

EfiHPC: Entorno Software para el Desarrollo de Códigos de programación eficientes en Centros de Supercomputación

Researchers:

- Javier Corral García, Felipe Lemus Prieto y Juan Manuel Murillo Rodríguez. Fundación [COMPUTAEX](#) [1].
- José Carlos Sancho Núñez. Universidad de Extremadura ([UEX](#) [2]).

Language Undefined

Description:

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. Este tipo de proyectos, ampliamente multidisciplinarios y heterogéneos, demandan una obtención de resultados con tiempos de ejecución realmente críticos. 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 optimizar adecuadamente sus códigos.

La mejora en sus códigos de programación, considerando su eficiencia, 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.

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. En consecuencia, el objetivo principal de este proyecto consiste en evaluar y analizar diversas técnicas software con el objetivo de conseguir mejoras adicionales en el código y reducir los tiempos de ejecución. Las técnicas serán 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. Posteriormente, se realizará un detallado análisis del impacto producido en la reducción de los tiempos de ejecución sobre infraestructuras HPC. De este modo se persigue proponer un amplio conjunto de técnicas que permitan aumentar la eficiencia de forma fácil y sencilla en estos entornos, posibilitando que los usuarios obtengan destacables mejoras de rendimiento con cambios menores en sus códigos. Además, se buscará que, gracias a estas técnicas, incluso usuarios novatos puedan hacer un uso más apropiado y eficiente de los recursos computacionales.

Asimismo, la aplicabilidad de la investigación desarrollada en este proyecto será de especial interés para otras líneas de investigación y proyectos de I+D+i de Cénits-COMPUTAEX, sobre todo en la implementación y el uso de herramientas centradas en la biotecnología aplicada a la salud, destacando su posible utilización sobre algoritmos de sistemas computacionales destinados al diagnóstico, el pronóstico y el desarrollo de terapias avanzadas o dirigidas en medicina personalizada.

Objectives:

Este proyecto está centrado en la escritura de códigos más eficientes y persigue que los usuarios puedan aumentar de forma significativa el rendimiento de sus programas, reduciendo los tiempos de ejecución (y en consecuencia, se posibilite una reducción del consumo energético), gracias a la aplicación de un conjunto de técnicas software. De este modo, se contribuirá además a que los programadores sean conscientes del notable impacto que la aplicación de ciertas técnicas, en pequeños y simples fragmentos de sus códigos, puede tener sobre la eficiencia.

Estas mejoras en el rendimiento pueden resultar especialmente significativas en relación a trabajos de HPC que tardan días o incluso semanas en completarse o que forman parte de proyectos que requieren miles de horas de CPU al año, algo realmente común en una amplia variedad de investigaciones científicas. En este sentido, es muy importante que los programadores sean conscientes del excesivo tiempo que puede malgastar un pequeño fragmento de código ineficiente cuando su ejecución es repetida durante innumerables ocasiones.

Adicionalmente, es habitual que cuando programadores no expertos se enfrentan a la programación HPC presenten notables dificultades para optimizar sus códigos, aunque puedan identificar correctamente las partes donde la ejecución requiere más tiempo de proceso gracias a profilers o herramientas de análisis. Además, estos programadores suelen centrar sus esfuerzos únicamente en las instrucciones que se pueden paralelizar, sin tener en cuenta la eficiencia de las partes secuenciales y, a menudo, ignorando el considerable impacto que éstas tienen en el tiempo final de ejecución o en el consumo energético asociado.

Sin embargo, en la literatura no existen demasiadas propuestas destinadas a optimizar los tiempos de ejecución mediante la aplicación de técnicas software que posibiliten la escritura de código eficiente. El presente trabajo pretende comprar además, el impacto de dicho problema tanto en grandes infraestructuras de HPC, como en otros dispositivos como los empleados en el Internet de las cosas (IoT, Internet of Things).

Hipótesis de partida

Este proyecto persigue proponer una serie de técnicas para la escritura eficiente de programas en las infraestructuras HPC, preservando la semántica del programa. Para ello se evaluarán los resultados obtenidos mediante la aplicación manual de estas técnicas, en comparación con los correspondientes a la utilización de las opciones de optimización ofrecidas por los compiladores. La hipótesis de partida considera que estas técnicas lograrán importantes reducciones en los tiempos de ejecución, incluso mejorando el impacto producido por las optimizaciones automáticas ofrecidas por los compiladores. Esto pondrá de manifiesto que estas últimas no permiten alcanzar las mismas mejoras que los programadores pueden lograr mediante la aplicación de las técnicas a desarrollar.

Objetivo general

El principal objetivo de este proyecto se centra en la escritura de códigos más eficientes y persigue que los usuarios puedan aumentar de forma significativa el rendimiento de sus programas, reduciendo los tiempos de ejecución (y en consecuencia se posibilite también una reducción del consumo energético), gracias a la aplicación de un conjunto de técnicas software. De este modo, se contribuirá además a que los programadores sean conscientes del notable impacto que la aplicación de ciertas técnicas, en pequeños y simples fragmentos de sus códigos, puede tener sobre la eficiencia.

Se plantea de este modo desarrollar un conjunto de estrategias y pautas que permitan a los programadores alcanzar mejoras realmente significativas en sus códigos, mediante la aplicación de una serie de pequeñas modificaciones y empleando un esfuerzo mínimo. El objetivo es mejorar el rendimiento, consiguiendo reducir lo máximo posible los tiempos de ejecución en los centros de cómputo. En este sentido, es necesario considerar el destacable efecto negativo que simples fragmentos de código pueden presentar sobre los tiempos de ejecución, debido a la importante limitación que la fracción secuencial de un programa ejerce sobre el *speedup* final alcanzado.

Asimismo, se pretende que los programadores comprendan el importante impacto que la utilización de ciertas técnicas puede tener en el rendimiento final de sus aplicaciones, especialmente en aquellas que están en continua ejecución o bien consumen un número considerablemente elevado de horas de CPU. En este sentido, algunos programadores no son conscientes, por ejemplo, de la significativa cantidad de tiempo que puede llegar a ser empleada por un pequeño fragmento de código ineficiente, especialmente cuando la ejecución de éste es repetida en numerosas ocasiones.

Funding sources:

La Fundación [COMPUTAEX](#) [1], como beneficiaria del programa de ayudas destinadas a la financiación de los programas de empleo creados para la ejecución del [Plan de Recuperación, Transformación y Resiliencia en la Comunidad Autónoma de Extremadura](#) [3], al amparo del Decreto 137/2021, de 15 de diciembre, por el que se establecen las bases reguladoras para la concesión de subvenciones destinadas a la financiación de los programas de empleo creados para la ejecución del [Plan de Recuperación, Transformación y Resiliencia](#) [4], financiado por la Unión Europea - [NextGenerationEU](#) [5], en la Comunidad Autónoma de Extremadura.



Financiado por
la Unión Europea
NextGenerationEU



GOBIERNO
DE ESPAÑA
MINISTERIO
DE TRABAJO
Y ECONOMÍA SOCIAL



Plan de
Recuperación,
Transformación
y Resiliencia

JUNTA DE EXTREMADURA

SERVICIO PÚBLICO
DE EMPLEO ESTATAL
SEPE

SEXPE
Servicio Extremeño
Público de Empleo

Source

URL: <https://web.computaex.es/en/proyectos/efihpc-entorno-software-desarrollo-codigos-programacion-eficientes-centros>

Links

[1] <https://web.computaex.es/fundacion> [2] <https://www.unex.es/> [3] https://extremaduratrabaja.juntaex.es/ayudas_plan_recuperacion_transformacion_risilencia [4] <https://planderecuperacion.gob.es/> [5] https://next-generation-eu.europa.eu/index_es