

## New OmpSs release 16.06

The main feature introduced in OmpSs 16.06 is the possibility of executing OmpSs applications on top of Distributed Memory Systems in a transparent way to the user.



The [Programming Models team](#) is glad to announce you the release of the new stable version of [OmpSs \(16.06\)](#), which is based on the latest Mercurium source-to-source compiler and Nanos++ 0.10 runtime library.

The main objective of this group is to investigate programming paradigms towards productive programming and their implementation through intelligent runtime systems that effectively exploit performance out of the target architecture (from multicore and SMT processors to shared- and distributed-memory systems, small and large-scale cluster systems, including both homogenous and heterogenous systems that use accelerators like GPUs).

The Programming Models team currently organizes its work around the design of the [OmpSs programming model](#)

. OmpSs is an effort to integrate features from the StarSs programming model developed by BSC into a single programming model. In particular, our objective is to extend OpenMP with new directives to support asynchronous parallelism and heterogeneity (devices like GPUs). However, it can also be understood as new directives extending other accelerator based APIs like CUDA or OpenCL. Our OmpSs environment is built on top of our Mercurium compiler and Nanos++ runtime system.

Summarizing the efforts of the last year, the group has released a new stable version of OmpSs. In this new version, apart from several bug-fixes in both tools, they have introduced the following features:

### **1. New cluster support**

- Execute OmpSs programs transparently on top of a distributed memory system (CUDA & OpenCL devices are also supported).
- Several optimization mechanisms allow to maximize the performance of applications running on a cluster.

### **2. Support for non-contiguous data**

- Tasks can reference non-contiguous, multi-dimensional data, which eases the implementation of some applications.

### **3. Thread manager**

- The Thread Manager module dynamically controls the amount of working threads needed for a specific amount of workload.

### **4. Task reductions**

- Extend the task construct adding support to the reduction clause.
- Enhance the support of user-defined reductions.

For more detailed information about OmpSs 16.06, visit <https://pm.bsc.es/ompss-release-1606>

Barcelona Supercomputing Center - Centro Nacional de Supercomputación

---

**Source URL (retrieved on 19 Mar 2025 - 00:32):** <https://www.bsc.es/es/news/bsc-news/new-ompss-release-1606>