Modelos de implementación de aplicaciones paralelas centradas en la optimización de recursos: una revisión de literatura

Miguel Méndez-Garabetti[1,2,3], Eduardo Piray[1,2], Natalia Magris[1,2] y Rodolfo Schmidt[1,2]

[1] Universidad Siglo 21, Córdoba, Argentina.
[2] Free and Open Source Software/Hardware Research Laboratory (FOSSHLab), Argentina.
[3] Facultad de Informática y Diseño, Universidad Champagnat, Mendoza, Argentina.
miguel.mendez@ues.21.edu.ar, edupiray@gmail.com, nmagris@outlook.com, rodolfoschmidt9@gmail.com

Abstract. En este estudio se presenta una revisión de literatura enfocada en modelos de implementación de aplicaciones paralelas orientadas a la optimización de recursos en el ámbito de la computación de alto rendimiento. Se analizan investigaciones clave y enfoques teóricos resaltando la relevancia de una adecuada gestión de recursos en el diseño y aplicación del paradigma paralelo en sistemas y aplicaciones computacionalmente complejas. Además, se examinan métricas y metodologías para medir la eficacia y eficiencia de los modelos propuestos. Esta revisión contempla los desafíos y obstáculos en la implementación de soluciones, como la escalabilidad, la heterogeneidad de recursos y las limitaciones intrínsecas a los sistemas y aplicaciones examinados. Este análisis ofrece un marco sólido para investigaciones futuras en computación paralela y distribuida, identificando áreas de mejora y desafíos en la implementación de modelos enfocados en la optimización de recursos, sentando las bases para el desarrollo de soluciones innovadoras y eficientes en la resolución de problemas de gran envergadura y complejidad.

Keywords: computación paralela · optimización de recursos · computación de alto rendimiento · arquitecturas de hardware · aplicaciones complejas.

1 Introducción

En la era actual de la computación, la demanda de soluciones eficientes y rápidas para abordar problemas complejos y de gran envergadura ha llevado al crecimiento exponencial de la computación de alto rendimiento (HPC, por sus siglas en inglés) [1]. Los clústeres de productos básicos revolucionaron la computación de alto rendimiento cuando aparecieron por primera vez hace dos décadas. A medida que la escala y la complejidad han aumentado, han surgido nuevos desafíos en confiabilidad y resiliencia sistémica, eficiencia energética y optimización, y la complejidad del software que sugiere la necesidad de reevaluar los enfoques actuales [2].

Como respuesta a este fenómeno, el presente artículo proporciona un acercamiento al estado del arte actual y futuro respecto a modelos y enfoques teóricos en la implementación de aplicaciones paralelas enfocadas en la optimización de recursos, utilizando conocimientos y perspectivas extraídos a través de una revisión de literatura y debates de la comunidad. A través del análisis de investigaciones clave y la discusión de la relevancia de una gestión de recursos adecuada en el diseño y aplicación del paradigma paralelo, se busca ofrecer un panorama completo sobre los desafíos y obstáculos presentes en el campo del HPC, incluyendo los retos hacia la construcción de sistemas de computación trans-exaescale [3]. Adicionalmente, se aborda el estudio de métricas y metodologías para evaluar la eficacia y eficiencia de los modelos propuestos, así como los retos relacionados con la escalabilidad, heterogeneidad de recursos y limitaciones intrínsecas a los sistemas y aplicaciones estudiadas. Este análisis pretende proporcionar un marco sólido para futuras investigaciones en el ámbito de la computación paralela y distribuida, identificando áreas de mejora y desafíos en la implementación de modelos que prioricen la optimización de recursos con el objetivo de impulsar el desarrollo de soluciones innovadoras y eficientes en la resolución de problemas de alta complejidad. El artículo continúa de la siguiente manera, en la sección siguiente se presenta el estado del arte en tendencias actuales referidas a HPC enfocados en la optimización de recursos, y en la sección 3 se brinda una breve conclusión de cara al trabajo en curso y futuro enfocado al desarrollo de un revisión exhaustiva.

2 Estado del arte

Este estudio parcial se centra en analizar la literatura vinculada a dos aspectos clave, 1) los desafíos y tendencias el área de la HPC y 2) los modelos actuales de programación orientadas a la computación de exaescala, que se describen a continuación.

Para el primero, en [4] se presentaron los principales desafíos a los cuales se enfrenta HPC cuando esta alcanza capacidades del orden de exaFLOPS, los cuales abarcan: la eficiencia energética, la tolerancia a fallas, la complejidad del software y el volumen de datos. En términos de eficiencia energética los sistemas de exaescala están alcanzando el objetivo de consumo de energía del orden de 20MW por un exaFLOP. De hecho, la supercomputadora Frontier [5] (líder del TOP500) logra una eficiencia energética entre 14,5MW y 15MW por exaFlop. En cuanto a tolerancia a fallas, se sigue investigando fallas en lo que respecta a protocolos checkpoint-restart, detección de corrupción de datos y comprensión de fallas [6]. Respecto a la complejidad del software se identifica una falta de modelos de programación adecuados que simplifiquen el desarrollo de aplicaciones científicas escalables, en el Exascale Computing Project (ECP) se está trabajando desde 2016 en este aspecto (ampliado en la próxima sección). Finalmente, para el desafío de volumen de datos, se resalta una brecha creciente entre la potencia de cálculo de los procesadores y el ancho de banda de datos, y que la nueva tecnología de hardware (i.e., aceleradores, memorias, redes, etc.)

son soluciones prometedoras para el desafío del movimiento de datos, pero entran en conflicto con el desafío de la complejidad del software, ya que introducen una complicación adicional al programar estos sistemas.

Para el segundo, de lo descrito anteriormente se advierte la importancia de manejar la complejidad del software mediante el desarrollo de modelos de programación adecuados en la era de la exaescala. En [7] se discute el desarrollo de un marco unificado y estandarizado para los modelos de programación Partitioned Global Address Space (PGAS), con el objetivo de facilitar a los desarrolladores la escritura de código para ejecutarse en una variedad de sistemas, principalmente para la supercomputación de exaescale. Por otro lado, en [8] se abordan los lenguajes de programación y los modelos utilizados en el código base de las aplicaciones derivadas del ECP, los cuales eran utilizados para optimizar el rendimiento en las futuras plataformas de hardware exaescala. En el ECP se definen 62 códigos de aplicación que se implementan en tres lenguajes de alto nivel (C, C++ y Fortran) y usan 22 combinaciones de modelos de programación de unidades de procesamiento gráfico. El lenguaje más utilizado en los códigos de aplicación es C++, y los modelos de programación más comunes en las aplicaciones ECP son CUDA y Kokkos [9]. El ECP está desarrollando un ecosistema de software de soporte completo que consta de solucionadores matemáticos, de visualización, lineales y no lineales, utilidades y bibliotecas de ajuste de rendimiento.

El ECP es un proyecto formal con métricas cuantitativas para el éxito que se miden a través de parámetros clave de rendimiento (KPP, por su siglas en inglés) definidos por el proyecto. Los proyectos de desarrollo de aplicaciones se agrupan en dos categorías de objetivos, denominadas genéricamente como primer y segundo KPP (KPP-1 y KPP-2). Las aplicaciones KPP-1 tienen una figura de mérito de rendimiento cuantitativo (FOM, por su siglas en inglés). Los FOM se definen como una proporción de las tasas de trabajo de rendimiento en la plataforma actual en relación con una medición de referencia desde el inicio del proyecto. El objetivo del KPP-1 es que el 50% de las aplicaciones logren un FOM$\geq$50 en sus problemas de desafío definidos. La segunda clase de aplicaciones se clasifica como proyectos KPP-2. Esta métrica tiene como objetivo evaluar la creación de nuevas capacidades científicas y de ingeniería que puedan explotar al máximo los recursos de exaescala. Al final del proyecto, el 50% de estas aplicaciones deben demostrar estas capacidades en sus problemas de desafío del proyecto. Todos los proyectos se verifican en computadoras de alto rendimiento donde se evalúan los niveles de mejora respecto de las métricas mencionadas. Según el análisis en [8], el desarrollo de aplicaciones de ECP apoya 62 códigos en 24 proyectos para adaptarlas a plataformas de exaescala desde 2023. Más de la mitad de las aplicaciones KPP-1 han logrado mejoras de rendimiento establecidas. Se aprecian tendencias como la contracción de lenguajes de implementación y la proliferación de modelos de programación. Fortran disminuye debido a la diversidad de hardware de GPU, mientras que C++ aumenta. Los modelos de programación de GPU no tienen favoritos claros, lo que indica que la elección se basa en los requisitos algorítmicos de la aplicación y no existe un enfoque

unificado para todos los dominios de la computación científica. Finalmente, respecto a mejoras que introduce la computación de exaescala en la ciencia, en [10] se pueden observar diferentes puntos de vista donde dan manifiesto sobre como la computación de exaescala podría proporcionar un impulso importante para cerrar aún más la brecha entre las observaciones experimentales y el modelado.

3 Conclusiones y Trabajo Futuro

Este trabajo presenta un análisis en desarrollo de la literatura de la HPC, subrayando la implementación de aplicaciones paralelas para optimizar recursos. Se identifican desafíos, destacando la importancia de un marco unificado para modelos de programación, que facilite el desarrollo de aplicaciones en sistemas diversos, especialmente para la supercomputación de exaescala. El estudio completo podrá revelar cómo la computación de exaescala puede cerrar la brecha entre observaciones experimentales y modelado en diversas disciplinas científicas.

Referencias

1. R. Buyya, *High Performance Cluster Computing: Architectures and Systems*. Prentice Hall, PTR, NJ, USA, 1999.
2. A. Geist and D. A. Reed, "A survey of high-performance computing scaling challenges," *Original Article The International Journal of High Performance Computing Applications*, vol. 31, no. 1, pp. 104–113, 2017.
3. B. S. Allen, M. A. Ezell, P. Peltz, D. Jacobsen, E. Roman, C. Lueninghoener, and J. L. Wofford, "Modernizing the HPC System Software Stack," jul 2020.
4. S. Heldens, P. Hijma, B. V. Werkhoven, J. Maassen, A. S. Belloum, and R. V. Van Nieuwpoort, "The landscape of exascale research: A data-driven literature analysis," *ACM Computing Surveys (CSUR)*, vol. 53, no. 2, pp. 1–43, 2020.
5. V. Rajaraman, "Frontier—world's first exaflops supercomputer," *Resonance*, vol. 28, no. 4, pp. 567–576, 2023.
6. M. Morán, J. Balladini, D. Rexachs, and E. Rucci, "Towards Management of Energy Consumption in HPC Systems with Fault Tolerance," in *2020 IEEE Congreso Bienal de Argentina (ARGENCON)*, pp. 1–8, Dec. 2020.
7. W. Lu, *Interoperable PGAS Programming Models for Exascale Supercomputing*. PhD thesis, State University of New York at Stony Brook, 2022.
8. T. M. Evans, A. Siegel, E. W. Draeger, J. Deslippe, M. M. Francois, T. C. Germann, W. E. Hart, and D. F. Martin, "A survey of software implementations used by application codes in the exascale computing project," *The International Journal of High Performance Computing Applications*, vol. 36, no. 1, pp. 5–12, 2022.
9. C. R. Trott, D. Lebrun-Grandié, D. Arndt, J. Ciesko, V. Dang, N. Ellingwood, R. Gayatri, E. Harvey, D. S. Hollman, D. Ibanez, N. Liber, J. Madsen, J. Miles, D. Poliakoff, A. Powell, S. Rajamanickam, M. Simberg, D. Sunderland, B. Turcksin, and J. Wilke, "Kokkos 3: Programming Model Extensions for the Exascale Era," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, pp. 805–817, Apr. 2022. Conference Name: IEEE Transactions on Parallel and Distributed Systems.
10. C. Chang, V. L. Deringer, K. S. Katti, V. Van Speybroeck, and C. M. Wolverton, "Simulations in the era of exascale computing," *Nature Reviews Materials*, pp. 1–5, 2023.