ANÁLISIS DE RENDIMIENTO DE APLICACIONES Y SISTEMAS COMPLEJOS MEDIANTE COMPUTACIÓN PARALELA Y TÉCNICAS DE OPTIMIZACIÓN DE RECURSOS

Miguel Méndez-Garabetti[1,2,3,4,5], Eduardo Piray[1,3,4], Javier Roseinstein[2], Ignacio Picotto[1], Rodrigo Elgueta[2,3,5], Marcos Benegas[2], Natalia Anahí Magris[1], Andrea Cabrera[2], Manuel Battaglia[2], Agustín Giorlando[3,5], Gabriel Nasiff[5], Jonathan Nicolás Guerra Moronta[5]

[1]*Universidad Siglo 21, Córdoba, Argentina.*

[2]*Facultad de Informática y Diseño, Universidad Champagnat, Mendoza, Argentina.*

[3]*Free and Open Source Software/Hardware Research Laboratory (FOSSHLab), Argentina.*

[4]*Departamento de Sistemas, Universidad CAECE, Mar del Plata, Argentina.*

[5]*Dirección de Posgrados, Facultad de Ingeniería, Universidad de Mendoza, Mendoza, Argentina.*

mmendezgarabetti@gmail.com, edupiray@gmail.com, roseinstein.javier@uch.edu.ar, ignaciopicotto@gmail.com, rodrigo.elgueta@uch.edu.ar, nmagris@outlook.com, andreabcabrera@gmail.com, manubattaglia94@gmail.com, ganasiff@gmail.com, j.guerra@alumno.um.edu.ar

RESUMEN

El crecimiento exponencial de la potencia de las tecnologías de procesamiento, almacenamiento, comunicaciones ha permitido que los sistemas de computación de alto rendimiento aborden problemas científicos que son cada vez más grandes y complejos. Estas aplicaciones, que son diseñadas para ejecutarse en plataformas de hardware específicas, suelen tener diversos requerimientos donde la capacidad de procesamiento no necesariamente es el principal recurso que determina el rendimiento de éstas. En ocasiones otros recursos, como la capacidad de E/S, la memoria y el ancho de banda suelen ser protagonistas. En este sentido, esta línea de investigación plantea indagar en diferentes aristas de la computación paralela, tales como el diseño de técnicas de optimización de recursos basados de plataformas de HPC.

Palabras claves: Performance. Mejora de rendimiento. Optimización de recursos. Computación de alto rendimiento. Arquitecturas paralelas.

CONTEXTO

La presente línea de investigación consiste en una iniciativa encabezada por el FOSSHLab (Free and Open Source Software/Hardware Research Laboratory), institución que impulsa el desarrollo de proyectos centrados en tecnologías de software y hardware libre. Actualmente la línea cuenta con tres proyectos de investigación llevados a cabo en universidades diferentes: 1) Universidad Siglo 21, en el marco de la carrera Licenciatura en Informática (resolución en trámite), 2) Universidad Champagnat, entorno a la Licenciatura en Sistemas de Información (resolución 59-2022-UCH), 3) Universidad CAECE, en el marco de la Licenciatura en Sistemas, la Ingeniería en Sistemas y la Licenciatura en Gestión de Sistemas y

Negocios (resolución en trámite) y 4) Universidad de Mendoza, en el marco de la Ingeniería en Informática y la Maestría en Teleinformática (resolución 36/2021). Cada proyecto está financiado individualmente por su respectiva universidad.

1. INTRODUCCIÓN

Desde el origen de la computación ha habido un constante y arduo esfuerzo por conseguir mayores prestaciones computacionales. Este desarrollo ha generado grandes logros, materializados en avances significativos en el hardware y software de computadoras, haciendo cada vez más rápidos y potentes los sistemas de procesamiento, almacenamiento y comunicaciones.

Este incremento de poder computacional ha permitido abordar problemas cada vez más grandes y complejos, aunque, para una gran cantidad de éstos, aún sigue siendo necesario contar con mayor capacidad de procesamiento. Este tipo de requerimientos ha desembocado en la utilización de múltiples procesadores agrupados para trabajar de forma conjunta. Esta práctica se conoce como Computación de Alto Rendimiento (HPC, High Performance Computing), concepto también asociado al de Computación Paralela [1]. En términos generales, se puede decir que el HPC consiste en la utilización de determinada cantidad de elementos de procesamiento en la resolución de un problema de forma cooperativa, tratando de minimizar el tiempo de procesamiento y dar así una respuesta lo más rápido posible.

Aprovechar al máximo el potencial de estas plataformas es un gran desafío debido a las complejas interacciones que se producen entre el hardware y el software, sumado a la complejidad de las aplicaciones que se deben ejecutar en estos entornos, siendo que muchas de ellas no han sido diseñadas específicamente para ser ejecutadas en estas plataformas. Es aquí donde el análisis de rendimiento, revisión de diseño y arquitectura de aplicaciones juega un rol central en el proceso de desarrollo de aplicaciones paralelas.

A medida que la tecnología avanza, el costo de los sistemas de HPC ha disminuido significativamente, lo que ha permitido que cada vez más organizaciones y empresas puedan acceder a estas capacidades. Sin embargo, el diseño y la optimización de aplicaciones para sistemas de HPC sigue siendo un desafío importante, ya que requiere conocimientos especializados tanto en el hardware como en el software.

En este contexto, es común observar que este tipo de soluciones requieren cada vez más capacidad de cómputo, comunicación y almacenamiento; si se desea dar respuesta en un plazo de tiempo durante el cual los resultados aún sigan teniendo validez. Es importante tener en consideración, que el aumento de recursos computacionales no implica de forma transparente un incremento en el rendimiento y, además, lograr mantener en funcionamiento plataformas HPC de gran escala generalmente implica un gran consumo de energía. Por lo tanto, es imprescindible además de esforzarse por mejorar el rendimiento, lograr optimizar la utilización de recursos. En otras palabras, resulta indispensable considerar la necesidad de abordar simultáneamente la consecución de dos objetivos: a) maximizar el rendimiento y b) minimizar el consumo energético.

En el desarrollo de aplicaciones paralelas es necesario contar con el manejo de aspectos propios del paralelismo, los que deben ser tenidos en cuenta para que una aplicación logre los objetivos principales de obtener el resultado esperado y el de alcanzar un desempeño acorde al entorno de ejecución [2]. Debido a esto, una vez que se ha implementado y testeado la aplicación, es necesario efectuar un análisis de rendimiento para determinar si coincide con el esperado. Si bien existen diferentes métricas para determinar el

desempeño de una aplicación paralela, (como el tiempo de ejecución, eficiencia, speed-up, balanceo de carga, entre otras) es necesario contar con herramientas que ofrezcan la información necesaria para determinar qué características deben ser mejoradas para lograr mejor rendimiento. La revisión y ajuste de una aplicación representa una parte importante y necesaria en el desarrollo de programas, ya que permite adaptar y mejorar el comportamiento de las aplicaciones mediante la detección de aquellos aspectos que impiden alcanzar el rendimiento esperado, y mediante la definición y aplicación de acciones tendientes a mejorar dichos aspectos [3].

En los clústeres de computadoras homogéneos la programación se lleva a cabo mediante la librería de paso de mensajes (MPI, Message Passing Inteface) [4], además, dentro de cada nodo individual, generalmente con arquitecturas multicore se puede utilizar OpenMP [5], bajo el paradigma de memoria compartida. Ambos enfoques pueden utilizarse de forma conjunta o híbrida, con el propósito de incrementar el nivel de paralelismo en función del tipo de problema que se intente resolver y la plataforma de hardware subyacente disponible.

En esta línea, también es posible utilizar estos enfoques con otro basado principalmente en memoria compartida, que es compatible con el de memoria compartida-distribuida, lo que se conoce como computación heterogénea. Dicha implementación consta del uso de paralelismo mediante unidades de procesamiento gráfico de propósito general (GPGPU, General Purpose Graphics Processing Units) mediante CUDA (Compute Unified Device Architecture), la arquitectura de cálculo paralelo de NVIDIA [6].

En la literatura podemos observar numerosos antecedentes que dan cuenta que el uso de la computación paralela permite incrementar el rendimiento de una aplicación. En [7] se propone la paralelización de cuatro aplicaciones científicas diseñadas con una programación totalmente secuencial. Se aplica una metodología sistemática para transformar un código base antiguo en código moderno, es decir, código paralelo y robusto.

En [8] [9], la utilización de la computación paralela/distribuida queda en clara evidencia su importancia, ya que es el medio que permite desarrollar sistemas de predicción de fenómenos naturales donde sus resultados lleguen a tiempo para poder ser tomados para la toma de decisiones.

2. LÍNEAS DE INVESTIGACIÓN Y DESARROLLO

A continuación, se describen brevemente cada línea de investigación:

1. Una de estas líneas de investigación se enfoca en la optimización de metaheurísticas paralelas de forma adaptativa. Este enfoque busca mejorar la calidad de la búsqueda y la eficiencia del algoritmo mediante la adaptación de los parámetros de la metaheurística de acuerdo con las características del problema en cuestión. De esta manera, se espera obtener una mayor capacidad de exploración del espacio de búsqueda y una mayor robustez en la resolución de problemas complejos.

2. Otra línea de investigación se centra en el estudio de las arquitecturas de hardware paralelo y sus implicancias en el rendimiento de los modelos. En este sentido, se busca comprender cómo las características de las arquitecturas de hardware, como el número de núcleos, la memoria caché y la velocidad de la comunicación entre ellos, afectan el rendimiento de los modelos y cómo se pueden optimizar para obtener mejores resultados. Esta investigación puede tener importantes

implicancias en la selección de la mejor arquitectura de hardware para aplicaciones específicas.

3. Finalmente, otra línea de investigación está relacionada con la generación de un modelo de programación centrado en el rendimiento y la optimización de recursos. Esta investigación busca desarrollar nuevas técnicas y herramientas de programación que permitan a los desarrolladores crear aplicaciones paralelas más eficientes en términos de rendimiento y uso de recursos. De esta manera, se espera poder aprovechar al máximo el potencial de las arquitecturas de hardware paralelo y mejorar la eficiencia de los modelos y otras aplicaciones similares.

3. FORMACIÓN DE RECURSOS HUMANOS

El equipo de trabajo de la presente línea de I+D+i cuenta con la dirección del Dr. Miguel Mendez-Garabetti y la codirección del Mgter. Javier Roseinstein. Además, participan cuatro docentes investigadores por las diferentes universidades, uno de ellos con presentación en curso de plan de trabajo doctoral en el marco de la presente línea. En relación a estudiantes de grado, se cuenta con tres estudiantes de grado, y un estudiante de posgrado.

4. RESULTADOS OBTENIDOS/ESPERADOS

Existen diferentes resultados esperados relacionados a cada proyecto, los cuales confluyen al resultado esperado de la línea de I+D+i general.

Por un lado, se espera desarrollar metaheurísticas paralelas de forma adaptativa, que permitan mejorar la calidad de la búsqueda y la eficiencia al ajustar los parámetros de acuerdo con las características específicas de cada problema. El resultado esperado es una mayor capacidad de exploración del espacio de búsqueda y una mayor capacidad de resolución de problemas complejos, lo que permitirá obtener soluciones más precisas y robustas en una variedad de aplicaciones prácticas.

En relación a la segunda línea de investigación, el resultado esperado consiste en obtener una comprensión detallada de cómo las características de las arquitecturas de hardware paralelo afectan el rendimiento y cómo se pueden optimizar para obtener mejores resultados. Además, se espera que este proyecto de investigación contribuya al desarrollo de nuevas arquitecturas de hardware que sean más eficientes y eficaces.

Y, por último, otro resultado esperado se relaciona con la propuesta de un modelo de programación centrado en el rendimiento y la optimización de recursos que crear aplicaciones paralelas más eficientes en términos de rendimiento y uso de recursos. Se espera que la investigación contribuya al desarrollo de nuevas técnicas y herramientas de programación que puedan ser utilizadas por la comunidad de desarrolladores de manera amplia y que permitan crear aplicaciones paralelas más eficientes y escalables en el futuro.

5. BIBLIOGRFÍA

[1] L. N. Long, “Parallel Computing”, *Seminar*, p. 134, 2008.

[2] J. J. Dongarra, *The sourcebook of parallel computing*, vol. 41. Morgan Kaufmann Publishers, 2002. doi: 10.5860/choice.41-0348.

[3] R. Suda, K. Naono, K. Teranishi, y J. Cavazos, “Software automatic tuning: Concepts and state-of-the-art results”, en *Software Automatic Tuning: From Concepts to State-of-the-Art Results*,

Springer New York, 2010, pp. 3–15. doi: 10.1007/978-1-4419-6935-4_1.
[4] A. Grama, A. Gupta, G. Karypis, y V. Kumar, *Introduction to Parallel Computing*. Harlow, England ; New York, 2003.
[5] B. Chapman, G. Jost, R. V. D. Pas, R. V. D. Pas, W. Gropp, y E. Lusk, *Using OpenMP – Portable Shared Memory Parallel Programming*. Cambridge, Mass: MIT Press, 2007.
[6] S. Tsutsui y P. Collet, Eds., *Massively Parallel Evolutionary Computation on GPGPUs*. New York, 2013.
[7] M. Aldinucci *et al.*, "Practical Parallelization of Scientific Applications with OpenMP, OpenACC and MPI", *J. Parallel Distrib. Comput.*, vol. 157, jun. 2021, doi: 10.1016/j.jpdc.2021.05.017.
[8] M. Méndez-Garabetti, G. Bianchini, y P. Caymes-Scutari, "HESS-IM: A Uncertainty Reduction Method that Integrates Remote Sensing Data Applied to Forest Fire Behavior Prediction", *Commun. Comput. Inf. Sci.*, p. 17, 2021.
[9] M. L. Tardivo, P. Caymes-Scutari, G. Bianchini, M. Méndez-Garabetti, A. Cencerrado, y A. Cortés, "A comparative study of evolutionary statistical methods for uncertainty reduction in forest fire propagation prediction", *Procedia Comput. Sci.*, vol. 108, pp. 2018–2027, ene. 2017, doi: 10.1016/j.procs.2017.05.252.