

Desafíos en la integración de QPUs a sistemas HPC clásicos

Challenges in integrating QPUs into classical HPC systems

Desafios na integraçao de QPUs em sistemas HPC clássicos

Adrian Pousa ^{1,(*)}, Victoria Sanz ²

Recibido: 10/10/2024

Aceptado: 10/10/2024

Resumen. - El Cómputo de Altas Prestaciones (HPC) aplica diferentes técnicas a aplicaciones complejas o de gran volumen, apoyándose en el software y hardware paralelo, para reducir el tiempo de ejecución respecto a ejecutar las mismas sobre una computadora simple. Por otro lado, la Computación Cuántica (QC) surge como un nuevo paradigma que aprovecha propiedades de la mecánica cuántica en computación. La QC posee características inherentemente paralelas y se espera que resuelva algunos problemas de forma más rápida que la computación clásica. Varios autores coinciden que la QC deberá integrarse al cómputo clásico HPC. El propósito de este artículo es abordar los desafíos de esta integración.

Palabras clave: Computación de Altas Prestaciones; Computación Cuántica; Integración HPC-QC.

(*) Autor de correspondencia

¹ Docente e Investigador, III-LIDI, Facultad de Informática, UNLP, La Plata, Argentina, apousa@lidi.info.unlp.edu.ar, ORCID iD: <https://orcid.org/0000-0001-8072-6626>

² Docente e Investigador, III-LIDI, Facultad de Informática, UNLP, La Plata, Argentina, CIC, Buenos Aires, Argentina, vsanz@lidi.info.unlp.edu.ar, ORCID iD: <https://orcid.org/0000-0002-3487-406X>

Summary. - High Performance Computing (HPC) applies different techniques to complex or high-volume applications, relying on parallel software and hardware, to reduce the execution time compared to running them on a simple computer. On the other hand, Quantum Computing (QC) emerges as a new paradigm that takes advantage of properties of quantum mechanics in computing. QC has inherently parallel characteristics and is expected to solve some problems faster than classical computing. Several authors agree that QC should be integrated into classical HPC computing. The purpose of this article is to address the challenges of this integration.

Keywords: High Performance Computing; Quantum Computing; HPC-QC Integration.

Resumo. - A Computação de Alto Desempenho (HPC) aplica diferentes técnicas a aplicações complexas ou de grande volume, contando com software e hardware paralelos, para reduzir o tempo de execução em comparação com a execução em um computador simples. Por outro lado, a Computação Quântica (QC) surge como um novo paradigma que aproveita propriedades da mecânica quântica na computação. O QC tem características inerentemente paralelas e espera-se que resolva alguns problemas mais rapidamente do que a computação clássica. Vários autores concordam que o CQ deve ser integrado na computação HPC clássica. O objetivo deste artigo é abordar os desafios dessa integração.

Palavras-chave: Computação de Alto Desempenho; Computação Quântica; Integração HPC-QC.

1. Introducción. - El Cómputo de Altas Prestaciones (HPC ó High Performance Computing) es la práctica de sumar la potencia computacional de varios dispositivos con el fin de mejorar el rendimiento de aplicaciones de las ciencias, ingeniería o negocios. Para ello se aplican distintas técnicas de programación para crear algoritmos paralelos, que se ejecutarán sobre arquitecturas de cómputo paralelas, permitiendo así reducir el tiempo de ejecución de la aplicación respecto a ejecutarla en una única computadora. (1; 2; 3)

La Computación Cuántica (QC ó Quantum Computing) comenzó a desarrollarse en la década de 1980 a partir de las propuestas de Paul Benio, Richard Feynman y David Deutsch de aprovechar las leyes de la mecánica cuántica (QM ó Quantum Mechanics) en computación. Entre sus propuestas destacan la viabilidad de las computadoras cuánticas (es decir, computadoras que operen bajo las leyes de la QM) y la posibilidad de usar dichos sistemas para realizar cómputo que resulta difícil de resolver sobre computadoras clásicas (por ejemplo, simulación de sistemas cuánticos). (4)

En QC, la unidad de información es el quantum bit (qubit). A diferencia de un bit clásico, que sólo puede tomar los valores 0 ó 1, un qubit puede tomar el valor 0, 1 o ambos a la vez. Esto es, el qubit puede encontrarse en una superposición de dos estados cuánticos. Esta superposición cuántica posibilita un procesamiento paralelo a gran escala. Intuitivamente, aquellos cómputos que necesiten ser aplicados a ambos estados (0 y 1) podrán resolverse con una única operación usando un qubit en una computadora cuántica, en contraste a una computadora clásica que necesitará realizar dos operaciones sobre los dos estados por separado. (4; 5; 6)

Dada la naturaleza paralela de las computadoras cuánticas y la posibilidad de que puedan resolver problemas de forma más rápida que las computadoras clásicas, es inevitable establecer una relación entre HPC y QC. Varios autores plantean una integración de la QC en sistemas HPC clásicos. En este artículo analizamos los desafíos de esta integración.

El resto del artículo está organizado de la siguiente manera. La sección 2 describe brevemente los sistemas HPC y su evolución. La sección 3 describe la QC y su estado actual. La sección 4 resume las propuestas de otros autores que relacionan HPC con QC. La sección 5 analiza los desafíos de dicha integración. Para analizar, la sección 6 presenta la conclusión y las líneas de trabajo futuro.

2. Evolución de sistemas HPC clásicos. - La idea de agregar la potencia computacional de varios procesadores y utilizarla para reducir el tiempo de resolución de problemas de gran complejidad se remonta a 1971. (7) Sin embargo, los primeros sistemas HPC tipo cluster se popularizaron recién a principios de 1990, cuando el hardware de cómputo y comunicación se volvió más asequible, y gracias a la consolidación del protocolo TCP/IP y al surgimiento de la biblioteca de desarrollo PVM (8) (posteriormente reemplazada por MPI (9)). Esta sección describe la evolución de los sistemas HPC, desde sus comienzos hasta la actualidad.

2.1. Clusters HPC. - Un cluster HPC es una arquitectura que conecta un conjunto de computadoras (nodos) mediante una red de comunicaciones (Figura 1a), permitiendo así sumar la potencia de cómputo de las mismas y utilizarla para resolver problemas en menor tiempo. La red de interconexión puede ser Ethernet o redes de alta velocidad tales como Infiniband o Myrinet. Las aplicaciones que corren sobre un cluster siguen el modelo de programación sobre memoria distribuida. En dicho modelo se generan procesos, uno por nodo del cluster, que interactúan entre sí intercambiando mensajes con el objetivo de resolver el problema en cuestión. La herramienta más usada para desarrollar aplicaciones bajo este modelo es MPI.

Un cluster es un recurso compartido por varios usuarios. Cada usuario puede utilizar todos los nodos del cluster o un subconjunto, pero siempre de forma exclusiva. Así, la aplicación de un usuario ejecuta sobre sus nodos sin ser interferida por aplicaciones de otros usuarios. Cuando un usuario utiliza un subconjunto de nodos del cluster, los nodos restantes pueden ser asignados a otros usuarios.

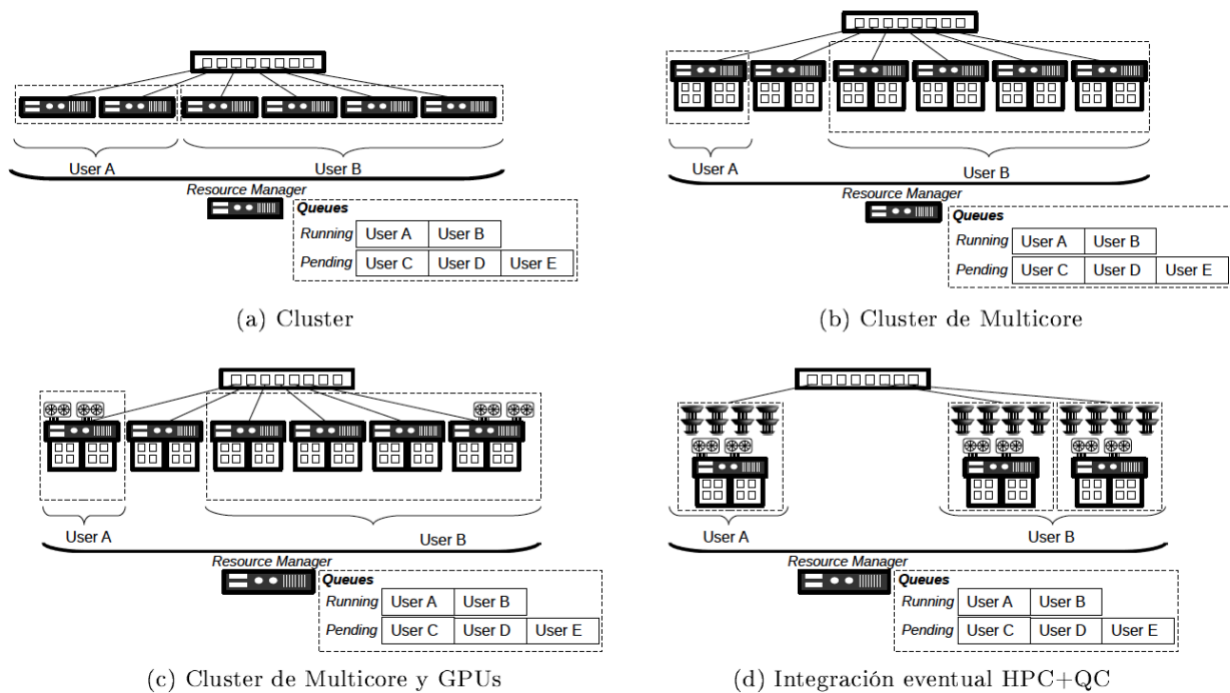


Figura 1. Sistemas HPC

Para gestionar los recursos del cluster se utilizan Resource Managers como OpenPBS (10), Torque (11), Slurm (12), entre otros. Los usuarios solicitan recursos al manager para ejecutar su aplicación. Si es posible satisfacer su pedido, el manager reserva los nodos y envía a ejecutar la aplicación del usuario. Por el contrario, encola el pedido y lo atenderá cuando los recursos estén disponibles.

2.2. Multicores y Clusters de multicore. - Hasta principios del 2000, la evolución de los procesadores se basaba en aumentar su frecuencia, y en consecuencia su potencia de cómputo, lo que mejoraba automáticamente el rendimiento de las aplicaciones. Sin embargo, llevar la frecuencia más allá de cierto límite provocaba que el procesador eleve su temperatura, siendo necesario consumir más energía para disipar el calor. Es por eso que, a principios del 2000, dicha tendencia llegó a su fin, dando lugar al surgimiento de los procesadores multicore. Los multicore incluyen varias unidades de procesamiento (cores) independientes, y permiten mejorar el rendimiento de las aplicaciones diseñadas para explotar el paralelismo a nivel de tarea.

Los clusters que conectan máquinas con procesadores multicore (Figura 1b) se benefician por contar con un mayor número de unidades de procesamiento. Para explotar el paralelismo del procesador multicore, la aplicación debe seguir el modelo de programación de memoria compartida. En dicho modelo se crean hilos, uno por core, que se comunican entre sí mediante la memoria compartida. Las herramientas más usadas para desarrollar aplicaciones bajo este modelo son Pthreads (13) y OpenMP (14). También es posible que la aplicación siga el modelo de programación de memoria distribuida, ubicando un proceso por core y comunicándolos mediante mensajes. Las aplicaciones que corren sobre un cluster de multicore pueden utilizar ambos modelos de programación (modelo híbrido) y combinar las distintas herramientas (Por ejemplo: MPI y OpenMP o MPI y Pthreads).

Cómo se mencionó anteriormente, en HPC no es habitual compartir un nodo entre varios usuarios. Los usuarios solicitan al *Resource Manager* nodos del cluster para uso exclusivo. Por lo tanto, si un usuario utiliza un subconjunto de cores del nodo solicitado, los cores ociosos no podrán ser utilizados por otros usuarios.

2.3. Cluster de multicore con GPUs como coprocesadores. - Las placas gráficas (GPUs) surgieron para liberar a la CPU del procesamiento gráfico. Las primeras GPUs estaban compuestas por procesadores no programables dedicados a tareas específicas (implementadas por hardware). En 2006, las GPUs comenzaron a incorporar procesadores

programables que permiten resolver problemas de propósito general (no solamente gráfico), surgiendo el concepto de GPGPU (General-Purpose Computing on GPUs).

Las GPUs mostraron ser más eficientes en rendimiento que los multicores y clusters para ciertas aplicaciones con alto grado de paralelismo (SIMD), de cómputo intensivo, que trabajan sobre datos de tipo regular. Sin embargo, las GPUs no reemplazaron a dichas arquitecturas, sino que se integraron como coprocesadores en los nodos del cluster (Figura 1c).

Las aplicaciones que corren sobre clusters que incorporan GPUs, pueden utilizar únicamente las GPUs del nodo o usarlas de manera colaborativa junto con los CPU cores. Algunos artículos (15; 16) muestran las ventajas de esta integración.

Respecto a la reserva de recursos, cuando un usuario necesita usar GPUs, el Resource Manager del cluster atiende la solicitud reservando nodos que las incorporan. Si el usuario utiliza sólo la/s GPU/s del nodo, los CPU cores no podrán ser utilizados por otro usuario. De igual manera, si el usuario utiliza sólo los CPU cores de un nodo, la/s GPU/s no podrán ser utilizadas por otro usuario.

3. Fundamentos de QC - Estado actual. - El qubit es la unidad de información de la QC. Un qubit puede estar en alguno de los estados básicos (0 ó 1) o en el estado de superposición (0 y 1). Un qubit en estado de superposición lleva asociadas dos probabilidades, que representan la probabilidad de valer 0 y 1 respectivamente. Asimismo, debe tenerse en cuenta que al observar o "medir" un qubit en estado de superposición este colapsa hacia el estado básico más probable (ya sea 0 ó 1). Por esta razón, para obtener los resultados de la ejecución de un algoritmo sobre una máquina cuántica deben realizarse múltiples ejecuciones (o shots). (4; 5; 6)

El principio de superposición dota a las computadoras cuánticas de un paralelismo inherente. En un sistema cuántico de N-qubits pueden encontrarse simultáneamente superpuestos 2^N estados, por tanto, el efecto de aplicar una operación a los N-qubits equivale a aplicarla a todos los estados en paralelo.

En la década de 1990 aparecieron las primeras computadoras cuánticas limitadas a 2 o 3 qubits. Actualmente, existen máquinas cuánticas individuales de decenas de qubits (17; 18; 19; 20). Estas computadoras operan sobre los qubits utilizando puertas cuánticas, que son equivalentes a las puertas lógicas utilizadas por las computadoras clásicas y permiten manipular el estado de los qubits para realizar cálculos (21).

En la misma década fueron propuestos los primeros algoritmos cuánticos (QAs ó Quantum Algorithms) (22; 23; 24). Un QA aplica operaciones sobre los qubits, con el objetivo de incrementar la probabilidad de los estados deseados y decrementar la de aquellos no deseados, y finaliza con la medición, la cual colapsa el estado del sistema a un estado básico con cierta probabilidad, obteniendo así la salida clásica (bits).

Las máquinas cuánticas deben estar en un ambiente controlado para evitar o mitigar la decoherencia, que surge por las interacciones de un qubit con su entorno y provocan perturbaciones que conducen a errores en la información cuántica. Por esta razón, requieren de temperaturas extremadamente bajas, cercanas al cero absoluto (-273 C), y estar blindadas ante señales externas (radio, luz, electromagnéticas) lo cual dificulta su comercialización. Es por ello que las máquinas cuánticas actuales son provistas por unas pocas instituciones y accedidas a través de servicios en la Nube (17; 18).

4. Integración HPC-QC: Trabajos relacionados. - Algunos autores plantean el reemplazo de los sistemas HPC por su equivalente cuántico (25; 26) al proponer clusters cuánticos sin ningún tipo de integración con arquitecturas HPC clásicas. Sin embargo, varios artículos sostienen que la QC no reemplazará al HPC tradicional, sino que serán herramientas complementarias (27; 28; 29; 30; 31; 32).

En (27) los autores proponen arquitecturas híbridas clásica-cuántica, que incluso pueden integrar múltiples Quantum Processing Units (QPUs) interconectadas.

Foster (28) plantea que la QC no será adecuada para todas las cargas de trabajo, sino que complementará a las arquitecturas HPC clásicas cuando sea necesario. Britt (27) plantea una idea similar y la posibilidad de usar máquinas cuánticas como coprocesadores, de la misma forma que sucede actualmente con las GPUs. Perelshtein (29) muestra la ventaja práctica de este enfoque al ejecutar algoritmos específicos sobre una arquitectura híbrida, que combina infraestructura clásica de alto rendimiento (CPUs, GPUs) y QPUs simuladas. Asimismo, Britt (27) pone de manifiesto que deben considerarse nuevas métricas, basadas en las puertas cuánticas y el número de shots, para evaluar el rendimiento de los algoritmos cuánticos.

En cuanto a la integración HPC-QC, los artículos (27; 30; 31; 32) realizan propuestas que abarcan: la integración de estas arquitecturas, el desarrollo de middlewares que faciliten el uso y administración de recursos clásicos-cuánticos, la integración de las herramientas de programación cuántica y aquellas usadas en HPC y el fácil acceso a la infraestructura como un todo.

Britt (30) expone algunos desafíos para alcanzar dicha integración: la incompatibilidad entre las tecnologías de computación cuántica y clásica, y el esfuerzo para gestionar las interacciones lógicas entre ambos tipos de sistemas y errores en la información cuántica.

5. Discusión. - Actualmente, las computadoras cuánticas necesitan de un ambiente controlado y no se encuentran integradas en sistemas HPC clásicos. Por esta razón, sólo pueden ser accedidas individualmente a través de servicios en la Nube.

Varios artículos proponen incorporar QPUs en sistemas HPC, cumpliendo el rol de coprocesador, al igual que sucede con las GPUs actualmente. Eventualmente, las QPUs podrían ser accedidas a través del nodo que las contiene. Por lo tanto, el nodo tendrá múltiples opciones de procesamiento (Figura Id): cores, GPU/s y QPU/s. El primer desafío es resolver la incompatibilidad tecnológica entre los sistemas clásicos HPC y las QPUs para lograr una correcta integración. Los sistemas clásicos están basados en silicio y este no es el caso de las computadoras cuánticas actuales. Aun cuando se consiga una integración adecuada, es necesario que la interacción entre ambos sistemas se encuentre libre de interferencias tal como se plantea en (31).

Otros de los desafíos plantean cómo se procesarán y distribuirán las cargas de trabajo en los sistemas integrados HPC-QC. En (28) los autores plantean que la QC, al igual que las GPUs en sistemas HPC actuales, podría no ser adecuada para todas las cargas de trabajo. En un sistema actual HPC clásico, compuesto por cores y GPUs, es común plantear tres tipos de estrategias de procesamiento: tipo pipeline, colaborativo o independiente. Estas mismas estrategias podrían emplearse al integrar las QPUs. Las aplicaciones que corran en este sistema híbrido HPC-QC deberán seleccionar las unidades de procesamiento (CPU cores, GPUs, QPUs) que mejor se adapten a la tarea específica a resolver.

Las aplicaciones tipo (pipeline), que atraviesan etapas donde la salida de una es la entrada de la siguiente, podrán resolver cada etapa sobre la unidad de procesamiento más adecuada. En algunos casos, podrá alcanzarse el máximo paralelismo, haciendo que todas las unidades de procesamiento trabajen simultáneamente, cada una cumpliendo una etapa.

Las aplicaciones cuyo cómputo sea factible de realizarse en todas las unidades de procesamiento podrán usar dichas unidades de forma colaborativa y en paralelo. En este caso, cada unidad de procesamiento hará el mismo trabajo sobre una parte de los datos de entrada. El desafío está en distribuir adecuadamente los datos entre las unidades en función de su potencia de cómputo, de manera que todas terminen la ejecución al mismo tiempo. En este caso, será necesario hacer un profiling, es decir ejecutar la aplicación sobre cada unidad de procesamiento para determinar su potencia de cómputo.

Otro tipo de aplicaciones podría beneficiarse al usar únicamente un tipo de unidad de manera independiente: sólo las QPUs, sólo los CPU cores, o sólo las GPUs. Un desafío no menos importante, y que dependerá de cómo se resuelva la tecnología de integración subyacente, está relacionado a cómo compartir datos entre el sistema clásico HPC y las QPUs. En un sistema HPC actual, los datos residen en la RAM de la CPU y se comparten con una GPU realizando copias

hacia y desde la memoria de la GPU a través del puerto PCI express. Algunas herramientas, como el manejo de memoria unificada en CUDA, hacen transparente al programador estas transferencias. De manera similar, será necesario contar con la tecnología adecuada que permita representar los datos en memoria de la CPU en los registros de las QPUs de manera transparente al programador.

Por lo tanto, los sistemas HPC-QC integrados plantean al menos dos desafíos importantes: la tecnología de integración y el correcto diseño de aplicaciones híbridas, que dependerá no sólo de la naturaleza de la aplicación, sino también del volumen de los datos de entrada. Esto es, si el volumen de los datos de entrada es pequeño, el overhead introducido para coordinar las unidades de procesamiento podría tener mayor impacto en el rendimiento final, pudiendo no resultar beneficioso utilizar una arquitectura integrada.

6. Conclusiones y trabajo futuro. - Varios autores coinciden que la QC no reemplazará a los sistemas HPC clásicos, sino que se complementarán. En particular, algunos plantean integrar las QPUs a los sistemas HPC existentes y usarlas como coprocesadores. Esto presenta dos desafíos. En primer lugar, se debe determinar cómo llevar a cabo la integración tecnológica HPC-QC. En segundo lugar, se debe estudiar para qué tipo de aplicaciones serán más adecuadas las QPUs respecto a otro tipo de arquitectura.

Como trabajo futuro, se pretende seleccionar problemas típicos de HPC que puedan ser resueltos en QC, comparar y analizar el rendimiento en ambos sistemas, y plantear soluciones que utilicen sistemas HPC-QC integrados con el fin de alcanzar mejoras significativas. Cabe destacar que la tecnología actual presenta ciertas limitaciones: las máquinas cuánticas cuentan con pocos qubits y no se encuentran integradas a sistemas HPC. Por esta razón, será de importancia contar con ambientes simulados adecuados para la experimentación.

Referencias

- [1] A. Grama, A. Gupta, G. Karypis, V. Kumar, *An Introduction to Parallel Computing*. 2da edn. Addison Wesley, 2003.
- [2] I. Foster, *Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering*. Pearson, 2023.
- [3] J. Dongarra, I. Foster, G. Fox, W. Gropp, K. Kennedy, L. Torczon, A. White, *Sourcebook of Parallel Computing*. Morgan Kaufmann, 2003.
- [4] S. Golestan, MR. Habibi, SY. Mousazadeh Mousavi, JM. Guerrero, JC. Vasquez, Quantum computation in power systems: An overview of recent advances. *Energy Reports*, Volume 9, Pages 584-596, 2023.
- [5] C. Hughes, J. Isaacson, A. Perry, RF. Sun, J. Turner, *What Is a Qubit? In: Quantum Computing for the Quantum Curious*. Springer, Cham, 2021.
- [6] SS. Gill, A. Kumar, H. Singh, et al. Quantum computing: A taxonomy, systematic review and future directions. *Softw: Pract Exper*, 52(1), 66-114 2022.
- [7] G. Hager, G. Wellein, *Introduction to HPC for Scientists and Engineers 2011*
- [8] Parallel Virtual Machine (PVM) https://www.epm.ornl.gov/pvm/pvm_home.html
- [9] Message Passing Interface (MPI) <https://www.mpi-forum.org/>
- [10] OpenPBS Resource Manager <https://www.openpbs.org/>
- [11] Torque Resource Manager <https://adaptivecomputing.com/cherry-services/torque-resourcemanager/>
- [12] Slurm Resource Manager <https://slurm.schedmd.com/>
- [13] Posix Threads (Pthreads) <https://posix.opengroup.org/>
- [14] Open Multi-Processing (OpenMP) <https://www.openmp.org/>
- [15] V. Sanz, A. Pousa, M. Naiouf, A. De Giusti, Accelerating pattern matching with CPU-GPU collaborative computing. *Algorithms and Architectures for Parallel Processing: 18th International Conference, ICA3PP 2018, Guangzhou, China, November 15-17, 2018, Proceedings, Part I* 18 310-322, Springer International Publishing, 2018.
- [16] V. Sanz, A. Pousa, M. Naiouf, A. De Giusti, Accelerating pattern matching with CPU-GPU collaborative computing. *Algorithms and Architectures for Parallel Processing: 19th International Conference, ICA3PP 2019, Melbourne, VIC, Australia, December 9 11, 2019, Proceedings, Part I* 19 391-403, Springer International Publishing, 2020.
- [17] Amazon Braket: Quantum Computers, QuEra. Accessed: Feb. 29, 2024.[Online]. Available: <https://aws.amazon.com/es/braket/quantum-computers/quera/>
- [18] Amazon Braket: Quantum Computers, Rigetti. Accessed: Feb. 29, 2024.[Online]. Available: <https://aws.amazon.com/es/braket/quantum-computers/rigetti/>
- [19] IBM: Quantum Computing Technology. Accessed: Feb. 29, 2024.[Online]. Available:

<https://www.ibm.com/quantum/technology>

[20] Google, Quantum AI.: Quantum Computer Datasheet. Accessed: Feb. 29, 2024.[Online]. Available: <https://quantumai.google/hardware/datasheet/weber.pdf>

[21] A. Barenco, CH. Bennett, R. Cleve, DP. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin, H. Weinfurter, Elementary gates for quantum computation. arXiv:quant-ph/9503016v1, 1995.

[22] PW. Shor, Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. arXiv:quant-ph/9508027v2, 1995.

[23] D. Deutsch, R. Jozsa, Rapid solution of problems by quantum computation. Proceedings of the Royal Society of London A439:553, 1992.

[24] LK. Grover, A fast quantum mechanical algorithm for database search. Proceedings, 28th Annual ACM Symposium on the Theory of Computing, 1996.

[25] SJ. Devitt, WJ. Munro, K. Nemoto, High Performance Quantum Computing. arXiv preprint arXiv:0810.2444, 2008.

[26] T. Haner, DS. Steiger, T. Hoefer, M. Troyer, Distributed Quantum Computing with QMPI. Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. SC 2021. Article No., 2021.

[27] KA. Britt, TS. Humble, High-Performance Computing with Quantum Processing Units. ACM. Journal on Emerging Technologies in Computing Systems, Volume 13 Issue 3 Article No.: 39, pp 1-13, 2015.

[28] S. Banerjee, I. Foster, W. Gropp, Infrastructure for Artificial Intelligence, Quantum and High-Performance Computing. 2020.

[29] M. Perelshtein, A. Sagingalieva, K. Pinto, V. Shete, A. Pakhomchik, A. Melnikov, F. Neukart, G. Gesek, A. Melnikov, V. Vinokur, Practical application-specific advantage through hybrid quantum computing. arXiv preprint arXiv:2205.04858 [quant-ph], 2022.

[30] KA. Britt, FA. Mohiyaddin, TS. Humble, Quantum Accelerators for High-Performance Computing Systems. arXiv preprint arXiv: 1712.01423, 2017.

[31] N. Saurabh, S. Jha, A. Luckow, A Conceptual Architecture for a Quantum-HPC Middleware. arXiv preprint arXiv:2308.06608v1, 2023.

[32] A. Elsharkawy, M. Xiao-Ting, P. Seitz, Y. Chen, Y. Stade, M. Geiger, Q. Huang, X. Guo, MA. Ansari, CB. Mendl, D. Kranzlmüller, M. Schulz, Integration of Quantum Accelerators with High Performance Computing – A Review of Quantum Programming Tools . arXiv preprint arXiv:2309.06167 [cs.ET], 2023.

Nota contribución de los autores:

1. Concepción y diseño del estudio
2. Adquisición de datos
3. Análisis de datos
4. Discusión de los resultados
5. Redacción del manuscrito
6. Aprobación de la versión final del manuscrito

AP ha contribuido en: 1, 2, 3, 4, 5 y 6.

VS ha contribuido en: 1, 2, 3, 4, 5 y 6.

Nota de aceptación: Este artículo fue aprobado por los editores de la revista Dr. Rafael Sotelo y Mag. Ing. Fernando A. Hernández Goberti.