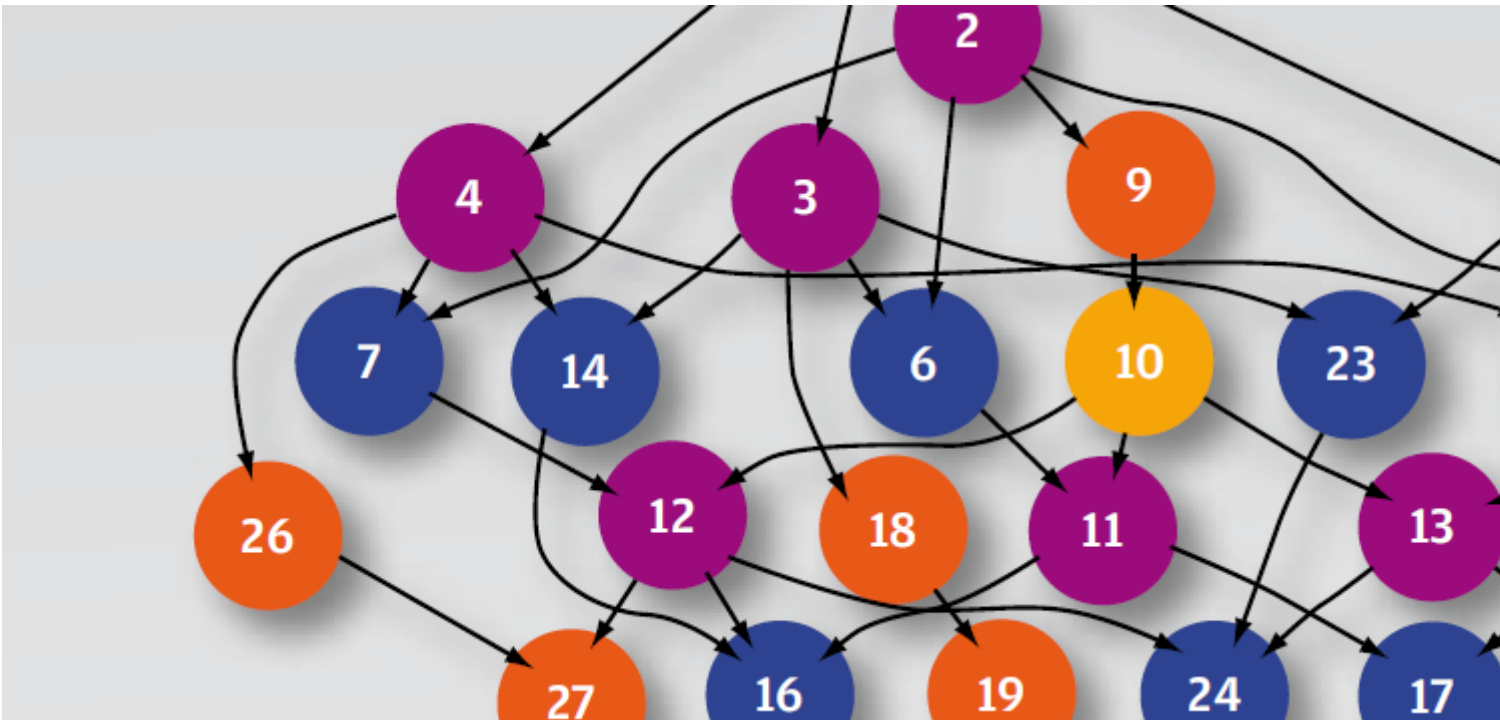


Release of OmpSs-2 programming model

Programming Models team at BSC has done the first public release (17.11) of the OmpSs-2 programming model.



OmpSs-2 extends the tasking model of OmpSs/OpenMP to support both task nesting and fine-grained dependencies across different nesting levels, which enables the effective parallelization of applications using a top-down methodology. OmpSs-2 is implemented on top of the Mercurium source-to-source compiler and the Nanos6 runtime system, which has been developed from scratch to efficiently support the enhanced task dependency system of OmpSs-2.

These are the most prominent features introduced in the first public version of OmpSs-2:

1. **Lifetime of task data environment**

A task is completed once the last statement of its body is executed. It becomes deeply completed when also all its children become deeply completed. The data environment of a task, which includes all the variables captured when the task is created, is preserved until the task is deeply completed. Notice that the stack of the thread that is executing the task is NOT part of the task data environment.

2. **Nested dependency domain connection**

Incoming dependences of a task propagate to its children as if the task did not exist.

When a task finishes, its outgoing dependencies are replaced by those generated by its children

3. **Early release of dependencies**

By default, once a task is completed it will release all the dependencies that are not included on any unfinished descendant task. If the wait clause is specified in the task construct, all its dependencies will be released at once when the task becomes deeply completed.

4. **Weak dependencies**

The weakin/weakout clauses specify potential dependencies only required by descendants tasks. These annotations do not delay the execution of the task.

5. **Task reductions**

Extend the task construct adding support for the reduction clause, which works like inout dependencies regarding points 2 & 3

6. **Native offload API**

A new asynchronous API to execute OmpSs-2 kernels on a specified set of CPUs from any kind of application, including Java, Python, R, etc.

7. **Task Pause/Resume API**

A new API that can be used to programmatically suspend and resume the execution of a task. It is currently used to improve the interoperability and performance of hybrid MPI and OmpSs-2 applications.

You can download Mercurium and Nanos6 from GitHub:

[Mercurium](#) [Nanos6](#)

Barcelona Supercomputing Center - Centro Nacional de Supercomputación

Source URL (retrieved on 20 Mar 2025 - 20:21): <https://www.bsc.es/ca/news/bsc-news/release-ompss-2-programming-model>