

## [El BSC hace pública la versión 2.8 de COMPSs y la versión 0.6.0 de la dislib](#)

El Barcelona Supercomputing Center ofrece a la comunidad HPC un conjunto de herramientas que ayudan a los desarrolladores a programar y ejecutar sus aplicaciones de forma eficiente en infraestructuras computacionales distribuidas; y la dislib, una librería de machine learning paralelizada con PyCOMPSs.



Esta versión incluye una extensión para mejorar la programabilidad de las reducciones definidas por el usuario. Con esta adición, la reducción se puede describir fácilmente con una anotación a las tareas, el runtime es capaz de descomponer la tarea en diversas tareas que se pueden ejecutar en paralelo y que tienen en cuenta la localidad de los datos a nivel de nodo, reduciendo las comunicaciones inter-nodo.

Se ha añadido una nueva anotación de tareas @container que permite la orquestación de aplicaciones PyCOMPSs que incluyen tareas desplegadas en contenedores. COMPSs da soporte a los entornos de contenedores Docker y Singulairy.

La nueva versión de la dislib incluye Daura, el algoritmo de clusterización más utilizado por la comunidad de dinámica molecular; un PCA con SVD paralelo; y nuevos métodos básicos como la multiplicación de matrices o el producto de Kronecker.

El grupo [Workflows and Distributed Computing team](#) del Barcelona Supercomputing Center-Centro Nacional de Supercomputación (BSC) está orgulloso de anunciar el lanzamiento de la versión 2.8 (nombre en clave Iris), del entorno de programación [COMPSs](#).

Esta versión de COMPSs, disponible a partir de hoy, actualiza el resultado del trabajo del equipo en los últimos años en la provisión de un conjunto de herramientas que ayudan a los desarrolladores a programar y ejecutar sus aplicaciones de forma eficiente en infraestructuras computacionales distribuidas, como clústers, nubes y clústeres gestionados con contenedores. COMPSs es un modelo de programación basado en tareas capaz de mejorar notablemente el rendimiento de aplicaciones a gran escala mediante la paralelización de manera automática de su ejecución.

COMPSs ha estado disponible en los últimos años por los usuarios del supercomputador MareNostrum y en la Red Española de Supercomputación (RES) y ha sido usado en varios proyectos de investigación como EUBra-BIGSEA, MUG, EGI, ASCETIC, TANGO, NEXTGenIO, y mF2C. En estos proyectos se ha utilizado COMPSs para desarrollar casos de uso proporcionados por diferentes comunidades de diversas disciplinas como la biomedicina, la ingeniería, la biodiversidad, la química, la astrofísica y ciencias de la tierra. Actualmente también está en extensión y en uso en aplicaciones en los proyectos ExaQUte, LANDSUPPORT, el BioExcel CoE, CLASS, ELASTIC, i l'EXPERTISE ETN, así como en un contrato de investigación con FUJITSU i en el proyecto Edge Twins HPC del tipo Innovation Launchpad. También ha sido aplicado a un caso de uso del ChEESE CoE.

La nueva versión incluye una extensión para mejorar la programabilidad de las reducciones definidas por el usuario. Se ha definido una nueva directiva `@reduction` para anotar las reducciones definidas por el usuario. El runtime de COMPSs es capaz de descomponer la tarea de reducción en diversas que pueden ser ejecutadas en paralelo siguiendo una forma de árbol invertido. El runtime ejecuta un algoritmo interno que tiene en cuenta la localidad de los datos, realizando primero la reducción de los datos de un mismo nodo y reduciendo posteriormente los datos entre nodos. Esto reduce extremadamente el número de comunicaciones entre nodos en comparación con una solución que no tiene en cuenta la localidad.

El runtime de COMPSs permite el despliegue utilizando contenedores desde la versión 1.4. Sin embargo, el apoyo anterior se limitaba a empaquetar toda la aplicación en un solo contenedor. A partir de esta imagen, el runtime era capaz de desplegarla en varios contenedores e iniciar la ejecución de la aplicación sin la interacción del usuario mediante el entorno de contenedores. Entornos soportados son Docker, Singularity y Mesos. En la versión 2.8, el soporte para contenedores se ha ampliado con una nueva directiva `@container` que se utiliza para anotar tareas incluidas en una imagen de contenedor. En este sentido, el runtime de COMPSs se convierte en un orquestador de workflows compuestos por tareas nativas y tareas incluidas en imágenes de contenedores, pudiendo interactuar con el entorno de contenedores sin la intervención del usuario. Las tareas anotadas con `@container` pueden ser invocación a binarios externos o código de usuario proporcionado por el desarrollador de la aplicación. En este segundo caso, el código de usuario se ha desplegado previamente en la imagen del contenedor.

La versión 2.8 de COMPSs también incluye: un nuevo tipo de parámetro, `IN_DELETE`, que especifica parámetros de un solo uso que son borrados después de la ejecución de la tarea; un nuevo formato de datos para colecciones en tareas MPI; y la posibilidad de usar colecciones de diccionarios Python como parámetros.

Además, la versión 2.8 de COMPSs viene dotada con otras funcionalidades menores, extensiones y resolución de errores.

COMPSs ha tenido alrededor de **1000 descargas** durante el año pasado y es utilizado por unos **20 grupos** en aplicaciones reales. Recientemente, COMPSs ha atraído el interés de áreas como la ingeniería, el reconocimiento de imágenes, la genómica y sismología, áreas en las que se han realizado cursos específicos y acciones de divulgación.

Los paquetes y la lista completa de funciones están disponibles en la página de [descargas](#). En esta página también se puede encontrar una máquina virtual que permite probar las funcionalidades de COMPSs a través de un tutorial paso a paso que guía al usuario en el desarrollo y ejecución de un conjunto de aplicaciones de ejemplo.

Además del software, hay un conjunto de guías de usuario y administrador, y artículos publicados en conferencias y revistas relevantes.

Para más información, visite la página web: <http://www.bsc.es/compss>



### **Nueva versión de la dislib**

El grupo también está orgulloso de anunciar la nueva versión de la [dislib](#) 0.6.0. La Biblioteca de Computación Distribuida (dislib) proporciona algoritmos distribuidos preparados para utilizarse como biblioteca. Hasta ahora, dislib se centra en algoritmos de machine learning y ofrece una interfaz inspirada en scikit-learn. El objetivo principal de dislib es facilitar la ejecución de algoritmos de análisis de datos masivos en plataformas distribuidas, como clusters, clouds y superordenadores. Dislib ha implementado encima del modelo de programación PyCOMPSs, la interfaz Python de COMPSs.

Dislib se basa en una estructura de datos distribuida, ds-array, que permite la ejecución paralela y distribuida de los métodos de machine learning. El código de la biblioteca de dislib se implementa como una aplicación PyCOMPSs, donde los diferentes métodos se anotan como tareas de PyCOMPSs. En el momento de ejecución, PyCOMPSs se encarga de todos los aspectos de paralelización y distribución de datos. Sin embargo, el código de usuario de dislib no es consciente de los aspectos de paralelización y distribución, y se escribe como un script de Python simple, con una interfaz muy similar a la de scikit-learn. Dislib incluye métodos para hacer clustering, clasificar, regresión, descomposición, selección de modelos y gestión de datos. Un contrato de investigación con FUJITSU ha financiado parcialmente la biblioteca dislib y actualmente se utiliza para evaluar el procesador A64FX.

Desde su reciente creación, dislib ha aplicado en casos de uso de astrofísica (DBSCAN, GAIA) y en flujos de trabajo de dinámica molecular (Dora y PCA, BioExcel CoE).

La nueva versión 0.6.0 incluye una nueva regresión lineal multivariante, una implementación paralela de la descomposición del valor singular (SVD), una nueva versión del análisis de componentes principales (PCA) que utiliza este SVD paralelo, el algoritmo ADMM Lasso y el algoritmo de clusterización Daura.

Dos de estos códigos, PCA con SVD y Daura clustering, se han desarrollado teniendo en cuenta las necesidades de la comunidad BioExcel, que requiere analizar los resultados de grandes trayectorias resultantes de simulaciones de dinámicas moleculares. Daura es el algoritmo de clusterización utilizado principalmente por la comunidad de dinámicas biomoleculares, y se incluye por ejemplo en el conjunto de herramientas GROMACS. Sin embargo, las implementaciones anteriores de este algoritmo son secuenciales y se limitan al uso de una memoria de un solo nodo. Esta limitación se supera con la nueva implementación de dislib, que permite agrupar trayectorias de simulación muy grandes. Del mismo modo, una versión anterior del PCA de la dislib utilizaba un método secuencial para encontrar los valores propios y los vectores propios de una matriz de covarianza. Este método secuencial limitaba la aceleración del PCA y el tamaño de los problemas a resolver. El nuevo PCA viene con un SVD paralelo que supera la limitación anterior y puede utilizar la memoria de varios nodos.

Dislib 0.6.0 viene con otras extensiones y con una nueva guía del usuario. El código es de código abierto y está disponible para [descargarlo](#).

Barcelona Supercomputing Center - Centro Nacional de Supercomputación

---

**Source URL (retrieved on 14 Jul 2024 - 08:55):** <https://www.bsc.es/es/noticias/noticias-del-bsc/el-bsc-hace-p%C3%BAblica-la-versi%C3%B3n-28-de-compss-y-la-versi%C3%B3n-060-de-la-dislib>