

## Nou llançament del model de programació OmpSs-2

L'equip de models de programació del BSC ha llançat la versió 18.06 d'OmpSs-2.



OmpSs-2 amplia el model de tasques de OmpSs / OpenMP per permetre al mateix temps tasques niuades i dependències de gra fi entre diferents nivells de nidificació. Amb això, s'aconsegueix una paral·lelització efectiva de les aplicacions gràcies a l'ús d'una metodologia vertical. En aquest llançament, s'ha millorat el model de tasques de OmpSs-2 per millorar la programació híbrida (MPI + OmpSs-2) i l'heterogènia (OmpSs-2 + CUDA C nuclis). La següent llista detalla les característiques més destacades que s'inclouen en el nou OmpSs-2:

### **1. Biblioteca Task-Aware MPI (TAMPI)**

S'ha desenvolupat aquesta biblioteca en el context del projecte [INTERTWinE](#) i està disponible en [github.com/bsc-pm/tampi](https://github.com/bsc-pm/tampi). Augmenta la interoperabilitat de les característiques de MPI per millorar la programació híbrida amb models de tasques com OmpSs-2. A més, s'ha afegit un nou "MPI threading level" (anomenat MPI\_TASK\_MULTIPLE), que permet l'ús segur d'operacions MPI síncrones i asíncrones en una tasca. Aquesta llibreria depèn de diverses les APIs proporcionades per Nanos6.

### **2. CUDA Unified Memory**

S'ha ampliat el model de tasques de OmpSs-2 per permetre tasques que estan escrites en CUDA. Els kernels de CUDA anotats amb el constructor *task* d'OmpSs-2 planifiquen com a tasques normals, de manera que se simplifica el desenvolupament d'aplicacions heterogènies. La implementació actual es basa en la Unified Memory de CUDA, que està disponible a les targetes d'última generació de NVIDIA i que permet el moviment automàtic de dades entre l'ordinador i el dispositiu gràfic i viceversa.

### 3. Prioritats en les tasques

La nova implementació del planificador permet especificar la clàusula *priority* d'OpenMP. Per defecte, les tasques tenen prioritats 0, però es pot canviar a través del valor sencer especificat com a paràmetre de la clàusula *priority*. Els valors elevats indiquen prioritats altes mentre que els inferiors suposen una prioritats baixa. Les prioritats també poden ser negatives per a indicar menys prioritats que la que apareix per defecte.

### 4. Reduccions sobre matrius (C & C++)

Les reduccions es beneficien de treballar amb múltiples còpies de les dades d'una tasca per facilitar l'execució en paral·lel. Mentre les reduccions escalars ja eren part del model de programació, aquest llançament inclou la possibilitat d'especificar el tipus matriu en la clàusula *reduction*. A més, la nova clàusula *weakreduction* es pot utilitzar per especificar la regió de la memòria on s'especificaran reduccions sobre matrius, possibilitant reduccions niuades i una reserva de memòria optimitzada.

### 5. Nanos6 API genèrica d'execució periòdica de serveis

Aquesta nova API coordina l'execució de tasques amb la de funcions arbitràries. El seu objectiu és millorar la interoperabilitat amb el programari que requereixi executar periòdicament una funció de manera que minimitza les interferències en l'ús de recursos.

### 6. External events API

Aquesta nova API es pot utilitzar per retardar l'alliberament de dependències d'una tasca fins que aquesta s'hagi executat i fins que s'hagin completat una sèrie d'esdeveniments externs (per exemple, fins a la finalització d'una operació MPI asíncrona). També està destinada a implementar el suport necessari que permet operacions MPI asíncrones a la biblioteca TAMPI.

[Descarregar a Github](#)

Barcelona Supercomputing Center - Centro Nacional de Supercomputación

---

Source URL (retrieved on 20 Mar 2025 - 04:28): <https://www.bsc.es/ca/noticies/noticies-del-bsc/nou-llan%C3%A7ament-del-model-de-programaci%C3%B3-ompss-2>