](#)









Time Predictable Parallel Programming Models

María A. Serrano
maria.serrano@bsc.es
Barcelona Supercomputing Center
and Technical University of Catalonia, Spain

Eduardo Quilones
eduardo.quilones@bsc.es
Barcelona Supercomputing Center, Spain

MOTIVATION

Convergence of HPC and EC

- Embedded systems are increasingly concerned with providing **higher performance in real-time**, challenging the performance capabilities of current architectures
- Next-generation many-cores can intercept this converging need for **predictable performance**

TOWARDS A PREDICTABLE OPENMP

OpenMP Tasking Model

- Allows expressing fine-grained and irregular parallelism with mature support to 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 express data **dependencies** among tasks.

```

#pragma omp parallel
#pragma omp single // T1
#pragma omp task // T2
#pragma omp taskwait // T3
#pragma omp taskwait // T4
#pragma omp taskwait // T5
#pragma omp taskwait // T6
#pragma omp taskwait // T7
#pragma omp taskwait // T8
                
```

Parallel Programming Models

- Provide the abstraction level required to exploit the parallel performance opportunities of many-core, while hiding processor complexities
- **OpenMP**, widely used in HPC, offers a sophisticated tasking execution model with interesting similarities with real-time scheduling models

Sporadic DAG Scheduling Model

- The system is expressed as a set of periodic tasks, each of them composed of connected sub-tasks (task parts in OpenMP), upon which **timing guarantees are derived**
- Tasks are represented with a Directed Acyclic Graph (DAG)

$$\tau = \langle G, T, D \rangle$$
 - $G = (V, E)$: DAG structure
 - $V = \{v_1, \dots, v_n\}$: Set of nodes. Each node, $v_i \in V$ is characterized by its worst case execution time (WCET) C_i
 - $E = E \cup V$: Set of edges (precedence constraints)
 - T : Period or minimum inter-arrival time
 - D : Deadline
- Some properties:
 - $vol(G) = \sum_{v_i \in V} C_i$: Volume of the DAG
 - WCET of the task when executing on a dedicated single-core platform
 - $len(G)$: Length of the longest path (critical path)
 - Minimum amount of time needed to execute the DAG-task on a sufficiently large number of processors
 - m : Number of available threads/cores

SCHEDULABILITY ANALYSIS OF OPENMP

Schedulability Analysis of an OpenMP task

Tied tasks

- task cannot **resume** its execution in a thread different to the one it started
- **do not allow** the implementation of **work-conserving** schedulers

Untied tasks

- Tasks can **freely migrate** across available threads during execution once all dependencies are fulfilled
- **allow** the implementation of **work-conserving** schedulers

allow deriving a tight response time upper bound:

$$R^{th} = len(G) + \frac{vol(G) - len(G)}{m} \leq D$$

Response Time Analysis of a set of OpenMP untied tasks

- **Limited Preemption**: a task implicitly executes in non-preemptive mode and preemption is allowed only at predefined locations inside the code (node boundaries in OpenMP DAGs)

Response time upper bound:

$$R^{th} = len(G) + \frac{vol(G) - len(G)}{m} + \sum_{v_i \in V} \frac{C_i}{m} \leq D$$

Barcelona Supercomputing Center - Centro Nacional de Supercomputación

Source URL (retrieved on 18 Mar 2025 - 14:32): <https://www.bsc.es/es/news/bsc-news/bsc-hosts-its-4th-doctoral-symposium>