Paralelización Automática y Estrategias de Desarrollo de Código Eficiente para Aumentar el Rendimiento en Centros de Supercomputación

Expertos de múltiples ramas del conocimiento se enfrentan diariamente a multitud de desafíos en proyectos científicos, técnicos o industriales que requieren el uso de la computación de alto rendimiento (HPC, High-Performance Computing) para satisfacer adecuadamente sus necesidades. Sin embargo, el desarrollo de algoritmos y programas que empleen correctamente este tipo de infraestructuras implica un importante conocimiento previo que muchos de estos usuarios no poseen, lo cual dificulta notablemente su labor.

Este tipo de proyectos, ampliamente multidisciplinares y heterogéneos, demandan una obtención de resultados con tiempos de ejecución realmente críticos, necesitando para ello la imprescindible utilización de la computación paralela. De este modo, con objeto de alcanzar resultados confiables y emplear eficientemente los recursos de cómputo disponibles, muchos de estos profesionales necesitan una importante ayuda adicional que les permita llevar a cabo la paralelización y optimización de sus códigos. De hecho, es habitual que el primer obstáculo al que se enfrenten sea precisamente la adaptación de sus propios algoritmos secuenciales. En algunos casos, implementar estos códigos ya resulta una tarea complicada para expertos de diversas ramas de la ciencia muy alejadas de la programación informática, lo cual prolonga significativamente su curva de aprendizaje. La situación empeora especialmente cuando precisan además el uso de la computación paralela. De esta forma, para estos usuarios obtener una solución adecuada puede convertirse en un objetivo de notable complejidad.

Este problema está principalmente motivado por el hecho de que los algoritmos paralelos no son diseñados únicamente para realizar el trabajo computacional, sino que también indican cómo será distribuido o dividido éste entre múltiples unidades de procesamiento. Así, es común que científicos e investigadores, que carecen de la formación específica requerida, empleen demasiado tiempo en desarrollar y ejecutar correctamente este tipo de algoritmos. Esto propicia además que sus códigos no sean tan eficientes como deberían, para lo cual se requiere mayor preparación y experiencia.

La completa paralelización automática de códigos de programación, considerando además su eficiencia y el correcto uso de los recursos de cómputo disponibles, sería de utilidad, no solo para estos investigadores, sino también para los propios administradores de los centros de computación de alto rendimiento. De esta forma, los usuarios podrían hacer un uso más eficiente tanto de su propio tiempo, como de las infraestructuras ofrecidas por estos centros, donde dicha eficiencia es clave en diversos aspectos, entre los que destacan: los tiempos de cómputo y de espera para acceder a las distintas infraestructuras, el número de núcleos o procesadores empleados y la disponibilidad de memoria y espacio en disco, sin olvidar el consumo energético, que actualmente supone un desafío clave para los centros HPC.

En consecuencia, uno de los objetivos principales de la presente tesis doctoral consiste en dar solución a los problemas identificados, proponiendo un transcompilador (compilador source-to-source) para la paralelización automática de códigos secuenciales, con el cual será posible obtener mejores rendimientos y eficiencias en las ejecuciones y escoger una correcta estrategia de planificación para cada código. De esta manera, el transcompilador determina las partes del código que pueden ser paralelizadas y genera automáticamente la correspondiente versión paralela. La realización de transformaciones source-to-source secuencial-paralelo permite además a los usuarios comparar ambos códigos de forma más simple, facilitando el aprendizaje y la mejora de sus capacidades en programación paralela.

Adicionalmente, es habitual que los programadores centren sus esfuerzos en aquellas instrucciones que pueden ser paralelizadas, sin tener en cuenta la eficiencia del resto del código, a menudo ignorando el importante efecto que las partes secuenciales tienen sobre los tiempos de ejecución. El impacto asociado es especialmente significativo en trabajos HPC que tardan varios días en ejecutarse o que forman parte de proyectos que requieren miles de horas de CPU anuales, algo muy común en variedad de investigaciones científicas. Por ello, esta tesis evalúa y analiza diversas técnicas software con el objetivo de conseguir mejoras adicionales en el código y reducir los tiempos de ejecución. Las técnicas han sido seleccionadas entre la literatura existente y escogidas por ser las más representativas o que mejoran de modo más notable la eficiencia, entendiendo ésta como el rendimiento computacional alcanzado por un programa en relación al alcanzable en una situación óptima. En primer lugar, se realiza un detallado análisis del impacto producido al aplicar las distintas estrategias en dispositivos IoT (Internet of Things, Internet de las cosas), debido a su mayor sencillez para realizar todas las mediciones necesarias. Tras comprobar su impacto en la reducción de los tiempos de ejecución y en el ahorro energético, las mismas estrategias son aplicadas sobre infraestructuras HPC. De este modo se demuestra que las técnicas propuestas permiten aumentar la eficiencia de forma fácil y sencilla en ambos entornos, posibilitando que los usuarios obtengan destacables mejoras de rendimiento con cambios menores en sus códigos.

Con todo lo anterior se consigue que incluso usuarios noveles puedan hacer un uso más apropiado y eficiente de los recursos computacionales, disminuyendo los problemas inherentes al aprendizaje de la programación paralela.

Proyectos relacionado:

- Javier Corral García. *Paralelización Automática y Estrategias de Desarrollo de Código Eficiente para Aumentar el Rendimiento en Centros de Supercomputación*. Tesis Doctoral. Programa de Doctorado en Tecnologías Informáticas (TIN) de la Universidad de Extremadura. Julio de 2021.

Noticias relacionadas:

- Ingeniero de CénitS presenta Tesis Doctoral centrada en aumentar el rendimiento de los centros de supercomputación

[CénitS [1]].

URL del envío:https://www.cenits.es/enlaces/publicaciones/paralelizacion-automatica-estrategias-desarrollo-codigo-eficiente-aumentar

Enlaces
[1] http://www.cenits.es/noticias/22072021-ingeniero-cenits-presenta-tesis-doctoral-centrada-aumentar-rendimiento-centros