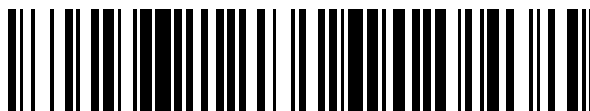


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **3 061 321**

51 Int. Cl.:

G06N 10/80 (2012.01)

G06N 10/20 (2012.01)

G06N 20/00 (2009.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **10.09.2021 PCT/ES2021/070654**

87 Fecha y número de publicación internacional: **17.03.2022 WO22053732**

96 Fecha de presentación y número de la solicitud europea: **10.09.2021 E 21866143 (7)**

97 Fecha y número de publicación de la concesión europea: **17.12.2025 EP 4231204**

54 Título: **Sistema de desarrollo y ejecución de aplicaciones en redes de computación cuántica híbrida**

30 Prioridad:

11.09.2020 ES 202030926

45 Fecha de publicación y mención en BOPI de la traducción de la patente:

01.04.2026

73 Titular/es:

PETERSSEN NODARSE, GUIDO ROGELIO (33,33%)

C/ Iglesia, 6

28754 Mangirón (Madrid), ES;

HEVIA OLIVER, JOSÉ LUIS (33,33%) y

PIATTINI VELTHUIS, MARIO GERARDO (33,33%)

72 Inventor/es:

PETERSSEN NODARSE, GUIDO ROGELIO;

HEVIA OLIVER, JOSÉ LUIS y

PIATTINI VELTHUIS, MARIO GERARDO

74 Agente/Representante:

SERRANO IRURZUN, Francisco Javier

ES 3 061 321 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

DESCRIPCIÓN

SISTEMA DE DESARROLLO Y EJECUCIÓN DE APLICACIONES EN REDES DE COMPUTACIÓN CUÁNTICA HÍBRIDA

5

CAMPO DE LA INVENCION

La presente invención se refiere al campo de la computación cuántica y, en particular, al ámbito de los sistemas de desarrollo de aplicaciones y ejecución de las mismas en redes de computación cuántica. Más concretamente, la invención se refiere a arquitecturas de computación híbrida, cuyas entradas pueden comprender algoritmos clásicos y/o cuánticos, y cuyas salidas son programas cuánticos que pueden ejecutarse en ordenadores cuánticos de idéntica o diferente naturaleza, preferentemente de modo agnóstico, es decir, sin limitación a un único lenguaje o entorno nativo de desarrollo de programas.

ANTECEDENTES DE LA INVENCION

La diversificación y las aplicaciones de las tecnologías de la información has crecido exponencialmente en lo que va de siglo. Con ello, la actividad basada en dichas tecnologías genera, a diario, grandes volúmenes de datos, que requieren cada vez mayores capacidades de procesamiento e inmediatez de acceso a los mismos. En este contexto, la ley de Moore está llevando a su límite a los procesadores de los ordenadores clásicos, tal y como los conocemos tradicionalmente. Esta limitación, unida a los desafíos que plantea el mantenimiento de la seguridad e integridad de los datos generados, ha hecho necesario el desarrollo de nuevas tecnologías de procesamiento de información, basadas en tecnologías no clásicas.

En las últimas décadas, las tecnologías cuánticas han evolucionado considerablemente en sus aplicaciones al ámbito de la computación. Si bien los primeros prototipos de ordenadores cuánticos contaban con severas limitaciones, tanto en fiabilidad como en rendimiento, algunos fabricantes (IBM, Google, Microsoft, Rigetti, D-Wave) han comenzado ya a ofrecer soluciones fiables a nivel comercial o, al menos, como prototipos estables más o menos desarrollados, que apuntan a que el sector de la computación cuántica se encuentra próximo a alcanzar una madurez suficiente para penetrar en el mercado a gran escala en los próximos diez o quince años. Si bien dichas soluciones presentan un estado prematuro de evolución como para explotarlas a nivel de negocio, de forma acelerada se están marcando las pautas y relaciones para que, a corto plazo, se creen las condiciones necesarias para que esta nueva tecnología sea una realidad, cambiando las formas no sólo de trabajar, sino de entender globalmente el sector de las tecnologías de la información.

35

La naturaleza probabilística de la mecánica cuántica (base de las tecnologías de computación cuántica), aporta una serie de ventajas en su aplicación al desarrollo de algoritmos, que permite ejecutar los mismos de forma más eficiente que mediante la computación clásica determinista. Ello hace posible resolver problemas cuya complejidad hace inviable su solución mediante ordenadores clásicos en tiempos de procesamiento razonables, incluso utilizando superordenadores o redes muy grandes de computación basadas en procesamiento paralelo a gran escala. Asimismo, las técnicas de algoritmia cuántica se han venido desarrollando de forma continuada desde la década de 1980 y, actualmente, se consideran en un estado suficientemente maduro para su implementación en sistemas físicos.

40

En las soluciones existentes de ordenadores cuánticos comerciales, cada fabricante proporciona, por lo general, un lenguaje, entorno o kit de desarrollo diferente para el desarrollo de algoritmos y aplicaciones de computación (por ejemplo, el kit de desarrollo QISKIT de IBM o el QDK de Microsoft). Ello se debe, además de a motivos comerciales, a que el hardware en que se basa cada máquina presenta unas enormes diferencias frente a las de los demás fabricantes (por razones tecnológicas y de diseño, número de bits cuánticos, etc.), lo que hace necesario que cada ordenador cuántico comercial sólo pueda ejecutar algoritmos nativos (es decir, programados en el lenguaje proporcionado por el fabricante de cada máquina). Esta situación impone también diferentes limitaciones sobre la reproducibilidad de los algoritmos en otros lenguajes, así como sobre su rendimiento (ya que, actualmente, cada ordenador cuántico se comporta de una forma diferente en función de los algoritmos ejecutados). Como consecuencia, la realidad hoy en día es que unos ordenadores son mejores para la implementación de determinadas rutinas que otros.

Como limitación adicional, es necesario asumir que la computación clásica (y su algoritmia correspondiente) convivirá, con una alta probabilidad, durante mucho tiempo con la computación cuántica. Asimismo, la sustitución de los sistemas clásicos por los cuánticos será previsiblemente lenta, lo que crea la necesidad de generar arquitecturas de computación híbridas, a través de las cuales sea posible adaptar o traducir los algoritmos clásicos conocidos a los algoritmos cuánticos aptos para su ejecución óptima en las máquinas de cada fabricante o lenguaje.

Aunque son inferiores a las plataformas nativas, las soluciones híbridas clásico-cuánticas superan ya ampliamente a las que pueden ejecutar en tiempos razonables los más potentes procesadores “clásicos”. Por ello, su uso puede solucionar, a corto y medio plazo, la transición del ámbito nativo clásico al ámbito nativo cuántico. De este modo, las arquitecturas híbridas pueden proporcionar herramientas eficaces para evaluar, probar y explotar las tecnologías cuánticas, pero sin tener que esperar al momento en que la computación cuántica nativa provea los servicios equivalentes. A modo de ejemplo, y según los estudios actuales, gracias a la arquitectura híbrida ya se podrían construir soluciones en el campo de la salud, de la química, de la logística, las finanzas, y, en general, de cualquier modelo de big data, que podrían verse muy beneficiadas de la nueva capacidad de cálculo cuántica para la solución de problemas actuales.

Un sistema conocido para implementar entornos de desarrollo de computación cuántica integrados se describe en la solicitud de patente_US 2003/0169041 A1. No obstante, esta solución requiere necesariamente el uso de un hardware controlador para cada tipo de arquitectura de computación cuántica conectada a dicho sistema.

Otra propuesta conocida para la implementación de entornos de desarrollo y ejecución de algoritmos cuánticos se describe en el artículo de investigación de Robert Wille et al., con título “*IBM's Qiskit Tool Chain: Working with and Developing for Real Quantum Computers*” y publicado en 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE), 20190325 EDAA. En este documento se divulga una herramienta de transpilación tradicional, basada en el uso de reglas de optimización de recursos del ordenador cuántico, por ejemplo mediante la simplificación de las rutinas, reducción de operaciones, etc. No obstante, esta herramienta no permite utilizar ordenadores cuánticos con diferentes arquitecturas.

Otra propuesta para integrar diferentes sistemas de computación cuántica bajo una única aplicación de desarrollo se divulga en el artículo «A Language and Hardware Independent Approach to Quantum-Classical Computing», Arxiv.org, Cornell University Library 201, Ithaca, NY 14853 (4 de octubre de 2017). Este documento describe un

5 modelo de programación y un marco de software que permite la aceleración cuántica dentro de flujos de trabajo de software de computación estándar o de alto rendimiento. Dicho marco sigue un modelo de máquina de coprocesador independiente del hardware de computación cuántica subyacente, lo que permite que los programas cuánticos se definan y ejecuten sobre una variedad de tipos de unidades de proceso cuántico, a través de una interfaz de programación de aplicaciones unificada. Sin embargo, esta propuesta no aporta una solución para la adaptación de algoritmos clásicos a sus equivalentes cuánticos, de forma agnóstica y multiplataforma, de manera que puedan ejecutarse en una pluralidad de máquinas cuánticas en una única plataforma. Asimismo, dado que esta plataforma propone un marco en el que deben desarrollarse adaptaciones específicas para cada ordenador cuántico distinto, no establece la forma en que esto debe hacerse, recayendo en última instancia en el usuario la responsabilidad de adaptar la plataforma a la solución de un problema concreto, o a una máquina cuántica específica.

15 Otras referencias conocidas, como «Quantum Volume: A Yardstick To Measure The Performance Of Quantum Computers – Moor Insights & Strategy» (2020), y Cross Andrew W. et al., «Validating quantum computers using randomized model circuits», Phys. Rev. A, vol. 100, n.º 3 (2019), describen diversos sistemas de seguimiento para medir el rendimiento de los ordenadores cuánticos.

20 La presente invención está orientada a superar las anteriores limitaciones del estado de la técnica, mediante un sistema de desarrollo de aplicaciones en redes de computación cuántica híbrida, que facilita la adaptación de algoritmos clásicos a sus equivalentes cuánticos, de forma agnóstica y multiplataforma, de forma que puedan ser ejecutados en máquinas de diferentes fabricantes según se requiera, ya sea por decisión del usuario o por detección automática basada en diferentes modelos de telemetría.

DESCRIPCIÓN BREVE DE LA INVENCION

25 La presente invención proporciona una solución a los problemas técnicos anteriormente descritos, mediante un sistema de computación cuántica híbrida que proporciona un marco de herramientas, servicios y procesos que tienen como objetivo consolidar el modelo de construcción de aplicaciones cuánticas, de forma que los equipos multidisciplinares de cualquier ámbito del conocimiento no tengan que preocuparse de las tecnologías específicas de cada fabricante, sino únicamente del planteamiento general de los algoritmos y de su ejecución. Con ello, es posible evitar las elevadas complejidades técnicas asociadas a cada sistema nativo.

35 Más concretamente, el objeto de la invención se refiere a un sistema de desarrollo y ejecución de aplicaciones en redes de computación cuántica híbrida. Dicho sistema comprende, preferentemente, uno o más clientes conectados a, al menos, un servidor, donde dicho servidor está conectado, a su vez, a uno o más ordenadores cuánticos, donde cada uno de dichos ordenadores cuánticos posee una interfaz de programación de aplicaciones propia, así como potenciales extensiones de las interfaces de programación de aplicaciones (APIs) correspondientes a cada fabricante, aportando capas de optimización y control de errores adicionales. Asimismo, el servidor puede constituir tanto un elemento único como, en diferentes realizaciones preferentes, una pluralidad de servidores interconectados, por ejemplo en agrupaciones de tipo granja con estructura cooperativa y/o competitiva.

40 Se proporciona un sistema según la reivindicación independiente 1.

En una realización preferente de la invención, el sistema comprende adicionalmente un subsistema de gestión de algoritmia, configurado con una o más interfaces de edición de meta-algoritmos cuánticos de forma manual.

5 En otra realización preferente de la invención, el sistema comprende adicionalmente un subsistema de telemetría, configurado con una o más funcionalidades de traza, depuración y medida del rendimiento de la ejecución de algoritmos de computación cuántica en los ordenadores cuánticos, y de su relación con los meta-algoritmos de computación cuántica. Más preferentemente, dicho subsistema de telemetría comprende uno o más módulos de aprendizaje automático o de inteligencia artificial, para la medida del rendimiento de la ejecución de algoritmos de computación cuántica en los ordenadores cuánticos, de la decoherencia de los mismos, y/o de su relación con los
10 meta-algoritmos de computación cuántica.

En otra realización preferente de la invención, el sistema comprende adicionalmente un subsistema de gobierno alimentado por las salidas del subsistema de telemetría, adaptado para proporcionar información de evolución, carga y depuración de la ejecución de algoritmos de computación cuántica en los ordenadores cuánticos.

15 En otra realización preferente de la invención, los ordenadores cuánticos y el servidor forman una red de computación distribuida o en la nube.

En otra realización preferente de la invención, uno o más de los ordenadores cuánticos comprenden una máquina virtual. Dicha máquina virtual adopta, más preferentemente, la forma de uno o más simuladores de máquinas reales, que pueden estar explotadas de forma aislada o interconectadas en agrupaciones (“clusters”) o granjas de servidores.

20 En otra realización preferente de la invención, el BPM está configurado con medios de recepción de mensajes de llamada a funciones remotas que parametrizan y encapsulan instrucciones enviadas por los clientes.

En otra realización preferente de la invención, el BPM comprende un primer servicio web de recogida de los mensajes de llamada y de almacenamiento de los mismos en una base de datos.

30 En otra realización preferente de la invención, el compilador, el transpilador y/o el motor de ejecución de unidades de procesamiento están implementados como uno o más componentes de tipo plugin.

En otra realización preferente de la invención, el BPM comprende un módulo de carga de los plugins.

35 En otra realización preferente de la invención, el BPM comprende un segundo servicio web de procesamiento asíncrono de respuestas.

En otra realización preferente de la invención, el BPM está configurado con medios de envío de mensajes de respuesta asíncrona al cliente.

40 En otra realización preferente de la invención, el BPM comprende un procesador de grupos (o “pools”) de comandos, adaptado para el procesamiento de los comandos pendientes de ser ejecutados en la base de datos. Más preferentemente, dicho procesador de grupos de comandos es programable en el tiempo, de forma recurrente y con una ejecución autónoma, en forma de demonios.

En otra realización preferente de la invención, uno o más de los ordenadores cuánticos comprenden un modelo de computación basado en puertas cuánticas. Asimismo, alternativa o complementariamente, uno o más de los ordenadores cuánticos pueden comprender un modelo de computación basado en temple cuántico (en inglés, "quantum annealing"). Ello asegura la compatibilidad del sistema con la mayoría de los fabricantes (IBM, Microsoft, Rigetti, D-Wave, Google, IonQ, Fujitsu, Amazon Braket, etc.), así como de simuladores de computación cuántica de terceros (QuTECH, CTIC, etc.).

DESCRIPCIÓN DE LAS FIGURAS

Para completar la descripción y facilitar la mejor comprensión de la invención, se añade a la descripción una serie de figuras que forman parte de la descripción e ilustran un modo de realización preferente del sistema descrito. Dichas figuras, por tanto, no deben ser interpretadas como limitante del alcance de la invención, sino como un mero ejemplo de cómo la invención se puede llevar a cabo.

La Figura 1 muestra un esquema de alto nivel del sistema de la invención, según una realización preferente de la misma.

La Figura 2 muestra un esquema detallado del sistema de la invención, según una realización preferente de la misma.

Referencias numéricas de las figuras:

Con objeto de ayudar a una mejor comprensión de las características técnicas de la invención, las citadas figuras se acompañan de una serie de referencias numéricas donde, con carácter ilustrativo y no limitativo, se representa lo siguiente:

1	Motor de gestión de procesos de negocio (BPM)
2	Compilador
3	Diccionario
4	Transpilador
5	Motor de ejecución de unidades de procesamiento
6	Subsistema de gestión de algoritmia
7	Subsistema de telemetría
8	Subsistema de gobierno
9	Mensajes de llamada a funciones remotas
10	Servicio web de recogida de mensajes
11	Base de datos
12	Servicio web de procesamiento asíncrono de respuestas
13	Mensajes de respuesta asíncrona
14	Procesador de grupos de comandos
15	Módulo de carga de plugins
100	Cliente

200	Servidor
300, 300', 300"	Ordenadores cuánticos

DESCRIPCIÓN DETALLADA DE LA INVENCION

5 Se procede a continuación a describir un ejemplo de realización preferida de la presente invención, aportada con fines ilustrativos pero no limitativos de la misma, basada en las Figuras 1-2 del presente documento.

10 La Figura 1 muestra un esquema a alto nivel de la invención, donde se representa una realización preferente del sistema de desarrollo y ejecución de aplicaciones en redes de computación cuántica híbrida reivindicado. Como se aprecia, dicho sistema comprende un conjunto de herramientas, servicios y procesos (denominados, de forma global, como "framework") que hacen posible ejecutar unidades de proceso cuánticas, de forma independiente del entorno o tecnología de computación bajo la que se ejecuten, y abstrayendo al aplicativo clásico de la complejidad que caracteriza su adaptación al ámbito cuántico para el usuario. Así pues, el objetivo fundamental del sistema es el de proporcionar a los aplicativos clásicos el equivalente a unidades de proceso tradicionales (objetos con métodos y funciones) que encapsulan, a alto nivel:

- 15 - Taxonomías de librerías con soluciones "Out of the Box".
- Interfaces para hacer posible la extensibilidad de las librerías predeterminadas del sistema.
- Una arquitectura cuántica mediante abstracción del tipo de tecnología, complejidad y características.
- Una arquitectura escalable (preferentemente, en redes de servidores distribuidos), capaz de ejecutar rutinas en múltiples unidades de computación, mediante técnicas y procesos de trabajo en paralelo.
- 20 - Una arquitectura confiable, capaz de aportar mecanismos de control de errores que permitan minimizar la decoherencia propia del hardware cuántico.
- Un lenguaje abstracto que permite diseñar circuitos cuánticos de alto nivel, independientes de la tecnología cuántica subyacente y del tipo de aproximación necesario para resolver un problema. Ello implica no sólo proporcionar reglas de uso de lógica de puertas y sus efectos, sino también de la captura de eventos y datos necesarios para la resolución de problemas más complejos.
- 25 - Un núcleo ("core") de ejecución modular y ampliable de forma totalmente parametrizada.
- Un transpilador capaz de generar código específico de acuerdo a las reglas del circuito, así como de optimizar el circuito para generar el mejor código posible, que generará las unidades compatibles con el núcleo ("core") de ejecución del sistema. Dicho transpilador es capaz, además, de aportar la capa de protección de decoherencia que sea posible al resultado total.
- 30 - Un diccionario de registro de componentes, algoritmos, requisitos, etc. que ofrezca una base reutilizable a nuevos problemas sobre soluciones ya desarrolladas y probadas, permitiendo su uso tanto público como privado, de acuerdo con términos definidos en niveles de contratos.
- Un primer motor de persistencia que almacenará telemetría completa de los procesos ejecutados, de manera que se pueda recoger toda la información posible de los experimentos realizados, proporcionando diferentes entradas a subsistemas de aprendizaje automático e inteligencia artificial (IA).
- 35 - Un segundo motor de persistencia que almacenará la configuración, gestión, parametrización, seguridad y monetización, así como los aplicativos de interfaz de usuario que permitirán gobernar el sistema. Dicho motor resulta necesario en todo sistema de información, y debe ofrecer un conjunto de herramientas tanto de utilidad como de interfaz de usuario que permitan el acceso a las funcionalidades del sistema de forma controlada, asistida y accesible a profesionales que no tienen por qué estar versados en los principios
- 40 más básicos del sistema cuántico subyacente.

- Aplicativos de interfaz de integración, que permitan la explotación de las librerías a cualquier sistema de terceros, empleando protocolos preferentemente abiertos. De ese modo, el sistema puede acoplarse a sistemas clásicos bajo demanda, proporcionando el contexto híbrido necesario para la coexistencia de la computación clásica y cuántica.

5

De acuerdo con lo anterior, el sistema de la invención representado en la Figura 1 comprende, más concretamente, los siguientes subsistemas:

- Motor de gestión (1) de procesos de negocio (en inglés, “*business process manager*”, o BPM):

10

El BPM (1) es el subsistema responsable de proporcionar el ecosistema de comandos y procesos asíncronos, basados en teoría de colas, por el cual el sistema garantiza procesos de ejecución totalmente desacoplados de los clientes. Para ello, el BPM (1) proporciona un motor escalable que programa y lanza los servicios de llamada, ejecución, telemetría, protocolo y aplicativo de gobierno. Sobre este motor se construirán los aplicativos (o “plugins”) que serán ejecutados de forma programada, de forma que el BPM (1) actúa como supervisor general del comportamiento de todos los módulos y unidades funcionales del sistema. Cada aplicativo o plugin se comporta, por tanto, como una unidad de ejecución aislada de las demás en el sistema, caracterizada preferentemente por un protocolo propio de mensajes de entrada, un protocolo de mensajes de respuesta y un conjunto de módulos binarios de ejecución. Es relevante aclarar, en este punto, que las tecnologías BPM son conocidas por sí solas con carácter general. Por ello, la presente invención no reivindica una tecnología BPM particular o ventajosa, sino su aplicación general en el ámbito del sistema descrito, así como su combinación novedosa con el resto de los elementos que forman dicho sistema.

15

20

- Compilador (2):

25

El compilador (2) es el subsistema responsable de generar los productos de metalenguaje del sistema (o meta-algoritmos), a partir de las gramáticas generadas por el usuario, procedentes de algoritmos de computación clásica o cuántica, basados en lenguajes, gramáticas o reglas semánticas determinadas, definición de puertas cuánticas, control de flujo, y su ejecución por parte de un motor de compilación. Ello permite que el usuario traduzca dichos algoritmos a un lenguaje apto para su procesamiento por parte del sistema, para su ejecución posterior bajo el lenguaje específico de que cada máquina de computación cuántica y requisitos de su fabricante.

30

Para realizar estas funciones, el compilador (2) está conectado a un diccionario (3), implementado en forma de una o más bases de datos, librerías de software o tablas de correspondencia, que permiten traducir el algoritmo planteado por el usuario en el metalenguaje utilizado por el sistema.

35

En este contexto, el uso preferente del compilador (2) será la traducción de algoritmos clásicos al citado metalenguaje en forma de meta-algoritmos, con el objetivo de su ejecución bajo una lógica cuántica que resulte, por tanto, ventajosa. No obstante, en otra realización preferente de la invención, el compilador (2) y el diccionario (3) pueden estar configurados para traducir algoritmos cuánticos desarrollados en un determinado lenguaje, al metalenguaje cuántico del sistema en forma de meta-algoritmos.

40

Con el objetivo de agilizar y simplificar las labores del usuario, la traducción realizada por el compilador (2) para el diseño de circuitos cuánticos puede estar, parcial o totalmente, implementada mediante una o más interfaces

gráficas de diseño, que permitirán representar de forma visual dichos circuitos, así como sus posibles variaciones, errores u otras métricas, previamente a su compilación como meta-algoritmos cuánticos en el metalenguaje del sistema. Así, por ejemplo, dichas interfaces pueden estar basados en herramientas o drivers de diseño conocidos, tales como Quirk, Flowchart, etc.

5

El compilador (2) puede disponer de tantas clases de compilación como diferentes gramáticas de entrada puedan agregarse al sistema. O, incluso, permitir su extensibilidad por parte de terceros, por ejemplo como fruto de colaboraciones o extensiones customizadas, vía interfaces de programación de aplicaciones (APIs). Esta característica presenta una evidente ventaja competitiva y brinda excelentes posibilidades de escalabilidad e integración con herramientas y soluciones especializadas en servicios muy puntuales de terceros. Es relevante aclarar, también con relación a este punto, que las tecnologías de compiladores son conocidas por sí solas con carácter general. Por ello, la presente invención no reivindica una tecnología de compilación particular o ventajosa, sino su aplicación general en el ámbito del sistema descrito, así como su combinación novedosa con el resto de los elementos que forman dicho sistema.

15

- Transpilador (4):

El transpilador (4) es el subsistema responsable de generar las unidades de proceso a partir de especificaciones de metalenguaje del sistema, preferentemente en forma de flujos y circuitos cuánticos (meta-algoritmo). Así pues, a partir de la entrada de un diseño de algoritmo cuántico, el transpilador (4) se encarga de generar las unidades de proceso interpretables por un ordenador cuántico nativo determinado, siguiendo un proceso similar al del compilador (2), exportando el metalenguaje del sistema al código aceptado por el ordenador cuántico y, preferentemente, aportando elementos de control que no tienen por qué estar soportados (o, al menos, no de forma generalizada) por cada proveedor cuántico especializado. Por ejemplo, con ello se dota al circuito de puertas no existentes, mediante transformaciones de los recursos equivalentes de los que dispone, “multicontrol gates” como extensión a la algoritmia, iteraciones integradas en el propio esquema del circuito, parametrizaciones de valores de entrada e iteradores, encapsulación de oráculos, definiciones de hamiltonianos annealing equivalentes según especificaciones, etc. Para ello, el transpilador (4) estará también conectado con el diccionario (3) y, más preferentemente, con componentes transformadores. Gracias a dichos elementos transformadores, se integran capacidades de alto nivel soportadas por el metalenguaje en el circuito agnóstico, implementándose con un circuito equivalente, soportado en función de los diferentes proveedores. A modo de ejemplo, es posible implementar como proveedores de transpilación aceptables por el kit de desarrollo Microsoft QDK (transpilador HOST Python, transpilador CIRCUITOS Q#), IBM qExperience QISKIT (transpilador HOST Python, transpilador CIRCUITOS qASM v2.0) o cualquier otro kit análogo de entre los disponibles en el mercado (QuTECH, DWAVE, FUJITSU, AMAZON BRACKET, etc.).

20

25

30

35

Preferentemente, el transpilador (4) es una clase maestra que codifica tanto las reglas de traducción/interpretación como las implementaciones de estas reglas, por ejemplo mediante “drivers” específicos para los diferentes fabricantes. Ello permite, además, ejecutar una parte del algoritmo en una primera máquina cuántica y continuar otra parte de dicho algoritmo en una segunda máquina, obteniendo un resultado combinado de este proceso, que es escalable a una pluralidad de proveedores cuánticos.

40

Por su parte, las unidades de proceso del transpilador (4) comprenden, principalmente, una o más de las siguientes funcionalidades:

- la definición de las entradas en el diccionario (3) con las especificaciones del tipo de hardware cuántico a emplear (servicio web de fabricante, acceso a entorno de ejecución cuántico, inicio de simulador cuántico...etc.);
- 5 - la transformación de los circuitos en metalenguaje agnóstico de la plataforma, para generar todo aquello que el proveedor destino no soporta per-se, salvo realizando transformaciones matemáticas en el algoritmo que generen un equivalente compatible con el proveedor destino. Se respeta el algoritmo, pero físicamente el circuito puede verse alterado para ser compatible con la plataforma destino;
- la generación del componente cuántico responsable de la ejecución en el hardware cuántico y que es específico de la plataforma;
- 10 - la evaluación del nivel de agregación de capas de control de decoherencia, de tal forma que el propio componente por diseño sea capaz de intentar detectar, medir y mitigar en determinado grado el error cuántico inherente del sistema, así como optimizaciones que permitan explotar al máximo la capacidad de la máquina cuántica de destino;
- almacenamiento y taxonomía de unidades compiladas.

15

Como en el caso de otros elementos, es relevante aclarar también que las tecnologías de transpiladores son conocidas por sí solas con carácter general. Por ello, la presente invención no reivindica una tecnología de transpilación particular o ventajosa, sino su aplicación general en el ámbito del sistema descrito, así como su combinación novedosa con el resto de los elementos que forman dicho sistema.

20

- Diccionario (3):

25

Como se ha mencionado previamente, este subsistema es responsable de proporcionar el almacenamiento, catalogación y funcionalidades de gestión de las librerías de software y de sus operaciones correspondientes para los procedimientos de compilación y transpilación del sistema. En diferentes realizaciones preferentes de la invención, el diccionario (3) puede comprender, además, uno o más metamodelos que proporcionan, interfaces a las operaciones registradas, información sobre el nivel de seguridad de acceso a las operaciones por cada algoritmo, información de control de nivel de decoherencia asociado, información de versión y marca de tiempo de la operación, coste de la ejecución asociado, registro de errores y "logs", resultados de ejecución de los algoritmos generados, etc.

30

35

Como en el caso de otros elementos, es relevante aclarar que el uso de diccionarios y, en general, de librerías de software, es conocido por sí solo con carácter general. Por ello, la presente invención no reivindica dicho uso como una característica particular o ventajosa, sino su aplicación general en el ámbito del sistema descrito, así como su combinación novedosa con el resto de los elementos que forman dicho sistema.

40

- Motor (5) de ejecución de unidades de procesamiento:

El motor (5) de ejecución es el subsistema responsable de la ejecución de unidades de procesamiento cuánticas con el sistema. Dichas unidades de proceso corresponden a componentes de código "clásico", responsables de

coordinar y controlar la ejecución de una unidad de proceso cuántico en las máquinas cuánticas adecuadas o, si se da el caso, en un simulador o en un proceso clásico sobre dispositivos hardware muy específicos. Preferentemente, el motor (5) de ejecución controla las generaciones de unidades de ejecución para cada posible fabricante, de acuerdo a las reglas abstractas codificadas en una clase maestra e implementada en diferentes clases “driver” que son responsables de conocer directamente la plataforma del fabricante.

5 Siguiendo las instrucciones del BPM (1), el motor (5) de ejecución corresponde preferentemente a un plugin, cuyo rol será el de procesar los algoritmos cuánticos generados por el transpilador (4), y lanzar dichos algoritmos para su procesamiento en el ordenador cuántico seleccionado. Asimismo, el motor (5) de ejecución permite procesar datos adicionales del sistema, por ejemplo vinculados a métricas concretas de ejecución del algoritmo en el ordenador cuántico seleccionado, o relacionados con procesos de aprendizaje automático, que permiten evaluar parámetros específicos de la ejecución realizada por algoritmo y ordenador cuántico, adquiriendo un conocimiento, por ejemplo, sobre cuáles son las mejores opciones de ejecución en cada caso, mediante inteligencia artificial. Preferentemente, el sistema permite también la entrada de parámetros de forma manual por parte del usuario, que sirven de punto de inicio del algoritmo. Ello se puede llevar a cabo en cada llamada al sistema o, alternativamente, los parámetros se pueden recuperar por medio del diccionario (3).

10 El motor (5) de ejecución de las unidades de procesamiento y su conexión a las diferentes plataformas nativas de computación cuántica pueden estar configuradas como una red de computación siguiendo diferentes topologías de red, siendo posible en diferentes realizaciones utilizar redes distribuidas, redes jerárquicas o topologías de computación en la nube, de forma escalable y/o elástica.

- Subsistema (6) de gestión de algoritmia:

25 De forma opcional en el sistema, este subsistema (6) es el responsable de proporcionar la experiencia del diseño de alto nivel de los circuitos y algoritmos cuánticos, operando preferentemente bajo el metalenguaje independiente de una tecnología cuántica/clásica especializada. De ese modo, además de la capacidad de construir plugins y extensiones de unidades de proceso a bajo nivel, mediante el acceso directo a las tecnologías cuánticas específicas, el subsistema (6) de gestión de algoritmia permite al usuario poder abstraerse de ese nivel, mediante funcionalidades de diseño y edición de los algoritmos generados por el compilador (2), mediante su interfaz de usuario.

30 En este sentido, el metalenguaje y su interpretación como meta-algoritmo por el subsistema (6) de gestión de algoritmia englobará, no sólo las características comunes de los circuitos cuánticos y la capacidad de abstraer las reglas especializadas de determinados ordenadores cuánticos, sino también la capacidad de encapsularlas para su reutilización en diseños de más alto nivel, así como la capacidad de definir los requisitos de entrada/salida (en términos de tipos de datos simples y complejos) y los eventos asociados a la naturaleza de cada elemento definido en el sistema, de forma que sea posible definir herramientas que reaccionen a los mismos.

40 En una realización preferente de la invención, el subsistema (6) de gestión de algoritmia proporciona, además, herramientas adicionales que permiten el diseño asistido de los algoritmos cuánticos. Dichas herramientas pueden comprender, por ejemplo, utilidades de tipo kit de desarrollo de software (SDK), tales como un editor gráfico con soporte completo de las capacidades del metalenguaje, que comprenda un diseñador gráfico configurado para componer tanto el flujo general del algoritmo, como el diseño de los circuitos cuánticos que lo completan, mediante

la representación visual organizada en líneas de bits cuánticos, puertas cuánticas, abstracciones de procedimientos cuánticos, conectores a registros de datos o mapeado de eventos, entre otros.

- Subsistema (7) de telemetría:

5

De forma opcional en el sistema de la invención, el subsistema (7) de telemetría es el responsable de concentrar todos los datos generados en el proceso completo hasta la ejecución de un determinado aplicativo. En diferentes realizaciones de la invención, cada uno de los módulos del sistema puede interactuar con el subsistema (7) de telemetría, de tal forma que sea posible trazar, depurar y medir todas y cada una de las funcionalidades de dicho sistema. Asimismo, las salidas del subsistema (7) de telemetría pueden utilizarse para alimentar un subsistema (8) de gobierno, para proporcionar información completa de evolución, carga y depuración de los procesos ejecutados.

10

- Subsistema (8) de gobierno:

15

Tal y como se ha descrito en el punto anterior, este subsistema (8) es el responsable de proporcionar una interfaz de usuario con el que gestionar internamente los diferentes módulos del sistema. Así pues, mediante las conexiones de los servicios de integración que cada uno de dichos módulos, se proporcionará el acceso a los datos de infraestructura, telemetría y parametrización, para que un administrador del sistema pueda gobernar la topología de los servicios y productos ofrecidos, o para que los usuarios puedan gestionar el ámbito de ejecución de los procesos del sistema.

20

Como en el caso de otros elementos, es relevante aclarar que los usos de subsistemas (6, 7, 8) de gestión de algoritmia, de telemetría o de gobierno son conocidos por sí solos con carácter general. Por ello, la presente invención no reivindica dichos usos como una característica particular o ventajosa de forma aislada, sino su aplicación general en el ámbito del sistema descrito, así como su combinación novedosa con el resto de los elementos que forman dicho sistema.

25

Ejemplo de realización preferente y de uso del sistema de la invención:

30

Finalmente, una vez descritos los elementos esenciales de la invención y, a modo de ejemplo de uso del sistema, la Figura 2 ilustra una realización detallada de los diferentes elementos del mismo y de su uso correspondiente, según se describe a continuación:

35

Para la ejecución de cada proceso, el BPM (1) proporcionará un marco general de ejecución escalable, preferente basado en la generación inicial de mensajes (9) de llamada a funciones remotas ("remote function call", o RFC) que parametrizan y encapsulan las instrucciones del usuario, preferentemente a través de un cliente (100) remoto conectado al sistema. Típicamente, los mensajes de llamada comprenderán documentos XML, que pueden comprender información sobre el tipo de comando de ejecución y de su algoritmo correspondiente, máquina de destino, fecha, usuario, localización, etc.

40

Los mensajes (9) de llamada son recogidos, preferentemente, por un primer servicio web (10) de recogida, configurado para el almacenamiento de los mensajes (9) en una base de datos (11), que es la responsable de persistir los comandos, sus contextos de ejecución, sus historiales de ejecución y sus telemetrías. A modo de ejemplo, dicha base de datos (11) puede comprender las siguientes entidades: almacén de comandos, almacén

de procesamiento, almacén de históricos, almacén de configuración o almacén de seguridad, entre otros.

Adicionalmente, el BPM (1) del sistema contempla el uso de un segundo servicio web (12) de procesamiento asíncrono de respuestas. Este servicio proporciona un API a los componentes de extensibilidad del motor (5) de ejecución, de forma que se proporciona una serialización de los mensajes (9) de llamada con serializaciones correspondientes de mensajes (13) de respuesta al cliente (100) del usuario. De esta forma, los componentes de servidor (200) del sistema pueden lanzar dichos mensajes (13) de respuesta a los clientes (100) de forma asíncrona, si éstos forman parte de su protocolo. Dado que el servicio (12) asíncrono de respuestas depende del tipo de funcionalidad implementada por los plugins, su existencia está, en general, condicionada al diseño de los mismos, y a su explotación por parte de los clientes (100) en el lado del usuario. Al igual que en el servicio web (10) de recogida, existe un API de llamada mediante el servicio web, por el que el cliente (100) puede recoger los mensajes (13) persistidos de respuesta asíncrona.

De forma adicional, el BPM (1) cuenta con un procesador (14), preferentemente paralelo, de grupos de comandos ("command pool"), que es el responsable del procesamiento de los comandos pendientes de ser ejecutados en la base de datos (11). Este procesador (14) es el responsable de planificar las ejecuciones de los plugins en un determinado nodo de un conjunto de servidores (200) empresariales de proceso.

Mediante un algoritmo de gestión de colas ("queues") simulado sobre la base de datos (11), se generan ráfagas de sondeos ("pools") sobre los comandos recibidos, de forma transaccional y preferentemente de forma instanciada en cada nodo. El BPM (1) las introduce en el CORE de procesamiento de mensajes y marca los mensajes en proceso en la base de datos para pasar el motor de persistencia de la base de datos (11). De este modo, comienza para cada comando un hilo propio y aislado de ejecución, donde se analiza el comando contenido en el mensaje (9) de llamada y se procede a la carga dinámica del plugin correspondiente, para poner en marcha la funcionalidad contenida en su interfaz.

Otra funcionalidad muy específica del procesador (14) de grupos de comandos hace posible codificar plugins especiales que se pueden programar en el tiempo, de forma recurrente y con una ejecución totalmente autónoma. De esta forma, es posible implementar "demonios" en el BPM (1) que ejecutan tareas de supervisión, mantenimiento, rendimiento y control.

El BPM (1) es también responsable de controlar un módulo (15) de carga que lanza los plugins de ejecución desarrollados por terceros, para cada tecnología de ordenador cuántico. A partir de dichos plugins, el módulo (15) de carga se comunica, secuencialmente, con el compilador (2), con el transpilador (4) y con el motor (5) de ejecución de unidades de procesamiento, estando mediadas dichas comunicaciones por el uso del diccionario (3), en las operaciones de compilación y de transpilación.

Principalmente y como se ha descrito previamente, el compilador (2) es el plugin responsable de realizar las labores de análisis del lenguaje de diseño de circuitos digitales y del lenguaje de diseño de flujos de orquestación. A través de una API correspondiente de compilación de circuitos y de compilación de flujos de orquestación, este plugin es el responsable de los comandos de análisis de los mensajes (9) de llamada de compilación, recoge la instancia de lenguaje de diseño de circuito cuántico, analiza el lenguaje, compila el lenguaje de diseño a lenguaje intermedio (metalenguaje) de circuito cuántico y persiste el producto en el diccionario (3) de circuitos cuánticos compilados.

Por su parte, el transpilador (4) es el plugin responsable de transpilar los meta-algoritmos de metalenguaje en el lenguaje nativo de cada fabricante. Recogiendo como entrada del mensaje (9) de llamada el comando de transpilación, se extrae el identificador del flujo que se quiere transpilar y los componentes API de transpilación, mediante el diccionario (3), proceden a la generación de código nativo del fabricante. Esto es, se generan los módulos de código asociados a una determinada plataforma cuántica hardware soportada, dejando pendiente en dichos módulos los elementos específicos de ejecución, que serán determinados por el comando de ejecución. Asimismo, el metalenguaje soporta, preferentemente, más características de las que puede aceptar un determinado fabricante, no estando por tanto limitado únicamente a las capacidades del SDK de un proveedor en particular.

Preferentemente, este ciclo de procesos de compilación, transpilación y ejecución persiste en el diccionario (3), de forma que todos los elementos de código se pueden clasificar por en función de los requisitos tecnológicos de un determinado fabricante y de sus requisitos de compilación. En este punto, el sistema inyectará las unidades de código necesarias para que todo flujo de ejecución sea agnóstico y adaptable a cada plataforma. Gracias a esta capacidad, el sistema de la invención es capaz de proporcionar numerosas ventajas al usuario, permitiendo, entre otras:

- Inyectar elementos no soportados directamente entre plataformas, mediante módulos propios que resuelven carencias, o potencian la plataforma específicamente.
- Inyectar elementos de control de errores adicionales o complementarios a las plataformas hardware soportadas.
- Inyectar elementos de optimización adicionales soportados o complementarios a las plataformas hardware soportadas.

A modo de ejemplo, el código inyectado en los procesos de compilación, transpilación y ejecución puede comprender añadir soportes de puertas que no existen en un determinado proveedor, estructuras de control en el propio circuito (como son las puertas de tipo "multi-control") que no son soportadas por múltiples fabricantes, estructuras de repetición no soportadas por los fabricantes en sus circuitos, o estructuras de puertas virtuales que encapsulan subcircuitos en una estructura modular.

A su vez, el motor (5) de ejecución de procesos se plantea como un plugin responsable de generar, mediante una API correspondiente, las unidades físicas de ejecución de una pluralidad de plataformas u ordenadores cuánticos (300, 300', 300''), tal y como los compiladores de cada fabricante requieren. El lanzamiento de procesos por parte del motor (5) de ejecución aporta la carga de ejecución de una aplicación cuántica nativa, la inyección del control de acceso a la seguridad del fabricante y la inyección de abstracción de resultados. De esta forma, se unifica la medición de resultados en un único formato común, independiente del fabricante. El motor (5) de ejecución también es el responsable de captar toda la telemetría de ejecución de cada unidad de aplicación cuántica.

Así pues, el motor (5) de ejecución de procesos estará configurado, gracias al aprendizaje automático y a la comparación entre diferentes plataformas (300, 300', 300'') nativas, para aprovechar al máximo la potencia de la API nativa de cada fabricante, permitiendo:

- Explotar APIs de acceso a recursos democratizados de acuerdo a las reglas del fabricante.
- Aplicar las credenciales de usuario de acuerdo a un determinado nivel de contratación.
- Aplicar abstracciones no soportadas por la plataforma directamente.
- Aplicar rutinas de inicialización de la máquina cuántica, de acuerdo a la tecnología y requisitos del

fabricante del hardware.

5 Como se ha mencionado previamente, en el sistema de la invención, uno o más de los ordenadores cuánticos (300, 300', 300'') comprenden un modelo de computación basado en puertas cuánticas. Asimismo, alternativa o complementariamente, uno o más de los ordenadores cuánticos (300, 300', 300'') pueden comprender un modelo de computación basado en temple cuántico (en inglés, "quantum annealing"). Ello asegura la compatibilidad del sistema con la mayoría de los fabricantes (IBM, Microsoft, Rigetti, D-Wave, Google, IonQ, Fujitsu, Amazon Braket, etc.), así como de simuladores de computación cuántica de terceros (QuTECH, CTIC, etc.).

REIVINDICACIONES

1.- Sistema de desarrollo y ejecución de aplicaciones en redes de computación cuántica híbrida, que comprende uno o más clientes (100) y al menos un servidor (200) al que están conectados los uno o más clientes (100), estando dicho al menos un servidor (200), a su vez, conectado a una pluralidad de ordenadores cuánticos (300, 300', 300''), y donde cada uno de dichos ordenadores cuánticos (300, 300', 300'') posee una interfaz de programación de aplicaciones propia, API;

en donde el servidor (200) comprende los siguientes elementos, implementados mediante software y hardware:

- un BPM (1), adaptado para la programación de unidades de procesamiento de algoritmos de computación cuántica en la pluralidad de ordenadores cuánticos (300, 300', 300''), a partir de instrucciones generadas por los uno o más clientes (100);
- un compilador (2), adaptado para compilar un primer algoritmo de computación clásica o cuántica como un meta-algoritmo de computación cuántica, a través de una serie de reglas de compilación definidas en un diccionario (3);
- un transpilador (4), adaptado para transpilar el meta-algoritmo de computación cuántica como un segundo algoritmo de computación cuántica, a través de una serie de reglas de transpilación definidas en el diccionario (3), y donde dicho segundo algoritmo de computación cuántica es compatible con la API de uno de los ordenadores cuánticos (300, 300', 300''); y
- un motor (5) de ejecución de unidades de procesamiento, adaptado para ejecutar el segundo algoritmo de computación cuántica en, al menos, el ordenador cuántico (300, 300', 300'') cuya interfaz de programación de aplicaciones es compatible;

y **caracterizado por que** el sistema está además adaptado para inyectar unidades de código en los procesos de compilación, transpilación y ejecución, donde dichas unidades de código inyectadas comprenden, al menos:

- unidades de código, inyectadas por el motor (5) de ejecución de unidades de procesamiento y adaptadas para controlar un acceso seguro al ordenador cuántico (300, 300', 300'') compatible con la API; y
- unidades de código de abstracción de resultados, inyectadas por el motor (5) de ejecución de unidades de procesamiento, para unificar la medición de resultados en el ordenador cuántico (300, 300', 300'') compatible con la API, en un formato común único, independientemente del fabricante del ordenador cuántico (300, 300', 300'') compatible con la API.

2.- Sistema según la reivindicación anterior, que comprende adicionalmente un subsistema de gestión algorítmica (6), configurado con una o más interfaces para la edición manual de meta-algoritmos de computación cuántica, y que comprende un diseñador gráfico configurado para componer tanto el flujo general de dichos meta-algoritmos como el diseño de los circuitos cuánticos que los completan, mediante una representación visual organizada en líneas de bits cuánticos, puertas cuánticas, abstracciones de métodos cuánticos, conectores de registro de datos o mapeo de eventos.

3.- Sistema según cualquiera de las reivindicaciones anteriores, que comprende adicionalmente un subsistema (7) de telemetría, configurado con una o más funcionalidades de traza, depuración y medida del rendimiento de la ejecución de algoritmos de computación cuántica en la pluralidad de los ordenadores cuánticos (300, 300', 300''), de la decoherencia de los mismos, y/o de su relación con los meta-algoritmos de computación cuántica.

4.- Sistema según la reivindicación anterior, donde el subsistema (7) de telemetría comprende uno o más módulos de aprendizaje automático o de inteligencia artificial para la medida del rendimiento de la ejecución de algoritmos de computación cuántica en los ordenadores cuánticos (300, 300', 300''), y de su relación con los meta-algoritmos de computación cuántica.

5.- Sistema según la reivindicación anterior, que comprende adicionalmente un subsistema (8) de gobierno alimentado por las salidas del subsistema (7) de telemetría, adaptado para proporcionar información de evolución, carga y depuración de la ejecución de algoritmos de computación cuántica en los ordenadores cuánticos (300, 300', 300'').

6.- Sistema según cualquiera de las reivindicaciones anteriores, donde los ordenadores cuánticos (300, 300', 300'') y el servidor (200) forman una red de computación distribuida o en la nube; y/o donde el servidor (200) forma parte de una agrupación de tipo cluster o granja de servidores, cooperativos o competitivos.

7.- Sistema según cualquiera de las reivindicaciones anteriores, donde uno o más de los ordenadores cuánticos (300, 300', 300'') comprenden una máquina virtual.

8.- Sistema según cualquiera de las reivindicaciones anteriores, donde el BPM (1) está configurado con medios de recepción de mensajes (9) de llamada a funciones remotas que parametrizan y encapsulan instrucciones enviadas por los clientes (100).

9.- Sistema según la reivindicación anterior, donde el BPM (1) comprende un primer servicio web (10) de recogida de los mensajes (9) de llamada y de almacenamiento de los mismos en una base de datos (11).

10.- Sistema según cualquiera de las reivindicaciones anteriores, donde el compilador (2), el transpilador (4) o el motor (5) de ejecución de unidades de procesamiento están implementados como uno o más plugins.

11.- Sistema según la reivindicación anterior, donde el BPM (1) comprende un módulo de carga (15) de los plugins.

12.- Sistema según la reivindicación anterior, donde el BPM (1):
- comprende un segundo servicio web (12) de procesamiento asíncrono de respuestas;
- está configurado con medios de envío de mensajes (13) de respuesta asíncrona al cliente (100); y/o
- comprende un procesador (14) de grupos de comandos, adaptado para el procesamiento de los comandos pendientes de ser ejecutados en la base de datos (11).

13.- Sistema según la reivindicación anterior, donde el procesador (14) de grupos de comandos, es programable en el tiempo, de forma recurrente y con una ejecución autónoma, en forma de demonios.

14.- Sistema según cualquiera de las reivindicaciones anteriores, donde uno o más de los ordenadores cuánticos (300, 300', 300'') comprenden un modelo de computación basado en puertas cuánticas.

15.- Sistema según cualquiera de las reivindicaciones anteriores, donde uno o más de los ordenadores cuánticos (300, 300', 300'') comprenden un modelo de computación basado en temple cuántico.

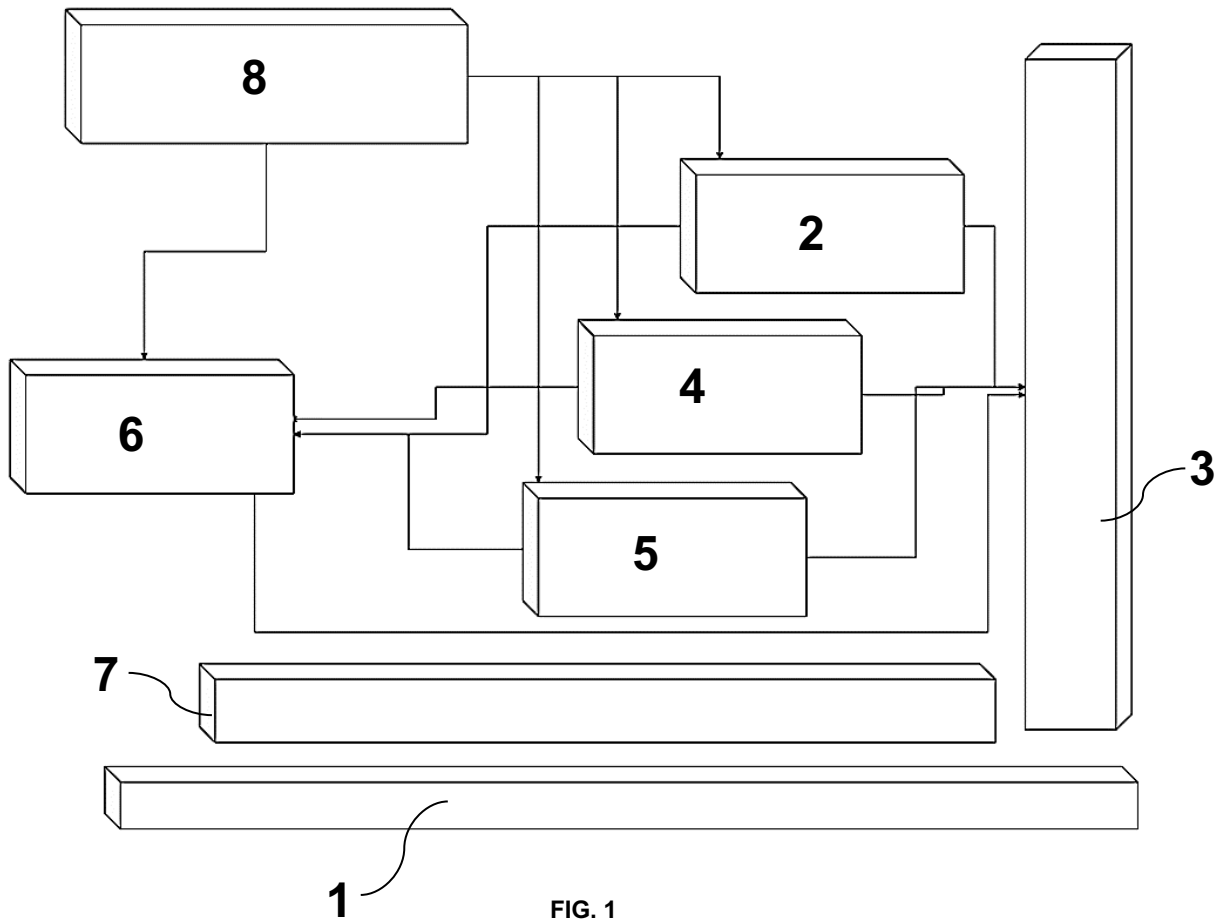


FIG. 1

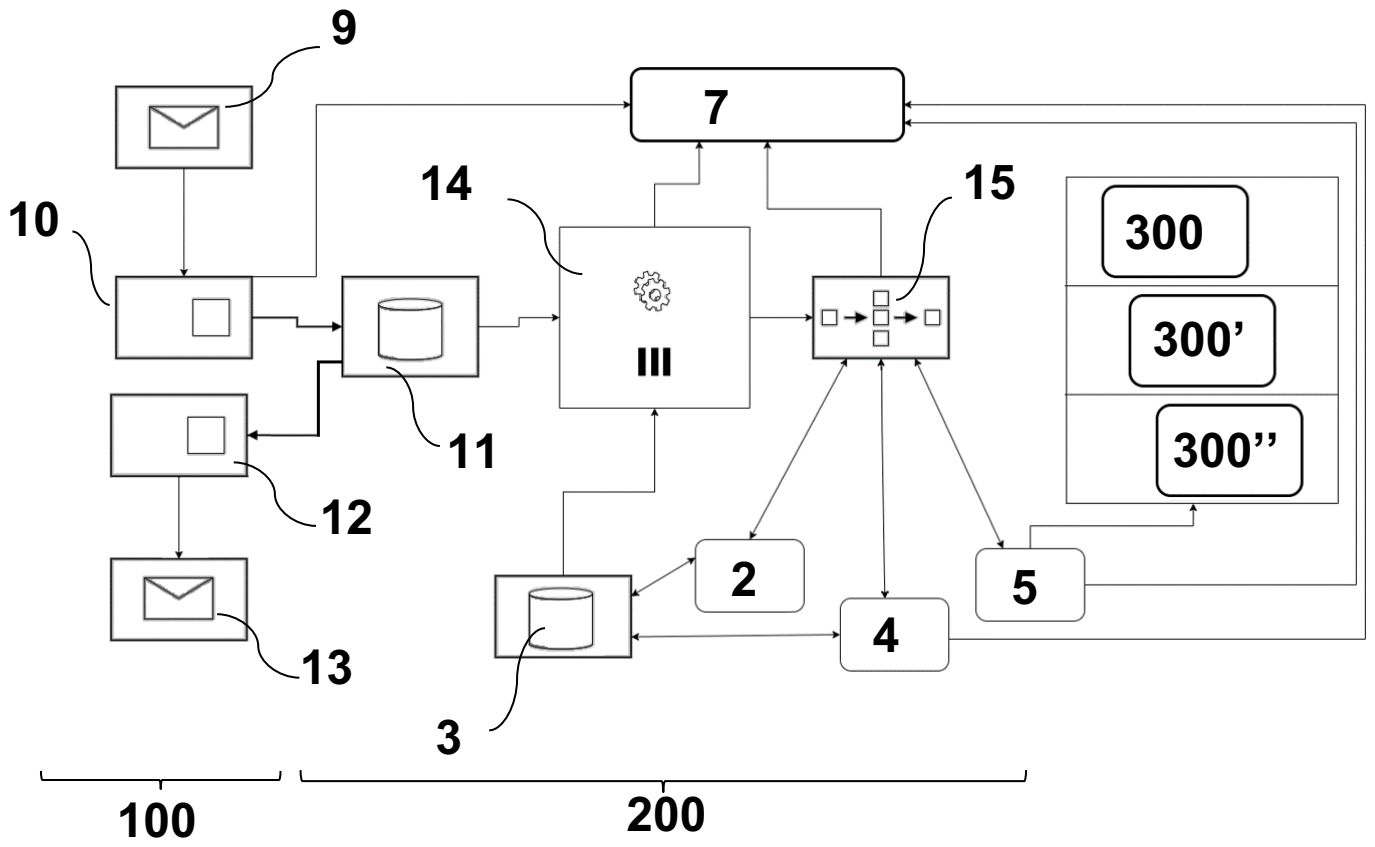


FIG. 2