Nuevo lanzamiento del modelo de programación OmpSs-2

El equipo de modelos de programación del BSC ha lanzado la versión 18.06 de OmpSs-2.

OmpSs-2 amplía el modelo de tareas de OmpSs/OpenMP para permitir al mismo tiempo tareas anidadas y dependencias de grano fino entre diferentes niveles de anidamiento. Con esto, se consigue una paralelización efectiva de las aplicaciones gracias al uso de una metodología vertical. En este lanzamiento, se ha mejorado el modelo de tareas de OmpSs-2 para mejorar la programación híbrida (MPI+OmpSs-2) y la heterogénea (OmpSs-2+CUDA C kernels). La siguiente lista detalla las características más destacadas que se incluyen en el nuevo OmpSs-2:

1. BibliotecaTask-Aware MPI (TAMPI)

Se ha desarrollado esta biblioteca en el contexto del proyecto INTERTWinE y está disponible en github.com/bsc-pm/tampi. Aumenta la interoperabilidad de las características de MPI para mejorar la programación híbrida con modelos de tareas como OmpSs-2. Además, se ha añadido un nuevo “MPI threading level” (llamado MPI_TASK_MULTIPLE), que permite el uso seguro de operaciones MPI síncronas y asíncronas en una tarea. Esta librería depende de varias las APIs proporcionadas por Nanos6..

2. CUDA Unified Memory

Se ha ampliado el modelo de tareas de OmpSs-2 para permitir tareas que están escritas en CUDA. Los kernels de CUDA anotados con el constructor *task* de OmpSs-2 planifican como tareas normales, con lo que se simplifica el desarrollo de aplicaciones heterogéneas. La implementación actual se basa en la *Unified Memory* de CUDA, que está disponible en las tarjetas de última generación de NVIDIA y que permite el movimiento automático de datos entre el ordenador y el dispositivo gráfico y viceversa.

3. Prioridades en las tareas

La nueva implementación del planificador permite especificar la cláusula *priority* de OpenMP. Por defecto, las tareas tienen prioridad 0, pero se puede cambiar a través del valor entero especificado como parámetro de la cláusula *priority*. Los valores elevados indican prioridad alta mientras que los inferiores suponen una prioridad baja. Las prioridades también pueden ser negativas para indicar menos prioridad que la que aparece por defecto.

4. Reducciones sobre matrices (C & C++)

Las reducciones se benefician de trabajar con múltiples copias de los datos de una tarea para facilitar la ejecución en paralelo. Mientras las reducciones escalares ya eran parte del modelo de programación, este lanzamiento incluye la posibilidad de especificar el tipo matriz en la cláusula *reduction*. Además, la nueva cláusula *weakreduction* se puede utilizar para especificar la región de la memoria donde se especificarán reducciones sobre matrices, posibilitando reducciones anidadas y una reserva de memoria optimizada.

5. Nanos6 API genérica de ejecución periódica de servicios

Este nueva API coordina la ejecución de tareas con la de funciones arbitrarias. Su objetivo es mejorar la interoperabilidad con el software que requiera ejecutar periódicamente una función de forma que minimiza las interferencias en el uso de recursos.

6. External events API

Esta nueva API se puede utilizar para retrasar la liberación de dependencias de una tarea hasta que ésta se haya ejecutado y hasta que se hayan completado una serie de eventos externos (por ejemplo, hasta la finalización de una operación MPI asíncrona). También está destinada a implementar el soporte necesario que permite operaciones MPI asíncronas en la biblioteca TAMPI.

Descargar en Github

Barcelona Supercomputing Center - Centro Nacional de Supercomputación

---

Source URL (retrieved on *26 Dic 2024 - 14:35*): https://www.bsc.es/es/noticias/noticias-del-bsc/nuevo-lanzamiento-del-modelo-de-programaci%C3%B3n-ompss-2