

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 755 720**

51 Int. Cl.:

G11C 7/10 (2006.01)

G11C 16/06 (2006.01)

G06F 12/02 (2006.01)

G11C 16/10 (2006.01)

G06F 3/06 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **24.06.2016 PCT/CN2016/087096**

87 Fecha y número de publicación internacional: **28.12.2017 WO17219364**

96 Fecha de presentación y número de la solicitud europea: **24.06.2016 E 16898145 (4)**

97 Fecha y número de publicación de la concesión europea: **28.08.2019 EP 3296996**

54 Título: **Método de procesado de datos, aparato de almacenamiento, disco de estado sólido y sistema de almacenamiento**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
23.04.2020

73 Titular/es:

**HUAWEI TECHNOLOGIES CO., LTD. (100.0%)
Huawei Administration Building, Bantian,
Longgang District, Shenzhen
Guangdong 518129, CN**

72 Inventor/es:

**WU, LIMING;
YAO, JIANYE;
HUANG, BIN;
CAO, HONGQIANG;
XU, CHAO y
LI, YIBIN**

74 Agente/Representante:

ELZABURU, S.L.P

ES 2 755 720 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

Método de procesado de datos, aparato de almacenamiento, disco de estado sólido y sistema de almacenamiento

Campo técnico

5 La presente invención se refiere al campo de los sistemas de almacenamiento, y, en particular, a un método de procesado de datos, a un aparato de almacenamiento, a un disco de estado sólido y a un sistema de almacenamiento.

Antecedentes

10 Un sistema de almacenamiento incluye, en general, un anfitrión, un controlador (o al que se hace referencia como controlador de almacenamiento, o controlador de matrices de almacenamiento), y una matriz de discos. El controlador es un puente entre el anfitrión y la matriz de discos, y está configurado para controlar la comunicación y el intercambio de datos entre el anfitrión y un disco.

15 En la actualidad, todos los sistemas de almacenamiento convencionales soportan funciones de compresión y descompresión de datos. Por ejemplo, antes de que se escriban datos en un disco, el sistema de almacenamiento generalmente comprime los datos. Como otro ejemplo, después de que se lean datos de un disco, el sistema de almacenamiento, en primer lugar, descomprime los datos, y, a continuación, devuelve datos descomprimidos al anfitrión. La compresión y descompresión de datos puede ahorrar espacio del disco.

20 En la técnica anterior, las funciones de compresión y descompresión de datos del sistema de almacenamiento se implementan principalmente por medio del controlador, y de manera específica, se pueden implementar por medio de un procesador en el controlador. Por ejemplo, para una solicitud de escritura (o a la que se hace referencia como entrada/salida (IO) de escritura), en primer lugar el controlador comprime los datos, y, a continuación, escribe datos comprimidos en un disco. Para una solicitud de lectura (o a la que se hace referencia como IO de lectura), en primer lugar el controlador lee datos comprimidos de un disco, a continuación descomprime los datos leídos, y devuelve datos descomprimidos al anfitrión. A partir del proceso anterior puede deducirse que es necesario que el controlador participe en un proceso completo de compresión y descompresión. No obstante, la compresión y descompresión de datos necesitan consumir una gran cantidad de recursos informáticos, creando una mayor presión computacional sobre el controlador.

25 El documento US 2012/079174 A1 da a conocer un método para una interfaz directa entre un controlador de memoria y un controlador de memoria no volátil usando un protocolo de órdenes que incluye la recepción de una orden desde un controlador de memoria hacia un controlador de memoria no volátil a través de una interfaz por cable por medio de un protocolo de órdenes.

30 El documento US 2014/059278 A1 da a conocer técnicas de *software* de fabricación y microprogramas de dispositivos de almacenamiento que incluyen el acceso a imágenes de microprogramas y la comunicación de una herramienta de *software* de fabricación.

Sumario

35 La presente invención queda definida por las reivindicaciones independientes. En las reivindicaciones dependientes se definen aspectos preferidos. Esta invención proporciona un método de procesado de datos, un aparato de almacenamiento, un disco de estado sólido, y un sistema de almacenamiento, con el fin de reducir la presión computacional de un controlador.

40 Según un primer aspecto, se proporciona un método de procesado de datos, donde el método se aplica a un disco de estado sólido (Disco de Estado Sólido, SSD). Específicamente, el método puede ser ejecutado por el SSD, por ejemplo, puede ser ejecutado por un procesador en el SSD. El método incluye: recibir, por parte del SSD, una solicitud de escritura de un controlador, donde la solicitud de escritura es portadora de datos destinados a escribirse; y comprimir, por parte del SSD, los datos destinados a escribirse con el fin de obtener datos comprimidos. Debe entenderse que el SSD puede comprimir los datos destinados a escribirse de acuerdo con un bloque lógico, es decir, la cantidad de bloques comprimidos obtenidos por medio de la compresión depende de la cantidad de direcciones de bloques lógicos (Dirección de Bloque Lógico, LBA) asignadas a los datos destinados a escribirse. Adicionalmente, el SSD puede usar una compresión de longitud fija o una compresión de longitud variable. El método incluye, además: almacenar, por parte del SSD, los datos comprimidos; y enviar, por parte del SSD, una primera información de retroalimentación al controlador, donde la primera información de retroalimentación indica la capacidad restante del SSD después de almacenar los datos comprimidos. Debe entenderse que la primera información de retroalimentación puede indicar la capacidad restante según una pluralidad de modalidades, por ejemplo, en una modalidad de indicación directa o en una modalidad de indicación indirecta. Por ejemplo, la modalidad de indicación directa puede ser: el SSD notifica directamente la capacidad restante del SSD al controlador; la modalidad de indicación indirecta puede ser: el SSD notifica un cambio de la capacidad restante del SSD al controlador, y el controlador también puede determinar la capacidad restante del SSD en función del cambio de la capacidad restante del SSD.

55 En la técnica anterior, la compresión de datos es llevada a cabo por un controlador, y el propio controlador crea un

- espacio de direcciones para gestionar la capacidad de un SSD. El espacio de direcciones es un espacio de direcciones de longitud fija. Una vez que se ha creado el espacio de direcciones, el tamaño del espacio de direcciones no cambia. Como consecuencia, incluso si el SSD puede notificar al controlador un aumento de la capacidad aportado por la compresión de datos, el controlador no puede utilizar una capacidad mayor, lo cual da como resultado un derroche de recursos de almacenamiento del SSD. En esta modalidad de implementación, en primer lugar, se transfiere una función de compresión desde el controlador al SSD, de manera que puede reducirse la presión computacional del controlador; en segundo lugar, se transfiere una función de gestión de capacidad del SSD al SSD, y el SSD retroalimenta la capacidad restante al controlador. Puede que el SSD puede obtener de manera precisa su propia capacidad restante, se evita el derroche de recursos de almacenamiento del SSD.
- 5
- 10 En referencia al primer aspecto, en algunas modalidades de implementación del primer aspecto, el método incluye, además: enviar, por parte del SSD, una segunda información de retroalimentación al controlador, donde la segunda información de retroalimentación indica la capacidad lógica del SSD, y la capacidad lógica del SSD es la suma de un volumen de datos sin comprimir de datos almacenados en el SSD y la capacidad de espacio físico libre del SSD.
- 15 En referencia al primer aspecto, en algunas modalidades de implementación del primer aspecto, antes del envío, por parte del SSD, de la primera información de retroalimentación al controlador, el método incluye, además: consultar, por parte del SSD, el espacio físico libre del SSD; y determinar, por parte del SSD, la capacidad del espacio físico libre como capacidad restante.
- 20 En referencia al primer aspecto, en algunas modalidades de implementación del primer aspecto, el método incluye, además: asignar, por parte del SSD, un bloque lógico en blanco a los datos destinados a escribirse, donde el bloque lógico en blanco es un bloque lógico con una dirección lógica libre, y cada bloque lógico se corresponde con un segmento de capacidad lógica del SSD; convertir, por parte del SSD, el bloque lógico en blanco en un bloque lógico válido, donde el bloque lógico válido es un bloque lógico cuya dirección lógica está ocupada por datos válidos; y consultar, por parte del SSD, la cantidad de bloques lógicos en blanco restantes en el SSD; y el envío, por parte del SSD, de la primera información de retroalimentación al controlador, incluye: enviar, por parte del SSD, la primera información de retroalimentación al controlador, donde la primera información de retroalimentación incluye la cantidad de los bloques lógicos en blanco restantes y/o la capacidad restante, y la capacidad restante se determina sobre la base de la cantidad de los bloques lógicos en blanco restantes. Debe entenderse que, en esta modalidad de implementación, después de asignar el bloque lógico en blanco a los datos destinados a escribirse, el SSD puede convertir directamente el bloque lógico en blanco en el bloque lógico válido; o después de asignar el bloque lógico en blanco a los datos destinados a escribirse, el SSD puede llevar a cabo, en primer lugar, operaciones de compresión y almacenamiento sobre los datos destinados a escribirse, y, a continuación, puede convertir el bloque lógico en blanco en el bloque lógico válido.
- 25
- 30
- 35 En referencia al primer aspecto, en algunas modalidades de implementación del primer aspecto, el SSD graba el bloque lógico en el SSD de acuerdo con una tabla de mapeo del SSD.
- 40 En referencia al primer aspecto, en algunas modalidades de implementación del primer aspecto, la compresión, por parte del SSD, de los datos destinados a escribirse con el fin de obtener datos comprimidos, incluye: después de la asignación, por parte del SSD, de un bloque lógico en blanco a los datos destinados a escribirse, comprimir, por parte del SSD, los datos destinados a escribirse para obtener los datos comprimidos; y el método incluye, además: determinar, por parte del SSD de acuerdo con el tamaño de los datos destinados a escribirse y el tamaño de los datos comprimidos, la cantidad de bloques lógicos en blanco que es necesario añadir en el SSD.
- 45 En referencia al primer aspecto, en algunas modalidades de implementación del primer aspecto, el método incluye, además: añadir, por parte del SSD de acuerdo con la cantidad de los bloques lógicos en blanco que es necesario añadir en el SSD, entradas correspondientes a bloques lógicos en blanco nuevos a la tabla de mapeo del SSD.
- 50 En referencia al primer aspecto, en algunas modalidades de implementación del primer aspecto, el método incluye, además: insertar, por parte del SSD en función de la cantidad de los bloques lógicos en blanco que es necesario añadir en el SSD, entradas correspondientes a bloques lógicos en blanco nuevos al final de la tabla de mapeo del SSD.
- 55 Manteniendo la tabla de mapeo de una longitud que varíe dinámicamente, puede grabarse adecuadamente un cambio de la capacidad lógica del SSD.
- En referencia al primer aspecto, en algunas modalidades de implementación del primer aspecto, el método incluye, además: actualizar, por parte del SSD, un bloque lógico aislado en el SSD a un bloque lógico en blanco en función de la cantidad de los bloques lógicos en blanco que es necesario añadir en el SSD, donde el bloque lógico aislado es un bloque lógico cuya dirección lógica no se puede usar en el SSD.
- La actualización dinámica del estado de un bloque lógico en la tabla de mapeo puede facilitar la gestión del bloque lógico en el SSD.
- En referencia al primer aspecto, en algunas modalidades de implementación del primer aspecto, el método incluye, además: recibir, por parte del SSD, una orden del controlador, donde la orden incluye un intervalo de direcciones

lógicas, y la orden se usa para dar instrucciones con el fin de marcar datos válidos en el intervalo de direcciones lógicas como datos no válidos; y convertir, por parte del SSD, un bloque lógico válido del intervalo de direcciones lógicas en un bloque lógico aislado, donde el bloque lógico aislado es un bloque lógico cuya dirección lógica no se puede usar. La orden, por ejemplo, puede ser una orden de reajuste/desmapeo (*trim/unmap*).

- 5 En esta solución, los bloques lógicos se clasifican en bloques lógicos en blanco, bloques lógicos válidos, y bloques lógicos aislados. Mediante la actualización y el mantenimiento de estados de bloques lógicos en el SSD, no solamente puede reflejarse bien el almacenamiento de datos de los bloques lógicos en el SSD, sino que también puede reflejarse adecuadamente un cambio dinámico de la capacidad lógica del SSD.

- 10 En referencia al primer aspecto, en algunas modalidades de implementación del primer aspecto, el método incluye, además: determinar, por parte del SSD, el tamaño del espacio físico ocupado por datos en el intervalo de direcciones lógicas; seleccionar, por parte del SSD, un espacio de sobredimensionamiento parcial del espacio de sobredimensionamiento, donde el tamaño del espacio de sobredimensionamiento parcial es igual al tamaño del espacio físico ocupado por los datos en el intervalo de direcciones lógicas; y usar, por parte del SSD, el espacio de sobredimensionamiento parcial como espacio físico libre del SSD.

- 15 En referencia al primer aspecto, en algunas modalidades de implementación del primer aspecto, la cantidad M de bloques lógicos en blanco que es necesario añadir en el SSD es igual a un resultado, redondeado a la baja, de dividir D por L, donde D indica la diferencia entre la longitud de los datos destinados a escribirse y la longitud de los datos comprimidos, y L indica la longitud de un bloque lógico en el SSD. M se puede obtener por medio de una consulta en tabla, o se puede obtener por medio de un cálculo en línea.

- 20 En referencia al primer aspecto, en algunas modalidades de implementación del primer aspecto, la cantidad M de bloques lógicos en blanco que es necesario añadir en el SSD es igual al resultado de dividir D por L, menos 1, donde D indica la diferencia entre la longitud de los datos destinados a escribirse y la longitud de los datos comprimidos, y L indica la longitud de un bloque lógico en el SSD. M se puede obtener por medio de una consulta en tabla, o se puede obtener por medio de un cálculo en línea.

- 25 En referencia al primer aspecto, en algunas modalidades de implementación del primer aspecto, después de la recepción, por parte del SSD, de una solicitud de escritura de un controlador, el método incluye, además: asignar, por parte del SSD, una dirección lógica a los datos destinados a escribirse; y grabar, por parte del SSD, información de mapeo en una entrada que se encuentra en la tabla de mapeo y se corresponde con la dirección lógica, donde la información de mapeo incluye información sobre una página física que almacena los datos comprimidos, una posición de inicio de los datos comprimidos en la página física, y la longitud de los datos comprimidos.

La dirección física de los datos comprimidos se puede localizar directamente usando la información de mapeo anterior, con lo cual se mejora la eficiencia de direccionamiento del SSD.

- 35 En referencia al primer aspecto, en algunas modalidades de implementación del primer aspecto, después de la recepción, por parte del SSD, de una solicitud de escritura de un controlador, el método incluye, además: asignar, por parte del SSD, una dirección lógica a los datos destinados a escribirse; grabar, por parte del SSD, información de mapeo en una entrada que se encuentra en la tabla de mapeo y se corresponde con la dirección lógica, donde la dirección de mapeo incluye información sobre una página física que almacena los datos comprimidos; y almacenar, por parte del SSD, metadatos de los datos comprimidos en la página física, donde los metadatos incluyen una posición de inicio de los datos comprimidos en la página física y la longitud de los datos comprimidos.

- 40 La tabla de mapeo y la página física graban conjuntamente una relación de mapeo entre la dirección lógica y la dirección física de un bloque de datos, de manera que puede reducirse la memoria ocupada por la tabla de mapeo en el SSD, y se puede ahorrar espacio de memoria del SSD.

- 45 En referencia al primer aspecto, en algunas modalidades de implementación del primer aspecto, la solicitud de escritura incluye la longitud de datos correspondiente a los datos destinados a escribirse, y la dirección lógica de los datos destinados a escribirse se asigna basándose en la longitud de los datos.

En referencia al primer aspecto, en algunas modalidades de implementación del primer aspecto, después del almacenamiento, por parte del SSD, de los datos comprimidos, el método incluye, además: enviar, por parte del SSD, la dirección lógica de los datos destinados a escribirse, en el SSD, al controlador.

- 50 En referencia al primer aspecto, en algunas modalidades de implementación del primer aspecto, el método incluye, además: recibir, por parte del SSD, una solicitud de lectura del controlador, donde la solicitud de lectura incluye una dirección lógica de datos destinados a leerse; leer, por parte del SSD, datos objetivo (en inglés, *target data*) en una dirección física correspondiente a la dirección lógica de los datos destinados a leerse; descomprimir, por parte del SSD, los datos objetivo con el fin de obtener los datos destinados a leerse; y enviar, por parte del SSD, los datos destinados a leerse al controlador.

- 55 En referencia al primer aspecto, en algunas modalidades de implementación del primer aspecto, antes del envío, por parte del SSD, de la primera información de retroalimentación al controlador, el método puede incluir, además: recibir,

- 5 por parte del SSD, un mensaje de solicitud de consulta enviado por el controlador, donde la solicitud de consulta se usa para consultar la capacidad restante del SSD, y el envío, por parte del SSD, de la primera información de retroalimentación al controlador, incluye: enviar, por parte del SSD, un mensaje de respuesta de consulta al controlador, donde el mensaje de respuesta de consulta incluye la primera información de retroalimentación. Específicamente, el SSD recibe un mensaje de consulta enviado por el controlador, donde el mensaje de consulta se usa específicamente para consultar la cantidad de los bloques lógicos en blanco restantes en el SSD; y se envía, por parte del SSD, un mensaje de respuesta de consulta al controlador, donde el mensaje de respuesta de consulta se usa para indicar la cantidad de los bloques lógicos en blanco restantes.
- 10 En referencia al primer aspecto, en algunas modalidades de implementación del primer aspecto, el envío, por parte del SSD, de la primera información de retroalimentación al controlador, incluye: enviar activamente, por parte del SSD, la primera información de retroalimentación al controlador. Específicamente, el SSD puede enviar activamente la cantidad de los bloques lógicos en blanco restantes al controlador. Debe entenderse que el SSD puede retroalimentar periódicamente la cantidad de los bloques lógicos en blanco restantes al controlador, o la notificación de los bloques lógicos en blanco restantes se puede activar por medio de un evento.
- 15 De acuerdo con un segundo aspecto, se proporciona un aparato de almacenamiento, donde el aparato de almacenamiento incluye un módulo configurado para ejecutar el método en el primer aspecto.
- 20 De acuerdo con un tercer aspecto, se proporciona un disco de estado sólido, y el mismo incluye una interfaz de comunicaciones, un soporte de almacenamiento, y un procesador, donde la interfaz de comunicaciones está configurada para comunicarse con un controlador, el soporte de almacenamiento está configurado para proporcionar espacio de almacenamiento, y el procesador está conectado a la interfaz de comunicaciones y al soporte de almacenamiento, y ejecuta el método del primer aspecto usando la interfaz de comunicaciones y el soporte de almacenamiento.
- 25 Según un cuarto aspecto, se proporciona un soporte legible por ordenador, donde el soporte legible por ordenador almacena código de programa destinado a ser ejecutado por un disco de estado sólido, y el código de programa incluye una instrucción usada para ejecutar el método del primer aspecto.
- En las modalidades de implementación anteriores, la tabla de mapeo se puede usar para indicar un espacio de direcciones lógicas del SSD.
- En las modalidades de implementación anteriores, el bloque lógico en el SSD es una unidad básica del espacio de direcciones lógicas del SSD.
- 30 En las modalidades de implementación anteriores, la determinación, por parte del SSD, de la dirección lógica de los datos destinados a escribirse, puede incluir: asignar, por parte del SSD, N bloques lógicos en blanco a N bloques de datos en los datos destinados a escribirse, donde los bloques lógicos en blanco son bloques lógicos con direcciones lógicas libres en el SSD, la longitud de cada bloque de datos en los N bloques de datos es igual que la longitud de un bloque lógico en el SSD, y N es un entero que es superior o igual a 1; y usar, por parte del SSD, las direcciones lógicas de los N bloques lógicos en blanco como direcciones lógicas de los datos destinados a escribirse.
- 35 En la técnica anterior, todas las direcciones lógicas del SSD son asignadas y gestionadas por el controlador. No obstante, el SSD es el que mejor conoce su propio estado de almacenamiento y compresión de datos. Por lo tanto, en esta modalidad de implementación, las funciones de asignación y gestión de las direcciones lógicas del SSD se transfieren desde el controlador al SSD, de manera que las modalidades de asignación y gestión de las direcciones locales del SSD resultan más adecuadas.
- 40 En las anteriores modalidades de implementación, la asignación, por parte del SSD, de N bloques lógicos en blanco a N bloques de datos en los datos destinados a escribirse, puede incluir: seleccionar aleatoriamente, por parte del SSD, N bloques lógicos en blanco de entre bloques lógicos en el SSD.
- 45 En las anteriores modalidades de implementación, la asignación, por parte del SSD, de N bloques lógicos en blanco a N bloques de datos en los datos destinados a escribirse, puede incluir: seleccionar, por parte del SSD, N bloques lógicos en blanco en la parte delantera de acuerdo con las posiciones de entradas correspondientes a los bloques lógicos en blanco de la tabla de mapeo.
- 50 En las anteriores modalidades de implementación, el SSD puede segmentar los datos destinados a escribirse de acuerdo con una granularidad de compresión preestablecida, para obtener N bloques de datos, donde la granularidad de compresión es igual a la longitud de un bloque de datos en el SSD, y N es un entero que es mayor de 1. La granularidad de compresión puede indicar específicamente una unidad de compresión de un bloque de datos en el SSD, es decir, el tamaño usado como unidad para la compresión.
- La granularidad de compresión se fija a un valor igual a la longitud del bloque lógico, con lo cual se simplifica la implementación dentro del SSD, y se facilita la gestión de la tabla de mapeo.
- 55 En la técnica anterior, la compresión de datos la lleva a cabo un controlador, y el propio controlador crea un espacio

de direcciones para gestionar la capacidad de un SSD. El espacio de direcciones es un espacio de direcciones de longitud fija. Una vez que se ha creado el espacio de direcciones, el tamaño del espacio de direcciones no cambia. Como consecuencia, incluso si el SSD puede notificar al controlador un aumento de capacidad aportado por la compresión de datos, el controlador no puede utilizar una capacidad mayor, lo cual da como resultado un derroche de recursos de almacenamiento del SSD. En la presente invención, en primer lugar, se transfiere una función de compresión desde el controlador a un SSD, de manera que puede reducirse la presión computacional del controlador; en segundo lugar, la función de gestión de capacidad del SSD se transfiere al SSD, y el SSD retroalimenta la capacidad restante al controlador. Debido a que el SSD puede obtener de manera precisa su propia capacidad restante, se evita el derroche de recursos de almacenamiento del SSD.

10 **Breve descripción de los dibujos**

Para describir más claramente las soluciones técnicas de la presente invención, lo siguiente describe de manera breve los dibujos adjuntos que muestran realizaciones preferidas de la presente invención. Aparentemente, los dibujos adjuntos de la siguiente descripción muestran meramente algunas realizaciones de la presente invención.

- 15 La FIG. 1 es un diagrama de flujo del procesado, por parte de un sistema de almacenamiento, de una solicitud de escritura de un anfitrión;
- la FIG. 2 es un diagrama de flujo del procesado, por parte de un sistema de almacenamiento, de una solicitud de escritura de un anfitrión;
- la FIG. 3 es un diagrama estructural esquemático de un sistema de almacenamiento según una realización de la presente invención;
- 20 la FIG. 4 es un diagrama de un ejemplo de una tabla de mapeo según una realización de la presente invención;
- la FIG. 5 es un diagrama de transición de estados de un bloque lógico de acuerdo con una realización de la presente invención;
- la FIG. 6 es un diagrama de un ejemplo de mapeo entre un bloque lógico y un bloque físico según una realización de la presente invención;
- 25 la FIG. 7 es un diagrama de un ejemplo de un procedimiento de compresión en un sistema de compresión de longitud fija según una realización de la presente invención;
- la FIG. 8 es un diagrama de un ejemplo de una modalidad de redondear bloques comprimidos según una realización de la presente invención;
- 30 la FIG. 9 es un diagrama de un ejemplo de una modalidad de almacenamiento de información de mapeo según una realización de la presente invención;
- la FIG. 10 es un diagrama de un ejemplo de una modalidad de almacenamiento de información de mapeo según una realización de la presente invención;
- la FIG. 11 es un diagrama estructural de un ejemplo de una página física según una realización de la presente invención;
- 35 la FIG. 12 es un diagrama de flujo del procesado de una solicitud de escritura según una realización de la presente invención;
- la FIG. 13 es un diagrama de flujo esquemático de una modalidad de implementación de la etapa 1214 en una compresión de longitud fija;
- 40 la FIG. 14 es un diagrama de flujo esquemático de una modalidad de implementación de la etapa 1214 en una compresión de longitud variable;
- la FIG. 15 es un diagrama de un ejemplo de un procedimiento de reajuste/desmapeo según una realización de la presente invención;
- la FIG. 16 es un diagrama de flujo esquemático del procesado de una solicitud de lectura según una realización de la presente invención;
- 45 la FIG. 17 es un diagrama estructural esquemático de un aparato de almacenamiento según una realización de la presente invención;
- la FIG. 18 es un diagrama estructural esquemático de un sistema de almacenamiento según una realización de la presente invención;
- 50 la FIG. 19 es un diagrama estructural esquemático de un disco de estado sólido según una realización de la presente invención; y
- la FIG. 20 es un diagrama estructural esquemático de un sistema de almacenamiento según una realización de la presente invención.

Descripción de las realizaciones

55 Lo siguiente describe de manera clara y completa las soluciones técnicas de la presente invención en referencia a los dibujos adjuntos que muestran realizaciones preferidas de la presente invención. Aparentemente, las realizaciones descritas constituyen una parte, más que la totalidad, de las realizaciones de la presente invención.

Debe señalarse que un disco (disco) en las realizaciones de la presente invención puede hacer referencia a un SSD, y una matriz de discos puede hacer referencia a una matriz de SSD.

60 Para facilitar su comprensión, en primer lugar se describe brevemente, en referencia a la FIG. 1 y la FIG. 2, una modalidad de compresión de datos en un sistema de almacenamiento existente.

La FIG. 1 es un diagrama de flujo de procesado, por parte de un sistema de almacenamiento, de una solicitud de escritura (o a la que se hace referencia como IO) de un anfitrión. El sistema de almacenamiento de la FIG. 1 es un sistema de almacenamiento basado en una compresión de longitud fija. La compresión de longitud fija puede comprimir un bloque de datos en uno o más bloques comprimidos de longitud fija. Por ejemplo, suponiendo que la granularidad de compresión del sistema de almacenamiento es 4 KB (es decir, 4 KB se usa como unidad para la compresión), el sistema de almacenamiento basado en una compresión de longitud fija comprime un bloque de datos de 4 KB en un múltiplo entero de 0,5 KB. Por ejemplo, el bloque de datos de 4 KB se comprime en una de las siguientes granularidades: 0,5 KB, 1 KB, 1,5 KB, 2 KB, 3 KB, 3,5 KB, ó 4 KB (cuando el tamaño comprimido del bloque de datos de 4 KB es mayor de 3,5 KB, no es necesario comprimir el bloque de datos). Específicamente, suponiendo que la longitud comprimida del bloque de datos de 4 KB está entre 0,5 KB y 1 KB, es necesario llevar a cabo una operación de alineamiento hacia arriba sobre el bloque de datos para obtener un bloque de datos de 1 KB. Una manera específica de alineamiento hacia arriba puede ser la adición de 0 al final de los datos comprimidos, para rellenar los datos comprimidos hasta 1 KB. Debe entenderse que la granularidad de compresión de un bloque de datos también puede ser cualquiera otra granularidad, tal como 8 KB ó 16 KB, y un bloque de datos de cada granularidad de compresión se puede comprimir en un múltiplo entero de 0,5 KB, un múltiplo entero de 1 KB, o similares. Esto no queda limitado específicamente en las realizaciones de la presente invención. Lo siguiente usa, principalmente, un ejemplo en el cual, con una granularidad de compresión de 4 KB, un bloque destinado a comprimirse de 4 KB se comprime en uno de 0,5 KB, 1 KB, 1,5 KB, 2 KB, 3 KB, 3,5 KB ó 4 KB con vistas a su descripción.

102. Un anfitrión envía una solicitud de escritura a un controlador.

20 La solicitud de escritura puede incluir datos destinados a escribirse en un disco. Al controlador también se le puede hacer referencia como controlador de almacenamiento, y el mismo puede gestionar discos que proporcionan espacio de almacenamiento físico. Si los discos forman una matriz, al controlador también se le puede hacer referencia como controlador de matriz.

104. El controlador segmenta, de acuerdo con una longitud fija, datos destinados a escribirse en un disco.

25 Específicamente, los datos destinados a escribirse en el disco se pueden segmentar de acuerdo con la granularidad de compresión, con el fin de generar un bloque de datos. Por ejemplo, si la granularidad de compresión es 4 KB, 4 KB se usa como unidad para segmentar los datos destinados a escribirse en el disco.

106. El controlador comprime un bloque de datos segmentado, y lleva a cabo una operación de alineamiento hacia arriba sobre los datos comprimidos.

30 Suponiendo que la longitud de los datos comprimidos formados después de que se comprima un bloque de datos de 4 KB es menor de 0,5 KB, puede llevarse a cabo una operación de redondeo por exceso sobre los datos comprimidos, y los datos comprimidos se rellenan hasta un bloque comprimido de 0,5 KB.

108. El controlador redondea bloques comprimidos, y selecciona un disco adecuado.

35 Por ejemplo, los bloques comprimidos se pueden redondear en una unidad de un tamaño de una página física (en este caso, como ejemplo para la descripción se usa una página física de 4 KB). Puede generarse una solicitud de escritura cada vez que se obtiene un bloque de datos de 4 KB por medio del redondeo. A continuación, los datos incluidos en una o más solicitudes de escritura generadas se pueden escribir en el disco seleccionado a la vez. Alternativamente, después de que se obtenga una pluralidad de bloques de datos de 4 KB por medio del redondeo, los bloques de datos se escriben en el disco seleccionado a la vez.

40 Adicionalmente, es necesario que el controlador seleccione un disco para almacenar los datos redondeados. Puede haber una pluralidad de políticas o modalidades para buscar un disco. Por ejemplo, el controlador puede seleccionar un disco aleatoriamente de entre una matriz de discos, siempre que se garantice que el espacio restante del disco sea mayor que la longitud de datos destinados a escribirse en el disco. Alternativamente, el controlador selecciona un disco adecuado basándose en factores, tales como el volumen actual de almacenamiento de datos de los discos de la matriz de discos, de manera que los datos almacenados en los discos se igualan.

110. El controlador envía una solicitud de escritura al disco seleccionado.

La solicitud de escritura puede incluir los datos destinados a escribirse en el disco, y una dirección lógica del disco asignada por el controlador a los datos en la solicitud de escritura.

112. Después de haberse procesado en las etapas anteriores, los datos finalmente llegan al disco, y el disco escribe los datos en una dirección física del mismo, y devuelve una respuesta de éxito de solicitud de escritura al controlador.

114. El controlador retroalimenta la respuesta de éxito de solicitud de escritura al anfitrión.

La FIG. 2 es un diagrama de flujo del procesado, por parte de un sistema de almacenamiento, de una solicitud de escritura de un anfitrión. El sistema de almacenamiento de la FIG. 2 es un sistema de almacenamiento basado en una compresión de longitud variable. A diferencia de la compresión de longitud fija, la compresión de longitud variable no

requiere comprimir un bloque de datos a una longitud fija. Como ejemplo sigue usándose una granularidad de compresión de 4 KB, y, suponiendo que la longitud comprimida de un bloque de datos de 4 KB es 2.778 B, no es necesario ajustar el bloque de datos de 2.778 B, es decir, la longitud comprimida no varía. Las etapas 202 a 214 de la FIG. 2 son básicamente similares a las etapas 102 a 114 de la FIG. 1, y para evitar repeticiones, no se describen de nuevo detalles en la presente. Una de las diferencias principales entre la FIG. 1 y la FIG. 2 reside en que la compresión de longitud variable se lleva a cabo sobre el bloque de datos en la etapa 206, pero la compresión de longitud fija se lleva a cabo sobre el bloque de datos en la etapa 106.

Tanto la FIG. 1 como la FIG. 2 describen un procedimiento de procesado de solicitudes de escritura. Cuando el anfitrión necesita leer datos del disco, el anfitrión envía una solicitud de lectura al controlador. Después de recibir la solicitud de lectura, el controlador en primer lugar obtiene datos del disco, a continuación descomprime los datos, y devuelve datos descomprimidos al anfitrión. El procedimiento de procesado de solicitudes de lectura es un proceso inverso al procedimiento de procesado de solicitudes de escritura, y para evitar repeticiones, no se describen detalles de nuevo en la presente.

Basándose en la descripción anterior, puede deducirse que las funciones de compresión y descompresión son implementadas principalmente por el controlador en la técnica anterior. Como la escala de procesado de datos del sistema de almacenamiento es cada vez mayor, las funciones de compresión y descompresión necesitan consumir una gran cantidad de recursos informáticos del controlador. Como consecuencia, el retardo en el procesado de la solicitud de escritura por parte del controlador es mayor, y, además, se reduce el rendimiento global del sistema de almacenamiento.

Con el desarrollo de las tecnologías, el SSD se convierte en uno de los soportes de almacenamiento usado principalmente por una matriz de discos. Una de las diferencias entre el SSD y una unidad de disco duro (Unidad de Disco Duro, HDD) convencional reside en que el SSD incluye un procesador que tiene capacidad de procesado de datos. En la técnica anterior, el procesador de SSD está configurado principalmente para llevar a cabo funciones simples, tales como gestión de bloques, recolección de basura, nivelación de desgaste, y conversión entre una dirección lógica y una dirección física. No obstante, con el desarrollo de las tecnologías, la capacidad de procesado del procesador en el SSD se hace cada vez más fuerte, y no se puede sacar el máximo partido de la capacidad de procesado del procesador en el SSD si dicho SSD se usa para llevar a cabo únicamente las funciones anteriores. Por lo tanto, para utilizar en su totalidad la capacidad de procesado del procesador en el SSD, y reducir la carga del controlador, las funciones de compresión y descompresión de datos se puede transferir al SSD. Por lo tanto, en las realizaciones de la presente invención, las funciones tanto de compresión como de descompresión de datos pueden ser implementadas por el SSD. Lo siguiente describe de forma detallada una estructura de composición de un sistema de almacenamiento en una realización de la presente invención y las funciones de los componentes en el sistema de almacenamiento en referencia a la FIG. 3.

La FIG. 3 es un diagrama estructural esquemático de un sistema de almacenamiento de acuerdo con una realización de la presente invención. El sistema de almacenamiento de la FIG. 3 incluye un anfitrión (anfitrión) 310, un controlador 320, y un SSD 330. Tal como puede observarse a partir de la FIG. 3, el controlador 320 está situado entre el anfitrión 310 y el SSD 330, y en calidad de puente, implementa el intercambio de datos entre el anfitrión 310 y el SSD 330.

Específicamente, el controlador 320 puede incluir componentes tales como una unidad de procesado central (Unidad de Procesado Central, CPU) (u otro chip que tenga capacidad de cálculo, por ejemplo, una matriz de puertas programable in situ (Matriz de Puertas Programable in Situ, FPGA)) y una memoria caché (Memoria Caché). El controlador 320 de la FIG. 3 está separado del anfitrión. No obstante, esta realización de la presente invención no se limita a ello. El controlador 320 puede ser un dispositivo del *hardware* independiente, o puede ser un módulo de *software* integrado en el anfitrión 310. Adicionalmente, el controlador 320 puede estar conectado a uno o más SSDs.

Todavía en referencia a la FIG. 3, el SSD puede incluir un chip de memoria *flash* (el chip de memoria *flash* es un soporte de almacenamiento en el SSD). El SSD (o un procesador del SSD) puede incluir una unidad de gestión de bloques, configurada para gestionar bloques en el SSD, donde un bloque (bloque) es una unidad de borrado básica en el SSD. Por ejemplo, el SSD mantiene un identificador de un bloque en el SSD usando la unidad de gestión de bloques, borra un bloque después de la recolección de basura, y graba y sustituye un bloque malo en el SSD. El SSD puede incluir, además, una unidad de conversión de direcciones lógicas a direcciones físicas, configurada para implementar el direccionamiento de datos dentro del SSD. Una unidad de recolección de basura está configurada para llevar a cabo una recolección de basura sobre datos no válidos del SSD, por ejemplo, mover datos válidos de un bloque a otro bloque. Una unidad de nivelación de desgaste está configurada para controlar los ciclos de escritura/borrado de bloques en el SSD, de manera que los grados de desgaste (a saber, ciclos de escritura/borrado) de los bloques en el SSD sean lo más uniformes posible.

Además, para implementar las funciones de compresión y descompresión de datos en el SSD, puede añadirse una unidad de compresión y descompresión al SSD. Específicamente, cuando recibe una solicitud de escritura del controlador, el SSD en primer lugar puede comprimir datos destinados a escribirse de la solicitud de escritura, y, a continuación, escribir datos comprimidos en el chip de memoria *flash* del SSD. Cuando recibe una solicitud de lectura del controlador, el SSD puede leer datos del chip de memoria *flash* del SSD, a continuación descomprimir los datos leídos, y devolver datos descomprimidos al controlador.

La compresión de datos en el SSD viene acompañada de un cambio de la capacidad lógica del SSD. Específicamente, se supone que el controlador tiene conocimiento de que la capacidad lógica del SSD es 400 GB, pero, en el SSD, los datos de 400 GB recibidos se comprimen de acuerdo con una relación de compresión de 4:1, y se obtienen datos comprimidos de 100 GB. Si el controlador no consigue tener conocimiento de que el disco todavía puede almacenar 300 GB de datos en este caso, es decir, el disco todavía tiene una capacidad lógica de 300 GB en este caso, se provoca un derroche de recursos del SSD. Lo siguiente explica una de las causas del problema.

En primer lugar, para una mejor descripción, se describen, en primer lugar, tres tipos de capacidades en relación con un SSD, es decir, la capacidad del soporte, la capacidad física y la capacidad lógica.

Capacidad del soporte: capacidad total del chip de memoria *flash* en el SSD. Por ejemplo, para un SSD con 400 GB nominales, la capacidad de un chip de memoria *flash* en el SSD es en general mayor de 400 GB, por ejemplo, puede ser 512 GB. Es decir, con una capacidad del soporte de 512 GB, se dispone de un espacio de sobredimensionamiento (OP) de 112 GB, o denominado espacio redundante. El espacio de sobredimensionamiento se usa para mejorar la fiabilidad del SSD, y, en general, no puede ser utilizado por un usuario. Ciertamente, las relaciones de redundancia usadas cuando fabricantes diferentes diseñan los SSD son diferentes, de manera que también cambia el tamaño del espacio de sobredimensionamiento del SSD. El espacio de sobredimensionamiento se puede usar para el movimiento de datos durante la sustitución de bloques malos o la recolección de basura.

Capacidad física: capacidad disponible nominal del SSD. Es equivalente a la capacidad de un soporte de almacenamiento que se puede usar para almacenar datos en el SSD por parte del usuario. Por ejemplo, para un SSD con 400 GB nominales, la capacidad física correspondiente es 400 GB. Con independencia de si el SSD comprime datos antes de que los datos se almacenen, la capacidad física del SSD no cambia.

Capacidad lógica: volumen total de datos que se puede almacenar realmente en el SSD, o capacidad del SSD que se puede usar para almacenar datos. Si el SSD no comprime los datos, la capacidad lógica es en general igual a la capacidad física. Si el SSD comprime los datos antes de almacenar los mismos, la capacidad lógica del SSD es igual a la suma de la capacidad de espacio físico libre del SSD y un volumen de datos no comprimidos de los datos almacenados en el SSD. Por lo tanto, la capacidad lógica cambia dinámicamente. Por ejemplo, para un SSD con una capacidad física de 400 GB, el tamaño de los datos es 1.200 GB, a continuación los datos se comprimen, el tamaño de los datos comprimidos almacenados realmente en el SSD es 400 GB, por lo tanto la capacidad lógica del SSD es 1.200 GB. La capacidad lógica está relacionada con la relación de compresión de datos almacenados en el disco, y cambia dinámicamente a medida que se almacenan datos diferentes.

Para facilitar su comprensión, lo siguiente describe una manera de calcular un cambio de la capacidad lógica del SSD de forma independiente usando, como ejemplo, una compresión de longitud fija y una compresión de longitud variable.

Suponiendo que la capacidad física nominal de un SSD es 400 GB, la capacidad lógica inicial del SSD es igual a 400 GB. Sigue usándose como ejemplo una granularidad de compresión de 4 KB, para un estado de compresión de un bloque de datos de 4 KB, el incremento de la capacidad lógica del SSD se puede calcular de las maneras diversas sucesivas:

1. Un bloque de datos de 4 KB se comprime en un bloque comprimido de 0,5 KB: el incremento de la capacidad lógica del SSD es $4 \text{ KB} - 0,5 \text{ KB} = 3,5 \text{ KB}$.

2. Un bloque de datos de 4 KB se comprime en un bloque comprimido de 1 KB: el incremento de la capacidad lógica del SSD es $4 \text{ KB} - 1 \text{ KB} = 3 \text{ KB}$.

3. Un bloque de datos de 4 KB se comprime en un bloque comprimido de 1,5 KB: el incremento de la capacidad lógica del SSD es $4 \text{ KB} - 1,5 \text{ KB} = 2,5 \text{ KB}$.

4. Un bloque de datos de 4 KB se comprime en un bloque comprimido de 2 KB: el incremento de la capacidad lógica del SSD es $4 \text{ KB} - 2 \text{ KB} = 2 \text{ KB}$.

5. Un bloque de datos de 4 KB se comprime en un bloque comprimido de 2,5 KB: el incremento de la capacidad lógica del SSD es $4 \text{ KB} - 2,5 \text{ KB} = 1,5 \text{ KB}$.

6. Un bloque de datos de 4 KB se comprime en un bloque comprimido de 3 KB: el incremento de la capacidad lógica del SSD es $4 \text{ KB} - 3 \text{ KB} = 1 \text{ KB}$.

7. Un bloque de datos de 4 KB se comprime en un bloque comprimido de 3,5 KB: el incremento de la capacidad lógica del SSD es $4 \text{ KB} - 3,5 \text{ KB} = 0,5 \text{ KB}$.

Lo siguiente proporciona un ejemplo específico de cálculo del incremento de la capacidad lógica del SSD.

Suponiendo que hay 4 M bloques de datos de 4 KB, la longitud total de estos bloques de datos es 16 GB.

Suponiendo que un total de 1 M bloques de datos de estos bloques de datos se comprimen a 2 KB cada uno, el incremento de la capacidad lógica del SSD es: $1 \text{ M} * (4 \text{ KB} - 2 \text{ KB}) = 2 \text{ GB}$.

Suponiendo que un total de 1 M bloques de datos de estos bloques de datos se comprimen a 1,5 KB cada uno, el incremento de la capacidad lógica del SSD es: $1\text{ M} \cdot (4\text{ KB} - 1,5\text{ KB}) = 2,5\text{ GB}$.

Suponiendo que un total de 1 M bloques de datos de estos bloques de datos se comprimen a 1 KB cada uno, el incremento de la capacidad lógica del SSD es: $1\text{ M} \cdot (4\text{ KB} - 1\text{ KB}) = 3\text{ GB}$.

- 5 Suponiendo que un total de 1 M bloques de datos de estos bloques de datos se comprimen a 0,5 KB cada uno, el incremento de la capacidad lógica del SSD es: $1\text{ M} \cdot (4\text{ KB} - 0,5\text{ KB}) = 3,5\text{ GB}$.

En conclusión, en el SSD se almacena un total de 16 GB de datos. Debido a la compresión, la capacidad física ocupada realmente por los 16 GB de datos es $2\text{ GB} + 1,5\text{ GB} + 1\text{ GB} + 0,5\text{ GB} = 5\text{ GB}$. En este caso, el incremento de la capacidad lógica del SSD es $2\text{ GB} + 2,5\text{ GB} + 3\text{ GB} + 3,5\text{ GB} = 11\text{ GB}$, y la capacidad lógica cambia a 411 GB.

- 10 En la compresión de longitud variable, los tamaños comprimidos de un bloque de datos de 4 KB pueden ser longitudes variables comprimidas, tales como 2.778 B, 1.678 B, 2.012 B y 1.212 B. Para un estado de compresión de un bloque de datos de 4 KB, lo siguiente proporciona un ejemplo de una manera de calcular el incremento de la capacidad lógica del SSD.

- 15 1. Un bloque de datos de 4 KB se comprime en un bloque comprimido de 2.778 B: el incremento de la capacidad lógica del SSD es $4.096\text{ B} - 2.778\text{ B} = 1.318\text{ B}$.

2. Un bloque de datos de 4 KB se comprime en un bloque comprimido de 1.678 B: el incremento de la capacidad lógica del SSD es $4.096\text{ B} - 1.678\text{ B} = 2.418\text{ B}$.

3. Un bloque de datos de 4 KB se comprime en un bloque comprimido de 2.012 B: el incremento de la capacidad lógica del SSD es $4.096\text{ B} - 2.012\text{ B} = 2.084\text{ B}$.

- 20 4. Un bloque de datos de 4 KB se comprime en un bloque comprimido de 1.212 B: el incremento de la capacidad lógica del SSD es $4.096\text{ B} - 1.212\text{ B} = 2.884\text{ B}$.

A partir de la descripción anterior puede deducirse que, con independencia de que se trate de una compresión de longitud fija o una compresión de longitud variable, la capacidad lógica del SSD cambia dinámicamente de acuerdo con el estado de compresión de los bloques de datos. Ciertamente, además de la compresión de datos, la operación de reajuste/desmapeo también puede hacer que cambie la capacidad lógica del SSD. La operación de reajuste/desmapeo se describe posteriormente de manera detallada.

- 25 En conclusión, la capacidad lógica del SSD cambia dinámicamente en función del estado de compresión de los datos. Si el controlador no puede tener conocimiento del cambio dinámico de la capacidad lógica del SSD, y si el controlador no puede percibir el incremento de la capacidad lógica, pero sigue considerando la capacidad lógica del SSD como invariable igual que en la técnica anterior, el espacio de almacenamiento ahorrado por medio de la compresión no se puede utilizar en su totalidad, y se provoca un derroche de espacio de almacenamiento del SSD. Por lo tanto, cuando las funciones de compresión y descompresión se transfieren al SSD es necesario considerar el problema de cómo diseñar un conjunto de soluciones que permitan una cooperación entre el controlador y el SSD y que posibiliten que el SSD detecte y gestione la capacidad lógica que cambia dinámicamente y presente el cambio dinámico de la capacidad lógica del SSD al controlador.

Para resolver el problema anterior, al SSD se le pueden añadir una unidad 331 de gestión de tablas de mapeo de longitud variable y una unidad 332 de gestión de estados de bloques lógicos (en referencia a la FIG. 3), y, al controlador, se le puede añadir una unidad 321 de gestión de direcciones lógicas de SSD. Lo siguiente describe funciones de estas unidades de manera detallada.

- 40 La unidad 331 de gestión de tablas de mapeo de longitud variable se puede configurar para gestionar y mantener una tabla de mapeo del SSD. La tabla de mapeo refleja el espacio de direcciones lógicas del SSD. La tabla de mapeo incluye una o más entradas (o a las que se hace referencia como ítems). Cada entrada se corresponde con un intervalo de direcciones lógicas del espacio de direcciones lógicas. El intervalo de direcciones lógicas correspondiente a cada entrada en la tabla de mapeo es la granularidad básica de la tabla de mapeo. Por ejemplo, suponiendo que cada entrada se corresponde con un intervalo de direcciones lógicas de 4 KB, la granularidad básica de la tabla de mapeo es 4 KB. En general, cada entrada se corresponde con un bloque lógico. Al intervalo de direcciones lógicas correspondiente a cada entrada se le hace referencia como dirección de bloque lógico (Dirección de Bloque Lógico, LBA). Adicionalmente, la tabla de mapeo se puede usar para mapear una LBA con una dirección de bloque físico (Dirección de Bloque Físico, PBA), y la relación de mapeo grabada en la tabla de mapeo se puede usar para implementar el direccionamiento de datos. En esta realización de la presente invención, la longitud de la tabla de mapeo del SSD puede cambiar dinámicamente a medida que cambia la capacidad lógica del SSD.

- 55 Además, los procesos de compresión y descompresión en el SSD se pueden enmascarar usando la tabla de mapeo. Específicamente, la tabla de mapeo del SSD refleja un estado de almacenamiento de datos en el espacio de direcciones lógicas del SSD, y el tamaño de un espacio de direcciones lógicas ocupado depende del tamaño original (tamaño sin comprimir) de los datos. Es decir, en el almacenamiento de los datos en el espacio de direcciones lógicas

no existe ningún fenómeno de compresión, y esto es equivalente a enmascarar el proceso de compresión de los datos por parte del SSD usando la tabla de mapeo.

Adicionalmente, la granularidad básica de la tabla de mapeo se puede mantener congruente con la granularidad de compresión. La tabla de mapeo puede estar ubicada en la memoria del SSD.

5 Todavía en referencia a la FIG. 3, la unidad 332 de gestión de estados de bloques lógicos puede estar configurada para gestionar y mantener el estado de un bloque lógico en el SSD. Específicamente, el SSD puede incluir bloques lógicos en tres estados: un bloque lógico en blanco (o al que se hace referencia como bloque en blanco), un bloque lógico válido (o al que se hace referencia como bloque válido) y un bloque lógico aislado (o al que se hace referencia como bloque aislado). Debe entenderse que un bloque lógico en blanco puede referirse a un bloque lógico con una dirección lógica libre. La cantidad de bloques lógicos en blanco puede reflejar la capacidad lógica restante del SSD; o la cantidad de bloques lógicos en blanco puede reflejar el espacio disponible restante del SSD. Adicionalmente, la cantidad de los bloques lógicos en blanco puede cambiar dinámicamente a medida que cambie la capacidad lógica del SSD. Por lo tanto, el cambio dinámico de la cantidad de los bloques lógicos en blanco se puede usar como parámetro del cambio dinámico de la capacidad lógica del SSD. Un bloque lógico válido puede referirse a un bloque lógico cuya dirección lógica almacena datos válidos, y los datos válidos pueden referirse a datos que no están reajustados o desmapeados. Un bloque lógico aislado puede referirse a un bloque lógico cuya dirección lógica no puede ser usada, por ejemplo, un bloque lógico cuya dirección lógica está ocupada por datos no válidos. Los datos no válidos pueden ser, por ejemplo, datos que están marcados como no válidos por una operación de reajuste/desmapeo.

20 Específicamente, la tabla de mapeo del SSD puede usar la forma que se muestra en la FIG. 4 para presentar un espacio de direcciones lógicas unidimensional del SSD. Una casilla cuadrada de la FIG. 4 representa la granularidad básica de la tabla de mapeo. La granularidad básica puede ser 4 KB/8 KB/16 KB/32 KB/64 KB, o similares. La longitud del espacio unidimensional de direcciones lógicas presentado por la tabla de mapeo puede ser variable, es decir, puede aumentar o disminuir dinámicamente. Adicionalmente, la tabla de mapeo puede incluir un bloque lógico aislado, y el bloque lógico aislado de la FIG. 4 se identifica con rayas diagonales. Los datos de un intervalo de direcciones lógicas correspondiente al bloque lógico aislado son datos no válidos, y la dirección lógica correspondiente al bloque lógico aislado no se puede usar temporalmente.

Un bloque lógico en el SSD puede transitar entre los siguientes tres estados. Lo siguiente describe un proceso de transición de estados de un bloque lógico en el SSD, de manera detallada y en referencia a la FIG. 5.

30 En primer lugar, cuando una dirección lógica correspondiente a un bloque lógico no se usa, el bloque lógico es un bloque lógico en blanco. Si la dirección lógica de un bloque lógico en blanco se usa para almacenar datos válidos, el bloque lógico en blanco cambia a bloque lógico válido.

35 A continuación, el bloque lógico válido puede cambiar a un bloque lógico aislado debido a una operación de reajuste/desmapeo (para obtener una descripción detallada sobre la operación de reajuste/desmapeo, véase el siguiente procedimiento de reajuste/desmapeo). Considérese el siguiente escenario: en primer lugar, un lote de datos con una relación de compresión de 4 se escribe en el SSD, y, a continuación, un lote de datos que no se pueden comprimir se escribe en el SSD. Suponiendo que la capacidad física del SSD es 400 GB, cuando, en el SSD, se escriben los datos con la relación de compresión de 4, el SSD puede almacenar realmente 1.600 GB de datos. En este caso, la capacidad lógica del SSD se amplía de 400 GB a 1.600 GB. En este caso, el espacio de direcciones lógicas reflejado por la tabla de mapeo es 1.600 GB. A continuación, se actualizan datos del SSD, y, en dicho SSD, se almacenan 400 GB de datos que no se pueden comprimir. En este caso, el SSD puede almacenar realmente solo 400 GB de datos. En este caso, la capacidad lógica correspondiente al SSD se reduce de 1.600 GB a 400 GB nuevamente. Durante la reducción de la capacidad lógica, una parte del espacio de direcciones lógicas ciertamente se convierte en no válida, y, en el espacio de direcciones lógicas, se genera un bloque lógico aislado.

45 A continuación, cuando aumenta la capacidad lógica del SSD, el bloque lógico aislado se puede actualizar nuevamente a un bloque lógico en blanco. Específicamente, cuando aumenta la capacidad lógica del SSD, aumenta el intervalo de direcciones lógicas del SSD, y, de manera correspondiente, aumenta la cantidad de los bloques lógicos en blanco del SSD. En este caso, preferentemente puede considerarse la actualización de un bloque lógico aislado en la tabla de mapeo a un bloque lógico en blanco. Cuando todos los bloques lógicos aislados de la tabla de mapeo se actualizan a bloques lógicos en blanco, se pueden añadir al final de la tabla de mapeo entradas correspondientes a bloques lógicos en blanco nuevos, es decir, se añaden los bloques lógicos en blanco.

55 La tabla de mapeo puede grabar una relación de mapeo de una dirección lógica a una dirección física del SSD. La relación de mapeo puede reflejarse específicamente por una relación de mapeo entre una dirección lógica de un bloque lógico y una dirección física de un bloque físico. El tamaño de un bloque físico en el SSD se puede fijar de acuerdo con los requisitos. Por ejemplo, un bloque físico puede incluir: 1/4 de página física, 1/2 de página física, una página física, una pluralidad de páginas físicas, un bloque (bloque) o una pluralidad de bloques. La relación de mapeo de un bloque lógico a un bloque físico puede ser un mapeo de una pluralidad de bloques lógicos a un bloque físico, es decir, una relación de mapeo de muchos-a-uno. Debe entenderse que el SSD puede dividir un intervalo de direcciones físicas del SSD en uno o más bloques físicos en función del tamaño del bloque físico. La FIG. 6 muestra uno de los bloques físicos. En referencia a la FIG. 6, con independencia de si el SSD soporta la compresión de longitud

fija o la compresión de longitud variable, cada bloque lógico puede corresponderse con un bloque físico, y, en cada bloque físico, se puede almacenar una pluralidad de bloques comprimidos.

La información de mapeo de la tabla de mapeo puede indicar una ubicación de almacenamiento de un bloque comprimido en un bloque físico. Puede existir una pluralidad de formas específicas de información de mapeo. Por ejemplo, la información de mapeo puede incluir información, tal como información sobre una página física que almacena un bloque comprimido (por ejemplo, un identificador o un número de la página física), una posición de inicio del bloque comprimido en la página física, y una longitud del bloque comprimido. En esta modalidad de indicación, el bloque comprimido se puede localizar rápidamente, y la eficiencia de direccionamiento es alta. Opcionalmente, la información de mapeo puede incluir la información sobre la página física que almacena el bloque comprimido. La posición de inicio del bloque comprimido en la página física y la longitud del bloque comprimido se pueden almacenar en la página física como metadatos, por ejemplo, se pueden almacenar en un encabezamiento de la página física. Puesto que la tabla de mapeo se almacena generalmente en el espacio de memoria del SSD, si únicamente se almacena información de páginas físicas de la información de mapeo, la longitud de la tabla de mapeo puede reducirse, y puede ahorrarse espacio de memoria del SSD. Con independencia de si el SSD soporta una compresión de longitud variable o una compresión de longitud fija, para la información de mapeo en el SSD pueden usarse una o más de las anteriores modalidades de implementación. Esto no presenta limitaciones específicas en esta realización de la presente invención. Posteriormente se describen detalles en referencia a un ejemplo específico.

Todavía en referencia a la FIG. 3, la unidad 321 de gestión de direcciones lógicas del SSD en el controlador se puede configurar para monitorizar o consultar la capacidad lógica restante del SSD, o un cambio de la capacidad lógica del SSD. Específicamente, la unidad 321 de gestión de direcciones lógicas del SSD puede tener conocimiento del cambio de la capacidad lógica del SSD monitorizando o consultando la cantidad de los bloques lógicos en blanco del SSD. El controlador puede monitorizar o consultar la cantidad de bloques lógicos en blanco en un SSD según una pluralidad de maneras. Lo siguiente proporciona varias soluciones opcionales. En la práctica, puede seleccionarse una o más de las siguientes soluciones.

Solución 1: el controlador detecta una cantidad de bloques lógicos en blanco en un SSD en tiempo real. Específicamente, cuando el controlador detecta que la cantidad (debe entenderse que, en la presente, cantidad se refiere a una cantidad de bloques lógicos en blanco restantes en un SSD que es grabada por el controlador, y la cantidad puede ser diferente a la cantidad de bloques lógicos en blanco restantes, reales, del SSD) de bloques lógicos en blanco del SSD que es grabada por el controlador es inferior a un umbral especificado (por ejemplo, 100), el controlador consulta activamente, del SSD, la cantidad de los bloques lógicos en blanco. Por ejemplo, en una consulta previa, el controlador tiene conocimiento de que un SSD incluye 500 bloques lógicos en blanco. En un proceso subsiguiente de escritura de datos, el controlador puede seguir actualizando la cantidad de bloques lógicos en blanco restantes del SSD que es grabada por el controlador. Cuando el controlador halla que la cantidad de los bloques lógicos en blanco restantes en el SSD que es grabada por el controlador es inferior a 100, el controlador puede consultar activamente, del SSD, la cantidad real de bloques lógicos en blanco restantes en el SSD.

Solución 2: el controlador determina, en función del tamaño de un volumen de datos correspondiente a un área de reajuste/desmapeo, si consultar la cantidad de bloques lógicos en blanco en un SSD. Por ejemplo, el controlador graba un volumen de datos reajustados o desmapeados, y cada vez que el volumen de datos reajustados o desmapeados llega a 1 GB, el controlador consulta activamente la cantidad de los bloques lógicos en blanco en el SSD.

Solución 3: el controlador determina, en función del volumen de datos escrito en un SSD, si consultar la cantidad de bloques lógicos en blanco en el SSD. Por ejemplo, cada vez que el controlador escribe un volumen de datos de 1 GB en el SSD, el controlador consulta activamente la cantidad de los bloques lógicos en blanco en el SSD.

Solución 4: un SSD notifica activamente la cantidad de bloques lógicos en blanco. Por ejemplo, el SSD puede notificar la cantidad de los bloques lógicos en blanco en el SSD periódicamente; o cuando un cambio de la cantidad de los bloques lógicos en blanco supera un umbral, el SSD notifica la cantidad de los bloques lógicos en blanco actuales al controlador.

Lo siguiente describe de forma detallada una modalidad de implementación de un proceso de compresión en un SSD de manera independiente usando, como ejemplo, la compresión de longitud fija y la compresión de longitud variable.

Modalidad de implementación de la compresión de longitud fija

En primer lugar, se describe, en referencia a la FIG. 7, un procedimiento de procesamiento de compresión en un sistema de compresión de longitud fija. Tal como se muestra en la FIG. 7, por encima de la línea de trazos se muestra un sistema que incluye un controlador; y, por debajo de la línea de trazos, se localiza un disco que incluye uno o más SSDs. En la FIG. 7, como ejemplo para la descripción sigue usándose una granularidad de compresión de 4 KB. En la práctica, pueden usarse también otras granularidades de compresión, tales como 8 KB y 16 KB, y las mismas no presentan limitaciones específicas en esta realización de la presente invención.

Específicamente, el controlador, en primer lugar, puede recibir datos que es necesario almacenar en el disco desde un anfitrión, y a continuación puede aplicar una deduplicación sobre los datos (es decir, eliminar datos duplicados; en relación con una modalidad de deduplicación específica, consúltese la técnica anterior). Seguidamente, el controlador

5 no comprime datos deduplicados, sino que envía directamente los datos deduplicados al SSD, y el SSD lleva a cabo el procesado de compresión. El SSD comprime los datos usando 4 KB como unidad, y el tamaño de los datos comprimidos varía con la relación de compresión de los datos. Puesto que el SSD usa la modalidad de compresión de longitud fija, es necesario llevar a cabo una operación de alineamiento hacia arriba sobre los datos comprimidos, de manera que la longitud de los datos comprimidos sea un múltiplo entero de 0,5 KB. Por ejemplo, si el tamaño de los datos comprimidos está entre 0,5 KB y 1 KB, de acuerdo con el principio de alineamiento hacia arriba, es necesario rellenar los datos comprimidos hasta 1 KB, y se obtiene un bloque comprimido de 1 KB.

10 A continuación, es necesario escribir el bloque comprimido en una dirección física del SSD. Debe entenderse que el SSD usa generalmente una página física como unidad básica de lectura o escritura. El tamaño de la página física está relacionado con el tipo del SSD, y no presenta limitaciones específicas en esta realización de la presente invención. En esta realización de la presente invención, como ejemplo para la descripción se usa una página física de 4 KB. Antes de que se escriban los datos en una página física, los bloques comprimidos pueden redondearse (o reagruparse) usando 4 KB como unidad, para redondear los bloques comprimidos redondeados a un tamaño de 4 KB. De esta manera, puede escribirse adecuadamente, cada vez, un grupo de bloques comprimidos redondeados en una página física de 4 KB. La FIG. 8 proporciona algunas modalidades de redondeo de bloques comprimidos. Lo siguiente proporciona, además, algunas modalidades de implementación opcionales de redondeo de una pluralidad de bloques comprimidos en una página física.

Dos bloques comprimidos se redondean en una página física:

(0,5 KB, 3,5 KB), (1 KB, 3 KB), (1,5 KB, 2,5 KB), (2 KB, 2 KB).

20 Tres bloques comprimidos se redondean en una página física:

(1 KB, 1 KB, 2 KB), (0,5 KB, 1,5 KB, 2 KB), (0,5 KB, 1 KB, 2,5 KB), (0,5 KB, 0,5 KB, 3 KB).

Cuatro bloques comprimidos se redondean en una página física:

(0,5 KB, 0,5 KB, 1,5 KB, 1,5 KB), (0,5 KB, 1 KB, 0,5 KB, 2 KB), (0,5 KB, 1,5 KB, 1 KB, 1 KB), (0,5 KB, 2,5 KB, 0,5 KB, 0,5 KB).

25 Para datos recibidos desde el controlador, el SSD no solamente necesita alinear y comprimir los datos con una granularidad de compresión, sino que también necesita asignar una dirección lógica de los datos en el SSD a los datos, y grabar una relación de mapeo entre la dirección lógica y una dirección física de los datos.

30 Específicamente, en referencia a la FIG. 9, puede grabarse información de mapeo en una entrada de una tabla de mapeo, y el SSD puede convertir directamente la dirección lógica en la dirección física de los datos usando la información de mapeo, para direccionar los datos. La información de mapeo puede incluir información sobre una página física, una dirección de inicio de un bloque comprimido en la página física, y la longitud del bloque comprimido.

Debe señalarse que la información sobre la página física puede ser un número de la página física. Una página física puede tener un número globalmente exclusivo; o un bloque físico tiene un número globalmente exclusivo, pero el número de la página física es exclusivo en el bloque físico.

35 Si la información de mapeo grabada en la entrada de la tabla de mapeo incluye los tres tipos de información anteriores, para una página física de 4 KB, un SSD con una capacidad del soporte de 512 GB requiere aproximadamente 34 bits (bit) para grabar una información de mapeo. El análisis específico es el siguiente:

Para la página física de 4 KB, puesto que 512 GB se corresponden con 128 M 4 KB, se requieren aproximadamente 28 bits para el direccionamiento de la página física de 4 KB.

40 Para la dirección de inicio del bloque comprimido en la página física, puesto que se lleva a cabo un alineamiento de límites en la dirección de inicio del bloque comprimido de acuerdo con los 0,5 KB en la modalidad de implementación de compresión de longitud fija, existen ocho posibilidades de la dirección de inicio del bloque comprimido en la página física, a saber, direcciones de inicio que se corresponden con 0, 0,5 KB, 1 KB, 1,5 KB, 2 KB, 2,5 KB, 3 KB y 3,5 KB. En este caso, todas las posibilidades de la dirección de inicio del bloque comprimido en la página física se pueden indicar usando 3 bits.

Para la longitud del bloque comprimido, puesto que todos los bloques comprimidos son múltiplos enteros de 0,5 KB, existen ocho posibilidades de la longitud del bloque comprimido, a saber, 0,5 KB, 1 KB, 1,5 KB, 2 KB, 2,5 KB, 3 KB, 3,5 KB y 4 KB. Por lo tanto, todas las posibilidades de la longitud del bloque comprimido se pueden indicar usando 3 bits.

50 A partir de la descripción anterior puede deducirse que, para ubicar un bloque comprimido (por ejemplo, un bloque comprimido de 0,5 KB ó un 1,5 KB), se requiere una información de mapeo de aproximadamente 34 bits. Debe indicarse que la cantidad de bits requerida por una información de mapeo varía de acuerdo con la capacidad y/o la granularidad de compresión del SSD. Esto no presenta limitaciones específicas en esta realización de la presente invención.

Modalidad de implementación de compresión de longitud variable

En la compresión de longitud variable, no es necesario llevar a cabo una operación de alineamiento hacia arriba sobre los datos comprimidos, y el tamaño de un bloque comprimido se puede mantener igual y no alinearse hacia arriba. Como ejemplo siguen usándose 4 KB, y suponiendo que los tamaños de los datos comprimidos de un bloque de datos de 4 KB son 2.778 B y 3.600 B, las longitudes de los bloques comprimidos se pueden mantener a 2.778 B y 3.600 B.

En referencia a la FIG. 10, suponiendo que la longitud de una página física es 16 KB, los bloques comprimidos de diferentes tamaños que se obtienen por medio de una compresión de longitud variable se pueden colocar en la página física secuencialmente hasta que no pueda colocarse un bloque comprimido sucesivo completo. Al final de la página física puede existir un espacio de almacenamiento parcial que no se puede usar. El espacio de almacenamiento parcial es demasiado pequeño para contener un bloque comprimido completo. Al espacio del almacenamiento parcial que no se puede usar se le puede hacer referencia como fragmento (fragmento), y el fragmento no puede almacenar datos válidos.

Suponiendo que la granularidad básica de una tabla de mapeo es 4 KB, y que el tamaño de la página física es 16 KB, en el SSD se puede grabar una relación de mapeo entre una dirección lógica y una dirección física de un bloque de datos de 4 KB usando una cualquiera de las tres siguientes soluciones.

Solución 1: la información de mapeo en una tabla de mapeo incluye información sobre una página física que almacena un bloque comprimido. Un encabezamiento de la página física almacena metadatos, y los metadatos indican una dirección de inicio del bloque comprimido en la página física y una longitud del bloque comprimido. Cuando va a leerse el bloque comprimido, en primer lugar puede hallarse la página física correspondiente de acuerdo con la información de mapeo almacenada en la tabla de mapeo; a continuación se leen en conjunto los datos en la página física, y, en la información de encabezamiento de la página física, se buscan la dirección de inicio y la longitud del bloque comprimido en la página física. Específicamente, la información de encabezamiento de la página física puede almacenar metadatos de una pluralidad de bloques comprimidos. Para ayudar a la búsqueda de metadatos de un bloque comprimido, en la información de encabezamiento de la página física se puede establecer una correspondencia entre una LBA del bloque de datos de 4 KB y metadatos. En este caso, el SSD puede buscar, de acuerdo con la LBA del bloque de datos, los metadatos correspondientes al bloque comprimido.

Solución 2: de manera similar a la Solución 1, en la Solución 2, la información sobre una página física de un bloque comprimido se almacena en una tabla de mapeo, y en la página física se almacenan metadatos del bloque comprimido. Una de las diferencias reside en que la tabla de mapeo almacena no solamente la información sobre la página física del bloque comprimido, sino también información de indexación. La información de indexación puede describir una posición del bloque comprimido en una secuencia de todos los bloques comprimidos almacenados en la página física. De esta manera, cuando se buscan los metadatos de la página física, no es necesario que el SSD ubique los metadatos del bloque comprimido buscando en una LBA compleja como se hace en la Solución 1, sino que puede ubicar los metadatos de acuerdo con la información de indexación, reduciéndose así el volumen de datos de los metadatos y ahorrándose espacio de almacenamiento de la página física.

Solución 3: la información de mapeo en una tabla de mapeo incluye información sobre una página física que almacena un bloque comprimido, una dirección de inicio del bloque comprimido en la página física, y la longitud del bloque comprimido. En esta solución, no es necesario almacenar metadatos en la página física, de manera que se simplifica la modalidad de implementación de la página física; adicionalmente, se omite un proceso de búsqueda de metadatos de acuerdo con la información de mapeo, mejorándose así la eficiencia de direccionamiento de una dirección física.

Lo siguiente describe la anterior solución 1 de forma detallada y en referencia a la FIG. 11. En referencia a la FIG. 11, una página física se divide en tres áreas. El área superior es un área de encabezamiento de la página física, usada para almacenar información de encabezamiento. El área central es un área de almacenamiento de datos, y almacena una pluralidad de bloques comprimidos. El área inferior es un área fragmentaria. Las áreas se describen a continuación por separado.

1. Área de encabezamiento

En el área de encabezamiento de la página física se almacena información de encabezamiento. La información de encabezamiento incluye metadatos de un bloque comprimido almacenado en la página física. El formato de los metadatos puede ser: LBA + Dirección inicio de la página física + Longitud del bloque comprimido. Ciertamente, si ya existe información de indexación en la tabla de mapeo, el formato de los metadatos puede ser: Información de Indexación + Dirección de inicio de la página física + Longitud del bloque comprimido. Adicionalmente, la información de mapeo también se puede almacenar en el espacio de sobredimensionamiento del SSD, de manera que puede ahorrarse espacio de almacenamiento de la página física.

2. Área de almacenamiento de datos

El área de almacenamiento de datos se usa para almacenar bloques comprimidos. Los bloques comprimidos se pueden almacenar juntos de manera compacta en la página física.

3. Área fragmentaria

El área fragmentaria es un área de almacenamiento no usada, es demasiado pequeña para almacenar un bloque comprimido completo, y por lo tanto simplemente se puede dejar de lado.

5 Lo anterior describe modalidades de implementación de compresión de longitud fija y de compresión de longitud variable. En referencia a la FIG. 12, lo siguiente describe de manera detallada un procedimiento de procesado de solicitudes de escritura durante la implementación de compresión de datos en un SSD.

10 La FIG. 12 es un diagrama de flujo de procesado de una solicitud de escritura de acuerdo con una realización de la presente invención. Debe entenderse que las etapas u operaciones mostradas en la FIG. 12 son solamente ejemplos. En esta realización de la presente invención, también pueden llevarse a cabo otras operaciones o variantes de diversas operaciones en la FIG. 12. Adicionalmente, las etapas de la FIG. 12 se pueden llevar a cabo de acuerdo con una secuencia diferente a la presentada en la FIG. 12, y, posiblemente, no es necesario llevar a cabo todas las operaciones en la FIG. 12.

1202. Un anfitrión envía una solicitud de escritura a un controlador.

15 Específicamente, la solicitud de escritura puede incluir la siguiente información: LUN_ID+LBA+LEN+DATA. DATA indica datos destinados a escribirse. LUN_ID indica un número correspondiente a un número de unidad lógica (Número de Unidad Lógica, LUN). En este caso, LBA indica una dirección lógica de DATA en el LUN, en lugar de una dirección lógica de un SSD. La dirección lógica del LUN se refiere en general a una LBA de una matriz de discos presentada al anfitrión. LEN indica una longitud de datos, y DATA son los datos destinados a escribirse. Debe entenderse que la LBA y la LEN definidas en un sistema se producen de manera general según cada sector, y el tamaño de un sector es generalmente 0,5 KB. Suponiendo que la LEN es 10, puede entenderse que la longitud de DATA es $10 \times 0,5 \text{ KB} = 5 \text{ KB}$. Ciertamente, cada sector puede tener además otros tamaños o formatos. Esto no presenta limitaciones específicas en esta realización de la presente invención. En la presente, como ejemplo para la descripción se usa un tamaño de sector de 0,5 KB.

25 1204. El controlador lleva a cabo un procesado de alineamiento de límites sobre los datos destinados a escribirse, y segmenta, de acuerdo con un tamaño fijo, datos obtenidos después del procesado de alineamiento de límites. Esta es una etapa opcional.

El procesado de alineamiento de límites sobre los DATA por parte del controlador es para facilitar la segmentación subsiguiente de los datos, de manera que el controlador pueda obtener una cantidad entera de bloques de datos segmentados cuando segmenta los datos de acuerdo con el tamaño fijo.

30 El controlador puede segmentar los datos según una pluralidad de modalidades. Lo siguiente proporciona dos modalidades de segmentación opcionales.

35 Modalidad de segmentación 1: el controlador lleva a cabo una segmentación de acuerdo con la granularidad de compresión del SSD, de manera que cada bloque de datos segmentado es una unidad comprimible. Por ejemplo, se supone que la granularidad de compresión del SSD es 4 KB, y que la LEN de los datos destinados a escribirse es 10 (es decir, 5 KB). En este caso, en primer lugar el controlador puede llevar a cabo un procesado de alineamiento de límites sobre los datos de 5 KB para obtener datos de 8 KB, y, a continuación, puede segmentar los datos de 8 KB con una granularidad de 4 KB, con el fin de obtener dos bloques de datos de 4 KB. Una de las modalidades de llevar a cabo el procesado de límites sobre los datos de 5 KB por parte del controlador puede ser: por ejemplo, el controlador lee 3 KB de datos del SSD, y redondea los datos y los 5 KB de datos para obtener los 8 KB de datos.

40 Modalidad de segmentación 2: el controlador lleva a cabo una segmentación basándose en una unidad máxima de datos escritos que pueden ser procesados por el SSD a la vez. Por ejemplo, si la unidad máxima de datos escritos que puede ser procesada por el SSD a la vez es 128 KB, en este caso la segmentación puede llevarse a cabo acorde a los 128 KB. Esta modalidad de procesado simplifica operaciones del controlador. Posteriormente, el SSD continúa con la segmentación de los datos de 128 KB de acuerdo con la granularidad de compresión. Por ejemplo, un procesador del SSD segmenta los datos de 128 KB entre treinta dos bloques de datos de 4 KB.

Debe indicarse que, después de que se segmenten los datos, el controlador puede enviar una solicitud de escritura al SSD usando un bloque de datos segmentado como unidad. En este caso, pueden determinarse parámetros, tales como LUN_ID+LBA+LEN+DATA de cada bloque de datos segmentado, sobre la base de parámetros tales como LUN_ID+LBA+LEN+DATA de datos completos antes de la segmentación.

50 1206. El controlador selecciona, de una matriz de discos, un SSD para almacenar bloques de datos segmentados.

55 El controlador puede seleccionar un SSD de acuerdo con una pluralidad de políticas, por ejemplo, puede seleccionar un SSD adecuado basándose en la nivelación del desgaste entre SSDs, en un procesado de tolerancia a fallos, en una relación de redundancia, y similares. Debe indicarse que, durante la selección del SSD, en primer lugar es necesario garantizar la existencia de suficientes bloques lógicos en blanco en el SSD. La cantidad de bloques lógicos en blanco en el SSD en la matriz de discos varía dinámicamente a medida que cambia la capacidad lógica del SSD.

El controlador puede consultar la cantidad de los bloques lógicos en blanco en el SSD de acuerdo con la anterior solución de consulta de bloques lógicos en blanco.

1208. El controlador envía, al SSD, una solicitud de escritura correspondiente a cada bloque de datos segmentado.

5 La solicitud de escritura puede ser portadora de información de LEN+DATA del bloque de datos segmentado. LEN indica la longitud de datos en la solicitud de escritura, y DATA indica el contenido de datos (o a lo que se hace referencia como los datos propiamente dichos, es decir, el bloque de datos) en la solicitud de escritura. Debe entenderse que, en la presente, puede que no se especifique la ubicación del SSD en la cual es necesario escribir los datos de la solicitud de escritura, es decir, la ubicación de almacenamiento real de los datos en el SSD puede ser determinada por el propio SSD.

10 1210. El SSD lleva a cabo un procesado de segmentación sobre datos de la solicitud de escritura.

15 Debe entenderse que la etapa 1210 es una etapa opcional. Si el controlador segmenta los datos de acuerdo con la granularidad de compresión en la etapa 1204, el tamaño de los datos en la solicitud de escritura recibida por el SSD es una unidad de compresión básica, de manera que no es necesario llevar cabo nuevamente una segmentación, y puede realizarse directamente una operación subsiguiente. Si el controlador segmenta los datos según N veces la granularidad de compresión en la etapa 1204, el tamaño de los datos recibidos por el SSD es N veces la granularidad de compresión, y los datos se pueden segmentar en N bloques de datos de acuerdo con la granularidad de compresión en la etapa 1210. En un caso especial, si el tamaño de un bloque de datos recibido por el SSD no es un múltiplo entero de la unidad de compresión, el SSD en primer lugar puede rellenar el bloque de datos hasta un múltiplo entero de la unidad de compresión, y, a continuación, puede llevar a cabo la segmentación de acuerdo con el tamaño de la unidad de compresión.

20 1212. El SSD asigna un bloque lógico en blanco a un bloque de datos segmentado.

25 El SSD puede consultar estados de bloques lógicos en el SSD, y seleccionar un bloque lógico en blanco para almacenar el bloque de datos. Un bloque lógico en blanco puede seleccionarse según una pluralidad de modalidades. Por ejemplo, puede seleccionarse aleatoriamente un bloque lógico en blanco del SSD, o puede seleccionarse preferentemente un bloque lógico en blanco con una dirección lógica en la parte delantera de acuerdo con una secuencia de bloques lógicos en blanco de una tabla de mapeo. Esto no presenta limitaciones específicas en esta realización de la presente invención.

30 El estado de un bloque lógico se puede indicar mediante un mapa de bits. El mapa de bits se puede almacenar en una memoria del SSD. Después de que el bloque de datos se almacene en el bloque lógico en blanco, puede variarse el estado del bloque lógico en blanco, de manera que el bloque lógico en blanco cambia a un bloque lógico válido.

1214. El SSD comprime el bloque de datos segmentado, escribe un bloque comprimido en el SSD, y graba información de mapeo en una tabla de mapeo.

35 Las modalidades de ejecución de la etapa 1214 por un SSD que soporte compresión de longitud fija y un SSD que soporte compresión de longitud variable pueden ser diferentes, y se describen posteriormente por separado en referencia a la FIG. 13 y la FIG. 14.

1216. El SSD ajusta el estado del bloque lógico en función de la variación de la capacidad lógica.

40 Específicamente, un bloque lógico aislado se convierte en un bloque lógico en blanco cada vez que la capacidad lógica aumenta en un tamaño de un bloque lógico. El bloque lógico aislado puede ser un bloque lógico aislado seleccionado aleatoriamente de entre todos los bloques aislados en el SSD; o el bloque lógico aislado puede ser un bloque lógico aislado con una dirección lógica en la parte delantera de todos los bloques aislados del SSD. Si en el SSD no existe ningún bloque lógico aislado, al final de la tabla de mapeo se puede añadir una entrada correspondiente a un bloque lógico en blanco. Esto es equivalente a añadir un bloque lógico en blanco en el SSD.

45 Después de que se haya completado el ajuste del estado del bloque lógico, puede enviarse al controlador la cantidad de bloques lógicos en blanco actuales en el SSD, para indicar la capacidad lógica restante del SSD. Por ejemplo, se supone que el SSD tiene en este momento 1.000 bloques lógicos en blanco (cada bloque lógico en blanco se corresponde con un espacio físico real de 4 KB, y hay en total un espacio lógico disponible real de 1.000*4 KB). Cuando el controlador consulta la cantidad de los bloques lógicos en blanco incluidos en el SSD, el SSD devuelve un resultado de consulta al controlador: actualmente hay 1.000 bloques lógicos en blanco. Por lo tanto, el controlador puede escribir mil bloques de datos de 4 KB en el SSD. Se supone que 500 de los 1.000 bloques de datos escritos por el controlador en el SSD se pueden comprimir a 2 KB cada uno de ellos, y que los otros 500 bloques de datos se pueden comprimir a 1 KB cada uno. Originalmente, es necesario consumir los 1.000 bloques lógicos en blanco para almacenar los bloques de datos. Actualmente solo se consumen $(500*2+500*1)/4=375$ bloques lógicos en blanco debido a la compresión. Es decir, únicamente se consumen 375 bloques lógicos en blanco para almacenar los mil bloques de datos de 4 KB. A continuación, cuando el controlador consulta la cantidad de bloques lógicos en blanco disponibles nuevamente, el SSD devuelve un resultado de consulta: en este momento, hay disponibles $1.000-375=625$ bloques lógicos en blanco.

1218. El SSD envía un mensaje de respuesta para la solicitud de escritura al controlador.

Después de que el bloque de datos de la solicitud de escritura se escriba en el SSD, el SSD envía el mensaje de respuesta al controlador. El mensaje de respuesta puede contener una dirección lógica, por ejemplo, una LBA, del bloque de datos en el SSD. Una de las formas específicas del mensaje de respuesta está relacionada con la modalidad de segmentación descrita en la etapa 1204.

Si el controlador segmenta los datos de acuerdo con la granularidad de compresión en la etapa 1204, no es necesario que el SSD segmente adicionalmente los datos, sino que puede asignar directamente una dirección lógica a los datos de la solicitud de escritura, y llevar a cabo operaciones de compresión y almacenamiento sobre los datos. El mensaje de respuesta correspondiente a la solicitud de escritura anterior puede ser una orden normal de escritura completada, y es necesario que el mismo lleve solamente una LBA correspondiente a un bloque de datos.

Si el controlador segmenta los datos de acuerdo con una granularidad mayor en la etapa 1204, es necesario que el mensaje de respuesta en la etapa 1218 lleve LBAs correspondientes a una pluralidad de bloques de datos segmentados. En este caso, se puede definir una nueva orden de escritura completada. Es necesario que el volumen de datos de la nueva orden de escritura completada sea mayor que el correspondiente a una orden normal de escritura completada. Por ejemplo, si una orden normal de escritura completada es 16 bytes, una orden nueva definida de escritura completada puede ser 32/64/128 bytes, para que contenga más LBAs. Debe indicarse que si es necesario traer una pluralidad de LBAs con la orden de escritura completada, la secuencia de la pluralidad de LBAs en la orden de escritura completada se puede mantener congruente con la secuencia de bloques de datos correspondiente a la pluralidad de LBAs. De esta manera, no es necesario consumir ningún byte adicional para indicar LBAs de datos que se corresponden con la pluralidad de LBAs respectivamente.

1220. El controlador establece una relación de mapeo entre una LBA de la matriz de discos y una LBA del SSD.

El controlador recibe la solicitud de escritura enviada por el anfitrión en la etapa 1202, y la solicitud de escritura lleva el LUN_ID y la LBA de la matriz de discos. La orden de escritura completada recibida por el controlador en la etapa 1218 incluye la LBA del SSD. El controlador establece la relación de mapeo entre la LBA de la matriz de discos y la LBA del SSD.

Por ejemplo, el controlador de almacenamiento escribe dos bloques de datos de 4 KB consecutivamente en un dispositivo de LUN cuyo LUN_ID es 0, a partir de una posición de inicio LBA=0 de acuerdo con la solicitud de escritura recibida en la etapa 1202. Puesto que la LBA y la LEN definidas en un sistema lo son generalmente según cada sector individual, y el tamaño de un sector es en general 0,5 KB, la LEN de un bloque de datos de 4 KB es 8. De esta manera, la dirección lógica del primer bloque de datos en la matriz de discos se puede indicar con (LUN_ID=0, LBA=0, LEN=8); y la dirección lógica del segundo bloque de datos en la matriz de discos se puede indicar con (LUN_ID=0, LBA=8, LEN=8). Suponiendo que el primer bloque de datos se escribe en el segundo bloque lógico en un SSD cuyo ID es 2, y el segundo bloque de datos se escribe en el tercer bloque lógico en un SSD cuyo ID es 3, la dirección lógica del primer bloque de datos en el SSD se puede indicar con (SSD_ID=2, LBA=8, LEN=8), y la dirección lógica del segundo bloque de datos en el SSD se puede indicar con (SSD_ID=3, LBA=16, LEN=8). En este caso, el controlador puede establecer una relación de mapeo que se muestra en la siguiente tabla:

LUN_ID, LBA, LEN	SSD (SSD_ID, LBA, LEN)
0, 0, 8	2, 8, 8
0, 8, 8	3, 16, 8

El controlador puede grabar la relación de mapeo entre la LAB de la matriz de discos y la LBA del SSD para facilitar una lectura posterior de los datos. Específicamente, cuando es necesario que el anfitrión lea datos que se han escrito previamente en el SSD, el controlador convierte una solicitud de lectura del anfitrión en una solicitud de lectura para el SSD sobre la base de la tabla de relaciones de mapeo grabadas, y devuelve datos correspondientes al anfitrión. Por ejemplo, es necesario que el anfitrión lea los segundos datos de 4 KB cuyo LUN_ID es 0, es decir, la segunda fila de la tabla anterior. Basándose en la relación de mapeo mostrada en la tabla anterior, el controlador halla la ubicación de una LBA 16 de los datos almacenados en el SSD cuyo ID es 3. Seguidamente, el controlador puede enviar una solicitud de lectura al SSD cuyo ID es 3, leer los datos en la ubicación de la LBA 16, y devolver los datos leídos al anfitrión.

1222. El controlador envía un mensaje de éxito de ejecución de la orden de solicitud de escritura al anfitrión.

La FIG. 13 es un diagrama de flujo esquemático de una modalidad de implementación de la etapa 1214 en compresión de longitud fija. Debe entenderse que las etapas u operaciones mostradas en la FIG. 13 son solamente ejemplos. En esta realización de la presente invención, también pueden realizarse otras operaciones o variantes de diversas operaciones de la FIG. 13. Adicionalmente, las etapas de la FIG. 13 se pueden realizar de acuerdo con una secuencia diferente a la presentada en la FIG. 13, y posiblemente no es necesario llevar a cabo todas las operaciones de la FIG. 13.

1302. Compresión de un bloque de datos y realización de una operación de alineamiento hacia arriba.

Por ejemplo, la granularidad de compresión de un bloque de datos es 4 KB. Una compresión de longitud fija requiere comprimir datos a un múltiplo entero de 0,5 KB. Después de que se haya comprimido el bloque de datos, si los datos comprimidos están entre 1,5 KB y 2 KB, se lleva a cabo una operación de alineamiento hacia arriba para obtener un bloque comprimido de 2 KB.

1304. Combinación de bloques comprimidos en una página física.

La manera de redondear bloques comprimidos se ha descrito anteriormente de forma detallada. Consúltese la FIG. 8 para obtener detalles, los cuales no se describen de nuevo en la presente. Debido a la compresión de longitud fija, una página física se puede llenar con una pluralidad de bloques comprimidos, y no se genera ningún fragmento.

1306. Escritura de bloques comprimidos redondeados en un soporte de un SSD.

Por ejemplo, después de que los bloques comprimidos se hayan redondeado en una página física completa, los bloques comprimidos se escriben en el soporte del SSD. Después de una escritura con éxito, se actualiza la información de mapeo de los bloques comprimidos en una tabla de mapeo, y, en dicha tabla de mapeo, se graba una dirección física correspondiente a una dirección lógica. Debe señalarse que esta realización de la presente invención no se limita a lo mencionado, y que todos los bloques comprimidos procesados también se pueden escribir en el soporte del SSD a la vez.

1308. Cálculo del cambio de la capacidad lógica.

Por ejemplo, puede calcularse el cambio de la capacidad lógica cada vez que se ha comprimido un bloque de datos; o puede calcularse el cambio de la capacidad lógica después de que se hayan comprimido todos los datos. Esto no presenta limitaciones específicas en esta realización de la presente invención. Anteriormente se ha descrito de forma detallada una manera específica de calcular el cambio de la capacidad lógica, y la misma no se describe de nuevo en la presente.

La FIG. 14 es un diagrama de flujo esquemático de una modalidad de implementación de la etapa 1214 en compresión de longitud variable. Debe entenderse que las etapas u operaciones mostradas en la FIG. 14 son solamente ejemplos. En esta realización de la presente invención también pueden realizarse otras operaciones o variantes de diversas operaciones de la FIG. 14. Adicionalmente, las etapas de la FIG. 14 se pueden llevar a cabo de acuerdo con una secuencia diferente a la presentada en la FIG. 14, y posiblemente no es necesario llevar a cabo todas las operaciones de la FIG. 14.

1402. Compresión de un bloque de datos para obtener un bloque comprimido.

Debido a la compresión de longitud variable, no es necesario llevar a cabo una operación de alineamiento hacia arriba sobre los datos en este caso, y el tamaño del bloque comprimido se mantiene igual.

1404. Combinación de bloques comprimidos en una página física.

Por ejemplo, los bloques comprimidos de longitud variable se pueden redondear de principio a fin en una página física. El área restante final que ya no puede contener un bloque comprimido completo puede servir como fragmento, y no almacena datos válidos.

1406. Escritura de bloques comprimidos redondeados en un soporte de un SSD según la modalidad de la Solución 1.

Solución 1: la información de mapeo en una tabla de mapeo incluye información sobre una página física que almacena un bloque comprimido. Un encabezamiento de la página física almacena metadatos, y los metadatos indican una dirección de inicio del bloque comprimido en la página física y una longitud del bloque comprimido. Cuando va a leerse el bloque comprimido, en primer lugar puede hallarse la página física correspondiente de acuerdo con la información de mapeo almacenada en la tabla de mapeo; a continuación se leen en conjunto los datos en la página física, y, en la información de encabezamiento de la página física, se busca la dirección de inicio y la longitud del bloque comprimido en la página física. Específicamente, la información de encabezamiento de la página física puede almacenar metadatos de una pluralidad de bloques comprimidos. Para ayudar a la búsqueda de metadatos de un bloque comprimido, en la información de encabezamiento de la página física se puede establecer una correspondencia entre una LBA del bloque de datos de 4 KB y metadatos. En este caso, el SSD puede buscar, de acuerdo con la LBA del bloque de datos, los metadatos correspondientes al bloque comprimido.

1408. Escritura de bloques comprimidos redondeados en un soporte de un SSD según la modalidad de la Solución 2.

Solución 2: de manera similar a la Solución 1, en la Solución 2, la información sobre una página física de un bloque comprimido se almacena en una tabla de mapeo, y, en la página física, se almacenan metadatos del bloque comprimido. Una de las diferencias reside en que la tabla de mapeo almacena no solamente la información sobre la página física del bloque comprimido, sino también información de indexación, en donde la información de indexación puede describir una posición del bloque comprimido en una secuencia de todos los bloques comprimidos almacenados

en la página física. De esta manera, cuando se buscan los metadatos de la página física, no es necesario que el SSD ubique los metadatos del bloque comprimido buscando en una LBA compleja como se hace en la Solución 1, sino que puede ubicar los metadatos de acuerdo con la información de indexación, reduciéndose así el volumen de datos de los metadatos y ahorrándose espacio de almacenamiento de la página física.

5 1410. Escritura de bloques comprimidos redondeados en un soporte de un SSD según la modalidad de la Solución 3.

Solución 3: la información de mapeo en una tabla de mapeo incluye información sobre una página física que almacena un bloque comprimido, una dirección de inicio del bloque comprimido en la página física, y la longitud del bloque comprimido. En esta solución, no es necesario almacenar metadatos en la página física, de manera que se simplifica la modalidad de implementación de la página física; adicionalmente, se omite un proceso de búsqueda de metadatos de acuerdo con la información de mapeo, mejorándose así la eficiencia de direccionamiento de una dirección física.

10

Debe entenderse que, en la práctica, es necesario llevar a cabo solamente una de la etapa 1406 a la etapa 1410.

1412. Cálculo del cambio de la capacidad lógica.

Por ejemplo, el cambio de la capacidad lógica se puede calcular cada vez que se ha comprimido un bloque de datos; o el cambio de la capacidad lógica se puede calcular después de que se hayan comprimido todos los datos. Esto no presenta limitaciones específicas en esta realización de la presente invención. Anteriormente se ha descrito de forma detallada una manera específica de calcular el cambio de la capacidad lógica, y la misma no se describe nuevamente en la presente.

15

Lo anterior describe transiciones de un bloque lógico entre tres estados de forma detallada y en referencia a la FIG. 5, donde un bloque lógico aislado se genera activando una orden de reajuste/desmapeo. Lo siguiente describe de forma detallada procedimientos de procesamiento relacionados con reajustes/desmapeos en referencia a la FIG. 15.

20

La FIG. 15 es un diagrama de un ejemplo de un procedimiento de reajuste/desmapeo de acuerdo con una realización de la presente invención. Debe entenderse que las etapas u operaciones mostradas en la FIG. 15 son solamente ejemplos. En esta realización de la presente invención, también pueden realizarse otras operaciones o variantes de diversas operaciones de la FIG. 15. Adicionalmente, las etapas de la FIG. 15 se pueden llevar a cabo de acuerdo con una secuencia diferente a la presentada en la FIG. 15, y, posiblemente, no es necesario realizar todas las operaciones de la FIG. 15.

25

Para facilitar su comprensión, se describe en primer lugar, brevemente, un proceso de una operación de reajuste/desmapeo.

En general, cuando es necesario que un anfitrión elimine algunos datos de un disco, un controlador envía una orden de reajuste/desmapeo a un SSD, y en primer lugar marca los datos como datos no válidos. Debido a las características del SSD, un bloque físico ocupado con los datos no válidos temporalmente no se puede escribir con datos nuevos (es decir, no se puede sobrescribir), y puede continuar almacenando datos nuevos únicamente después de que el bloque físico en el cual están ubicados los datos no válidos se borre usando un mecanismo de recolección de basura.

30

Cuando el controlador reajusta o desmapea un intervalo de direcciones lógicas, el bloque lógico correspondiente al intervalo de direcciones lógicas puede cambiar de un bloque lógico válido a un bloque lógico aislado. El intervalo de direcciones lógicas en una orden de reajuste/desmapeo se puede seleccionar en concordancia con un múltiplo de un bloque lógico, para facilitar la operación de reajuste/desmapeo del SSD.

35

El Reajuste/Desmapeo viene acompañado de un decremento de la capacidad lógica del SSD y de un incremento de la capacidad lógica disponible del SSD. Específicamente, el intervalo de direcciones lógicas al cual va dirigida la orden de reajuste/desmapeo se puede marcar como no válido, y, en este caso, es necesario quitar de la capacidad lógica del SSD una capacidad correspondiente al intervalo de direcciones lógicas. Adicionalmente, el espacio físico correspondiente al intervalo de direcciones lógicas no se puede utilizar inmediatamente, ya que la operación de reajuste/desmapeo únicamente marca datos en el espacio físico como no válidos, y es necesario liberar el espacio físico usando el mecanismo de recolección de basura. No obstante, debido a que existe un espacio de sobredimensionamiento, el incremento de la capacidad lógica correspondiente al espacio físico se puede reflejar inmediatamente. Posteriormente se describen detalles en referencia a las etapas de la FIG. 15.

40

45

1502. Un control envía una orden de reajuste/desmapeo a un SSD.

En general, el controlador recibe en primer lugar una orden de reajuste/desmapeo enviada por un anfitrión, donde la orden se usa para dar instrucciones con el fin de marcar un segmento de datos como datos no válidos. El controlador halla un intervalo de direcciones lógicas del segmento de datos en el SSD, a continuación envía la orden de reajuste/desmapeo al SSD, y da instrucciones al SSD para que marque datos del intervalo de direcciones lógicas como datos no válidos o cancele la relación de mapeo entre una dirección lógica y una dirección física de los datos del intervalo de datos. La orden de reajuste/desmapeo enviada por el controlador al SSD puede incluir una dirección lógica de datos que deben ser reajustados o desmapeados, por ejemplo, la LBA+LEN de los datos a reajustar o desmapear en el SSD.

50

55

1504. El SSD lleva a cabo una operación de reajuste/desmapeo sobre datos en un intervalo de direcciones lógicas especificado.

5 Específicamente, el SSD puede seleccionar un bloque lógico en el intervalo de direcciones lógicas, a continuación extraer información de mapeo de una entrada que está en una tabla de mapeo y se corresponde con el bloque lógico, hallar una dirección física del bloque lógico en el bloque físico, y, a continuación, marcar como no válidos datos en la dirección física.

10 En el bloque físico se pueden mantener dos valores estadísticos, en donde uno de los valores puede indicar la cantidad total de bloques de datos almacenados en el bloque físico, y el otro valor puede indicar la cantidad de bloques de datos válidos almacenados en el bloque físico. El SSD, en función de la cantidad de los bloques de datos válidos en el bloque físico o el porcentaje de los bloques de datos válidos, puede determinar si llevar a cabo una recolección de basura. Cuando se escriben datos en el bloque físico, la cantidad de los bloques de datos válidos correspondientes al bloque físico aumenta. Cada vez que se reajusta o desmapea un bloque de datos válido en el bloque físico, la cantidad de los bloques de datos válidos en el bloque físico se decrementa en 1. La operación de reajuste/desmapeo se lleva a cabo cíclicamente según la manera anterior hasta que se procesan todos los bloques lógicos del intervalo de direcciones lógicas.

1506. El SSD calcula un decremento de la capacidad lógica provocado por la orden de reajuste/desmapeo.

Después de que se cancelen todas las relaciones de mapeo de todos los bloques lógicos en un intervalo de direcciones en correspondencia con la orden de reajuste/desmapeo, puede calcularse y ajustarse la capacidad lógica del SSD. Para obtener una modalidad de cálculo específica, consúltese la siguiente fórmula:

20 Decremento de la capacidad lógica = Capacidad lógica de una dirección lógica en el intervalo de la operación de reajuste/desmapeo

1508. El SSD calcula el incremento de la capacidad lógica disponible provocado por la orden de reajuste/desmapeo.

25 Específicamente, se supone que los datos no válidos ocupan 80 KB de espacio físico del SSD. Aunque los 80 KB de espacio físico se pueden liberar únicamente después de una recolección de basura, primero se pueden tomar prestados 80 KB de espacio de sobredimensionamiento del espacio de sobredimensionamiento del SSD, y, posteriormente, después de que los 80 KB de espacio físico ocupados por los datos no válidos se liberen usando un mecanismo de recolección de basura, los datos del espacio de sobredimensionamiento de 80 KB se mueven al espacio de almacenamiento correspondiente a la capacidad física.

30 Usando este mecanismo, una vez que la operación de reajuste/desmapeo genera datos no válidos, se puede tomar prestado el espacio de sobredimensionamiento para "liberar" el espacio físico ocupado por los datos no válidos. En este caso, la capacidad lógica disponible del SSD se puede incrementar en 80 KB de manera correspondiente. Si el tamaño de un bloque lógico es 8 KB, 10 bloques lógicos aislados se pueden convertir en bloques lógicos en blanco. Esto es equivalente a la adición de 10 bloques lógicos en blanco disponibles. Los bloques lógicos en blanco que se liberan se pueden usar para almacenar datos escritos nuevos.

35 Por ejemplo, suponiendo que el intervalo de direcciones lógicas indicado por la orden de reajuste/desmapeo recibida por el SSD es 100 K, debido al almacenamiento comprimido, el espacio físico ocupado realmente por datos en el intervalo de direcciones lógicas de 100 K es 80 K. En este caso, puede tomarse prestado un espacio de sobredimensionamiento de 80 K del espacio de sobredimensionamiento. El espacio de sobredimensionamiento de 80 K se puede usar para almacenar datos nuevos. En este ejemplo, el decremento de la capacidad lógica del SSD provocado por la orden de reajuste/desmapeo es 100 K. Usando el espacio de sobredimensionamiento, el incremento de la capacidad lógica disponible del SSD es 80 K.

45 Lo anterior describe de manera detallada un procedimiento de procesado de solicitudes de escritura en referencia a la FIG. 10. El procedimiento de procesado de solicitudes de lectura (o a las que se hace referencia IO de lectura) es el proceso inverso del procedimiento de procesado de solicitudes de escritura. Lo siguiente describe un procedimiento de procesado de solicitudes de lectura de acuerdo con una realización de la presente invención, de forma detallada y en referencia a la FIG. 16.

50 La FIG. 16 es un diagrama de flujo esquemático de procesado de una solicitud de lectura de acuerdo con una realización de la presente invención. Debe entenderse que las etapas u operaciones mostradas en la FIG. 16 son solamente ejemplos. En esta realización de la presente invención, también pueden realizarse otras operaciones o variantes de diversas operaciones de la FIG. 16. Adicionalmente, las etapas de la FIG. 16 se pueden llevar a cabo de acuerdo con una secuencia diferente a la presentada en la FIG. 16, y posiblemente no necesario realizar todas las operaciones de la FIG. 16.

1602. Un anfitrión envía una solicitud de lectura al controlador.

55 La solicitud de lectura enviada por el anfitrión puede incluir un LUN_ID, una LBA de datos destinados a leerse en una matriz de discos, y la longitud de los datos destinados a leerse.

1604. El controlador envía la solicitud de lectura a un SSD.

Específicamente, el controlador puede consultar, de acuerdo con el LUN_ID, la LBA de los datos destinados a leerse en la matriz de discos, y la longitud de los datos destinados a leerse que se reciben en la etapa 1602, una tabla de mapeo mantenida por el propio controlador (la tabla de mapeo se puede usar para indicar una relación de mapeo entre una LBA de la matriz de discos presentada al anfitrión y una LBA del SSD), y determinar un SSD en el que están ubicados los datos destinados a leerse, y la LBA y la longitud de los datos destinados a leerse en el SSD.

1606. El SSD lee datos comprimidos de un soporte del SSD sobre la base de una tabla de mapeo.

Suponiendo que el SSD usa una modalidad de compresión de longitud fija, y la tabla de mapeo almacena información sobre una página física que almacena los datos comprimidos, una dirección de inicio de los datos comprimidos en la página física, y la longitud de los datos comprimidos, el SSD puede ubicar directamente los datos comprimidos de acuerdo con la información almacenada en la tabla de mapeo.

Suponiendo que el SSD usa una modalidad de compresión de longitud variable, que la tabla de mapeo almacena solamente información sobre la página física que almacena datos comprimidos, y que la dirección de inicio y la longitud de los datos comprimidos en la página física están ubicadas en un encabezamiento de la página física, el SSD en primer lugar puede hallar la página física de acuerdo con la información sobre la página física, hallar la dirección de inicio y la longitud de los datos comprimidos en la página física a partir de la información de encabezamiento de la página física, y, a continuación, ubicar los datos comprimidos de acuerdo con la dirección de inicio y la longitud de los datos comprimidos en la página física.

1608. El SSD lleva a cabo una operación de descompresión sobre los datos comprimidos con el fin de obtener datos destinados a leerse.

El SSD puede descomprimir los datos comprimidos usando una unidad de descompresión interna, y puede restablecer los datos destinados a leerse.

1610. El SSD devuelve los datos destinados a leerse al controlador.

1612. El controlador devuelve al anfitrión los datos destinados a leerse.

Lo anterior describe de manera detallada un método de procesado de datos de acuerdo con una realización de la presente invención y en referencia a la FIG. 1 a la FIG. 16. Lo siguiente describe un SSD de acuerdo con una realización de la presente invención de forma detallada y en referencia a la FIG. 17 y la FIG. 18. Debe entenderse que el SSD de la FIG. 17 ó la FIG. 18 puede implementar cada etapa llevada a cabo por el SSD de la FIG. 1 a la FIG. 16. Para evitar repeticiones, en este caso no se describe nuevamente detalles.

Lo siguiente describe una realización de un aparato y una realización de un sistema de la presente invención. Puesto que la realización del aparato y la realización del sistema se pueden usar para ejecutar el método anterior, en relación con las partes que no se describen de forma detallada, consúltense las realizaciones de método anteriores.

La FIG. 17 es un diagrama estructural esquemático de un aparato de almacenamiento según una realización de la presente invención. Debe entenderse que la función o lógica del aparato 1700 de almacenamiento se puede implementar usando solamente *software*, o se puede implementar usando solamente *hardware*, o se puede implementar usando una combinación de *software* y *hardware*. Por ejemplo, el aparato 1700 de almacenamiento se puede implementar usando una combinación de *software* y *hardware*. En este caso, el módulo de recepción en la siguiente descripción se puede implementar usando un circuito receptor, y el módulo de compresión en la siguiente descripción se puede implementar usando código de ordenador. Como otro ejemplo, el aparato 1700 de almacenamiento se puede implementar usando solamente *hardware*, y la función o lógica de cada módulo en la siguiente descripción se puede implementar usando un circuito lógico. Como otro ejemplo, el aparato 1700 de almacenamiento se puede implementar usando solamente *hardware*, y cada módulo de la siguiente descripción puede ser un módulo funcional que se corresponde con código de ordenador.

El aparato 1700 de almacenamiento incluye:

un primer módulo de recepción 1710, configurado para recibir una solicitud de escritura de un controlador, en donde la solicitud de escritura es portadora de datos destinados a escribirse;

un módulo 1720 de compresión, configurado para comprimir los datos destinados a escribirse con el fin de obtener datos comprimidos;

un primer módulo 1730 de almacenamiento, configurado para almacenar los datos comprimidos; y

un primer módulo emisor 1740, configurado para enviar una primera información de retroalimentación al controlador, donde la primera información de retroalimentación indica la capacidad restante del aparato 1700 de almacenamiento después de que se almacenen los datos comprimidos.

- 5 Opcionalmente, en algunas realizaciones, el aparato 1700 de almacenamiento incluye, además: un segundo módulo emisor, configurado para enviar una segunda información de retroalimentación al controlador, donde la segunda información de retroalimentación indica la capacidad lógica del aparato 1700 de almacenamiento, y la capacidad lógica del aparato 1700 de almacenamiento es la suma del volumen de datos sin comprimir de datos almacenados en el aparato 1700 de almacenamiento y la capacidad de espacio físico libre del aparato 1700 de almacenamiento.
- 10 Opcionalmente, en algunas realizaciones, el aparato 1700 de almacenamiento incluye, además: un módulo de asignación, configurado para asignar un bloque lógico en blanco a los datos destinados a escribirse, donde cada bloque lógico se corresponde con un segmento de capacidad lógica del aparato 1700 de almacenamiento; un primer módulo de conversión, configurado para convertir el bloque lógico en blanco en un bloque lógico válido, donde el bloque lógico válido es un bloque lógico cuya dirección lógica está ocupada por datos válidos; y un primer módulo de consulta, configurado para consultar la cantidad de bloques lógicos en blanco restantes, en el aparato 1700 de almacenamiento; donde el primer módulo emisor 1740 está configurado específicamente para enviar la primera información de retroalimentación al controlador, donde la primera información de retroalimentación incluye la cantidad de los bloques lógicos en blanco restantes y/o la capacidad restante, y la capacidad restante se determina basándose en la cantidad de los bloques lógicos en blanco restantes.
- 15 Opcionalmente, en algunas realizaciones, el módulo 1720 de compresión está configurado, específicamente, para: después de que el módulo de asignación asigne el bloque lógico en blanco a los datos destinados a escribirse, comprimir los datos destinados a escribirse con el fin de obtener los datos comprimidos; y el aparato 1700 de almacenamiento incluye, además: un primer módulo de determinación, configurado para determinar, de acuerdo con el tamaño de los datos destinados a escribirse y el tamaño de los datos comprimidos, la cantidad de bloques lógicos en blanco que es necesario añadir en el aparato 1700 de almacenamiento.
- 20 Opcionalmente, en algunas realizaciones, el aparato 1700 de almacenamiento incluye, además: un módulo de adición, configurado para añadir, de acuerdo con la cantidad de los bloques lógicos en blanco que es necesario añadir en el aparato 1700 de almacenamiento, entradas correspondientes a bloques lógicos en blanco nuevos a la tabla de mapeo.
- 25 Opcionalmente, en algunas realizaciones, el aparato 1700 de almacenamiento incluye, además: un módulo de actualización, configurado para actualizar un bloque lógico aislado en el aparato 1700 de almacenamiento a un bloque lógico en blanco de acuerdo con la cantidad de los bloques lógicos en blanco que es necesario añadir en el aparato 1700 de almacenamiento, donde el bloque lógico aislado es un bloque lógico cuya dirección lógica no se puede usar en el aparato 1700 de almacenamiento.
- 30 Opcionalmente, en algunas realizaciones, la cantidad M de bloques lógicos en blanco que es necesario añadir en el aparato 1700 de almacenamiento es igual a un resultado, redondeado a la baja, de dividir D por L, donde D indica la diferencia entre la longitud de los datos destinados a escribirse y la longitud de los datos comprimidos, y L indica la longitud del bloque lógico en el aparato 1700 de almacenamiento.
- 35 Opcionalmente, en algunas realizaciones, el aparato 1700 de almacenamiento incluye, además: un segundo módulo de recepción, configurado para recibir una orden del controlador, donde la orden incluye un intervalo de direcciones lógicas, y la orden se usa para dar instrucciones con el fin de marcar datos válidos en el intervalo de direcciones lógicas como datos no válidos; y un segundo módulo de conversión, configurado para convertir un bloque lógico válido del intervalo de direcciones lógicas en un bloque lógico aislado, donde el bloque lógico aislado es un bloque lógico cuya dirección lógica no se puede usar.
- 40 Opcionalmente, en algunas realizaciones, el aparato 1700 de almacenamiento incluye, además: un segundo módulo de determinación, configurado para determinar el tamaño del espacio físico ocupado por datos en el intervalo de direcciones lógicas; un módulo de selección, configurado para seleccionar un espacio de sobredimensionamiento parcial del espacio de sobredimensionamiento, donde el tamaño del espacio de sobredimensionamiento parcial es igual al tamaño del espacio físico ocupado por los datos en el intervalo de direcciones lógicas; y un tercer módulo de determinación, configurado para usar el espacio de sobredimensionamiento parcial como el espacio físico libre del aparato 1700 de almacenamiento.
- 45 Opcionalmente, en algunas realizaciones, el aparato 1700 de almacenamiento incluye, además: un segundo módulo de consulta, configurado para consultar el espacio físico libre del aparato 1700 de almacenamiento antes de que se envíe la primera información de retroalimentación al controlador; y un cuarto módulo de determinación, configurado para determinar la capacidad del espacio físico libre como capacidad restante.
- 50 Opcionalmente, en algunas realizaciones, el aparato 1700 de almacenamiento incluye, además: un quinto módulo de determinación, configurado para asignar una dirección lógica a los datos destinados a escribirse después de que se reciba la solicitud de escritura desde el controlador; y un primer módulo de grabación, configurado para grabar información de mapeo en una entrada que está en la tabla de mapeo y se corresponde con la dirección lógica, donde la información de mapeo incluye información sobre una página física que almacena los datos comprimidos, una posición de inicio de los datos comprimidos en la página física y la longitud de los datos comprimidos.
- 55 Opcionalmente, en algunas realizaciones, el aparato 1700 de almacenamiento incluye, además: un sexto módulo de determinación, configurado para asignar una dirección lógica a los datos destinados a escribirse después de que se

reciba la solicitud de escritura del controlador; un segundo módulo de grabación, configurado para grabar información de mapeo en una entrada que está en la tabla de mapeo y se corresponde con la dirección lógica, donde la información de mapeo incluye información sobre una página física que almacena los datos comprimidos; y un segundo módulo de almacenamiento, configurado para almacenar metadatos de los datos comprimidos en la página física, donde los metadatos incluyen una posición de inicio de los datos comprimidos en la página física y la longitud de los datos comprimidos.

Opcionalmente, en algunas realizaciones, la solicitud de escritura incluye la longitud de datos correspondiente a los datos destinados a escribirse, y la dirección lógica de los datos destinados a escribirse se asigna basándose en la longitud de los datos.

Opcionalmente, en algunas realizaciones, el aparato 1700 de almacenamiento incluye, además: un tercer módulo emisor, configurado para enviar la dirección lógica de los datos destinados a escribirse en el aparato 1700 de almacenamiento al controlador después de que se reciba la solicitud de escritura del controlador.

Opcionalmente, en algunas realizaciones, el aparato 1700 de almacenamiento incluye, además: un tercer módulo de recepción, configurado para: recibir una solicitud de lectura del controlador, donde la solicitud de lectura incluye la dirección lógica de datos destinados a leerse; un módulo de lectura, configurado para leer datos objetivo en una dirección física correspondiente a la dirección lógica de los datos destinados a leerse; un módulo de descompresión, configurado para descomprimir los datos objetivo con el fin de obtener los datos destinados a leerse; y un cuarto módulo emisor, configurado para enviar los datos destinados a leerse al controlador.

Debe señalarse que el aparato 1700 de almacenamiento puede ser, por ejemplo, un SSD. El aparato 1700 de almacenamiento se puede implementar por medio de un SSD que incluya una interfaz de comunicaciones, un soporte de almacenamiento, y un procesador. El procesador, ejecutando una instrucción de ordenador en una memoria, puede llevar a cabo operaciones descritas en las realizaciones del método.

El procesador puede llevar a cabo, además, usando una conexión lógica de *hardware* interno, operaciones descritas en las realizaciones del método, por ejemplo, puede implementar las operaciones usando un circuito integrado de aplicación específica (circuito integrado de aplicación específica, ASIC, en forma abreviada), o usando un dispositivo lógico programable (dispositivo lógico programable, PLD en forma abreviada). El PLD puede ser un dispositivo lógico programable complejo (en inglés: *complex programmable logic device*, CPLD en forma abreviada), una FPGA, un módulo lógico de matriz genérica (en inglés: *generic array logic*, GAL en forma abreviada), o cualquier combinación de los mismos. En esta realización de la presente invención, por ejemplo, el aparato de almacenamiento es un SSD. El SSD incluye un soporte de almacenamiento y un procesador. El soporte de almacenamiento está configurado para proporcionar espacio de almacenamiento. El procesador del SSD incluye: un primer circuito receptor, configurado para recibir una solicitud de escritura de un controlador, donde la solicitud de escritura es portadora de datos destinados a escribirse; un circuito de compresión, configurado para comprimir los datos destinados a escribirse con el fin de obtener datos comprimidos; un primer circuito de almacenamiento, configurado para almacenar los datos comprimidos; y un primer circuito emisor, configurado para enviar una primera información de retroalimentación al controlador, donde la primera información de retroalimentación indica la capacidad restante del SSD después de que se almacenen los datos comprimidos.

La FIG. 18 es un diagrama estructural esquemático de un sistema de almacenamiento según una realización de la presente invención. El sistema de almacenamiento 1800 de la FIG. 18 incluye un controlador 1810 y el aparato 1700 de almacenamiento descrito en la FIG. 17.

El controlador 1810 está configurado para: generar una solicitud de escritura que incluye los datos destinados a escribirse; recibir un mensaje de respuesta para la solicitud de escritura del aparato 1700 de almacenamiento, donde el mensaje de respuesta incluye una dirección lógica de los datos destinados a escribirse en el aparato 1700 de almacenamiento; y grabar una relación de mapeo entre la dirección lógica de los datos destinados a escribirse en una matriz de discos y la dirección lógica de los datos destinados a escribirse en el aparato 1700 de almacenamiento.

Opcionalmente, en una realización, el controlador 1810 está configurado, además, para: recibir datos destinados a escribirse nuevos de un anfitrión; determinar la capacidad restante del aparato 1700 de almacenamiento de acuerdo con la primera información de retroalimentación; y cuando la capacidad restante del aparato 1700 de almacenamiento es mayor que el tamaño de los datos destinados a escribirse nuevos, escribir los datos destinados a escribirse nuevos en el aparato 1700 de almacenamiento.

La FIG. 19 es un diagrama estructural esquemático de un disco de estado sólido de acuerdo con una realización de la presente invención. El disco 1900 de estado sólido de la FIG. 19 incluye:

- una interfaz 1910 de comunicaciones, configurada para comunicarse con un controlador;
- un soporte 1920 de almacenamiento, configurado para proporcionar espacio de almacenamiento; y
- un procesador 1930, conectado a la interfaz 1910 de comunicaciones y al soporte 1920 de almacenamiento, y configurado para: recibir una solicitud de escritura del controlador a través de la interfaz 1910 de comunicaciones,

5 donde la solicitud de escritura es portadora de datos destinados a escribirse; comprimir los datos destinados a escribirse con el fin de obtener datos comprimidos; almacenar los datos comprimidos en el soporte 1920 de almacenamiento; y enviar una primera información de retroalimentación al controlador a través de la interfaz 1910 de comunicaciones, donde la primera información de retroalimentación indica la capacidad restante del soporte 1920 de almacenamiento después de que se almacenen los datos comprimidos.

10 Opcionalmente, en algunas realizaciones, el procesador 1930 está configurado, además, para enviar una segunda información de retroalimentación al controlador a través de la interfaz 1910 de comunicaciones, donde la segunda información de retroalimentación indica la capacidad lógica del soporte 1920 de almacenamiento, y la capacidad lógica del soporte 1920 de almacenamiento es la suma del volumen de datos sin comprimir de datos almacenados en el soporte 1920 de almacenamiento y la capacidad de espacio físico libre del soporte 1920 de almacenamiento.

15 Opcionalmente, en algunas realizaciones, el procesador 1930 está configurado, además, para: asignar un bloque lógico en blanco a los datos destinados a escribirse, donde el bloque lógico en blanco es un bloque lógico con una dirección lógica libre, y cada bloque lógico se corresponde con un segmento de capacidad lógica del soporte 1920 de almacenamiento; convertir el bloque lógico en blanco asignado en un bloque lógico válido, donde el bloque lógico válido es un bloque lógico cuya dirección lógica está ocupada por datos válidos; y consultar la cantidad de bloques lógicos en blanco restantes en el disco 1900 de estado sólido; y el procesador 1930 está configurado, específicamente, para enviar la primera información de retroalimentación al controlador a través de la interfaz 1910 de comunicaciones, donde la primera información de retroalimentación incluye la cantidad de los bloques lógicos en blanco restantes y/o la capacidad restante, y la capacidad restante se determina basándose en la cantidad de los bloques lógicos en blanco restantes.

20

Opcionalmente, en algunas realizaciones, el procesador 1930 está configurado, específicamente, para: después de que el procesador 1930 asigne el bloque lógico en blanco a los datos destinados a escribirse, comprimir los datos destinados a escribirse con el fin de obtener los datos comprimidos; y el procesador 1930 está configurado, además, para determinar, de acuerdo con el tamaño de los datos destinados a escribirse y el tamaño de los datos comprimidos, la cantidad de bloques lógicos en blanco que es necesario añadir en el disco 1900 de estado sólido.

25

Opcionalmente, en algunas realizaciones, el procesador 1930 está configurado, además, para añadir, de acuerdo con la cantidad de los bloques lógicos en blanco que es necesario añadir en el disco 1900 de estado sólido, entradas correspondientes a bloques lógicos en blanco nuevos a una tabla de mapeo del disco 1900 de estado sólido.

30 Opcionalmente, en algunas realizaciones, el procesador 1930 está configurado, además, para actualizar un bloque lógico aislado en el disco 1900 de estado sólido a un bloque lógico en blanco de acuerdo con la cantidad de los bloques lógicos en blancos que es necesario añadir en el disco 1900 de estado sólido, donde el bloque lógico aislado es un bloque lógico cuya dirección lógica no se puede usar en el disco 1900 de estado sólido.

Opcionalmente, en algunas realizaciones, la cantidad M de bloques lógicos en blanco que es necesario añadir en el disco 1900 de estado sólido es igual al resultado, redondeado a la baja, de dividir D por L, donde D indica la diferencia entre la longitud de los datos destinados a escribirse y la longitud de los datos comprimidos, y L indica la longitud de un bloque lógico en el disco 1900 de estado sólido.

35

Opcionalmente, en algunas realizaciones, el procesador 1930 está configurado, además, para: recibir una orden del controlador a través de la interfaz 1910 de comunicaciones, donde la orden incluye un intervalo de direcciones lógicas, y la orden se usa para dar instrucciones con el fin de marcar datos válidos en el intervalo de direcciones lógicas como datos no válidos; y convertir un bloque lógico válido del intervalo de direcciones lógicas en un bloque lógico aislado, donde el bloque lógico aislado es un bloque lógico cuya dirección lógica no se puede usar.

40

Opcionalmente, en algunas realizaciones, el procesador 1930 está configurado, además, para: determinar el tamaño del espacio físico ocupado por datos en el intervalo de direcciones lógicas; seleccionar un espacio de sobredimensionamiento parcial del espacio de sobredimensionamiento, donde el tamaño del espacio de sobredimensionamiento parcial es igual al tamaño del espacio físico ocupado por los datos en el intervalo de direcciones lógicas; y usar el espacio de sobredimensionamiento parcial como espacio físico libre del soporte 1920 de almacenamiento.

45

Opcionalmente, en algunas realizaciones, antes de que el procesador 1930 envíe la primera información de retroalimentación al controlador a través de la interfaz 1910 de comunicaciones, el procesador 1930 está configurado, además, para: consultar el espacio físico libre del soporte 1920 de almacenamiento; y determinar la capacidad del espacio físico libre como capacidad restante.

50

Opcionalmente, en algunas realizaciones, después de que el procesador 1930 reciba la solicitud de escritura del controlador a través de la interfaz 1910 de comunicaciones, el procesador 1930 está configurado, además, para: asignar una dirección lógica a los datos destinados a escribirse; y grabar información de mapeo en una entrada que está en la tabla de mapeo y se corresponde con la dirección lógica, donde la información de mapeo incluye información sobre una página física que almacena los datos comprimidos, una posición de inicio de los datos comprimidos en la página física, y la longitud de los datos comprimidos.

55

5 Opcionalmente, en algunas realizaciones, después de que el procesador 1930 reciba la solicitud de escritura del controlador a través de la interfaz 1910 de comunicaciones, el procesador 1930 está configurado, además, para: asignar una dirección lógica a los datos destinados a escribirse; grabar información de mapeo en una entrada que está en la tabla de mapeo y se corresponde con la dirección lógica, donde la información de mapeo incluye información sobre una página física que almacena los datos comprimidos; y almacenar metadatos de los datos comprimidos en la página física, donde los metadatos incluyen una posición de inicio de los datos comprimidos en la página física y la longitud de los datos comprimidos.

10 Opcionalmente, en algunas realizaciones, la solicitud de escritura incluye la longitud de datos correspondiente a los datos destinados a escribirse, y la dirección lógica de los datos destinados a escribirse se asigna basándose en la longitud de los datos.

Opcionalmente, en algunas realizaciones, después de que el procesador 1930 almacene los datos comprimidos, el procesador 1930 está configurado, además, para enviar la dirección lógica de los datos destinados a escribirse en el soporte 1920 de almacenamiento al controlador a través de la interfaz 1910 de comunicaciones.

15 Opcionalmente, en algunas realizaciones, el procesador 1930 está configurado, además, para: recibir una solicitud de lectura del controlador a través de la interfaz 1910 de comunicaciones, donde la solicitud de lectura incluye la dirección lógica de datos destinados a leerse; leer datos objetivo en una dirección física correspondiente a la dirección lógica de los datos destinados a leerse; descomprimir los datos objetivo con el fin de obtener los datos destinados a leerse; y enviar los datos destinados a leerse al controlador a través de la interfaz 1910 de comunicaciones.

20 La FIG. 20 es un diagrama estructural esquemático de un sistema de almacenamiento según una realización de la presente invención. El sistema de almacenamiento 2000 de la FIG. 20 incluye un controlador 2010 y el disco 1900 de estado sólido de la FIG. 19.

25 El controlador 2010 está configurado para: generar una solicitud de escritura que incluye los datos destinados a escribirse; recibir un mensaje de respuesta para la solicitud de escritura del procesador 1930 a través de la interfaz 1910 de comunicaciones, donde el mensaje de respuesta incluye una dirección lógica de los datos destinados a escribirse en el soporte 1920 de almacenamiento; y grabar una relación de mapeo entre la dirección lógica de los datos destinados a escribirse en una matriz de discos y la dirección lógica de los datos destinados a escribirse en el soporte 1920 de almacenamiento.

30 Opcionalmente, en algunas realizaciones, el controlador 2010 está configurado, además, para: recibir datos destinados a escribirse nuevos de un anfitrión; determinar la capacidad restante del disco 1900 de estado sólido de acuerdo con la primera información de retroalimentación; y, cuando la capacidad restante del disco 1900 de estado sólido, es mayor que el tamaño de los datos destinados a escribirse nuevos, escribir los datos destinados a escribirse nuevos en el disco 1900 de estado sólido.

35 La expresión "y/o" de esta memoria descriptiva representa que pueden existir tres relaciones. Por ejemplo, A y/o B puede representar los siguientes tres casos: solamente existe A, existen tanto A como B, y solamente existe B. Además, el carácter "/" en esta memoria descriptiva indica en general una relación de "o" entre los objetos asociados.

40 Una persona con conocimientos habituales en la materia puede percibir que las unidades y las etapas de algoritmos de los ejemplos descritos en referencia a las realizaciones dadas a conocer en esta memoria descriptiva, se pueden implementar por medio de *hardware* electrónico o una combinación de *software* de ordenador y *hardware* electrónico. El hecho de que las funciones se lleven a cabo por *hardware* o *software* depende de las aplicaciones particulares y de las condiciones limitativas del diseño de las soluciones técnicas. Una persona versada en la materia puede usar diferentes métodos para implementar las funciones descritas para cada aplicación particular, pero no debe considerarse que la implementación va más allá del alcance de la presente invención.

45 Una persona versada en la materia puede entender claramente que, por motivos de comodidad y brevedad en la descripción, con vistas a un proceso de funcionamiento detallado del sistema, del aparato y la unidad anteriores, puede hacerse referencia a un proceso correspondiente en las realizaciones de método anteriores, y no se describen detalles.

50 En las diversas realizaciones proporcionadas en esta invención, debe entenderse que el sistema, el aparato y el método dados a conocer se pueden implementar de otras maneras. Por ejemplo, la realización descrita del aparato es meramente un ejemplo. Por ejemplo, la división en unidad es meramente una división de funciones lógicas y puede ser otra división en una implementación real. Por ejemplo, una pluralidad de unidades o componentes puede estar incluida o integrada en otro sistema, o algunas de las características pueden ignorarse o no llevarse a cabo. Adicionalmente, los acoplamientos mutuos o acoplamientos directos o conexiones de comunicación visualizados o descritos se pueden implementar a través de algunas interfaces, acoplamientos indirectos o conexiones de comunicación entre los aparatos o unidades, o conexiones eléctricas, conexiones mecánicas o conexiones en otros formatos.

55 Las unidades descritas como partes independientes pueden ser o no independientes físicamente, y las partes visualizadas como unidades pueden ser o no unidades físicas, pueden estar ubicadas en una posición o pueden estar distribuidas en una pluralidad de unidades de red. Puede seleccionarse parte o la totalidad de las unidades de acuerdo

con las necesidades reales para alcanzar los objetivos de las soluciones de las realizaciones.

Además, unidades funcionales de las realizaciones de la presente invención pueden estar integradas en una unidad de procesado, o cada una de las unidades puede existir sola físicamente, o dos o más unidades están integradas en una unidad.

- 5 Cuando las funciones se implementan en forma de una unidad funcional de *software* y se comercializan o usan como un producto independiente, las funciones se pueden almacenar en un soporte de almacenamiento legible por ordenador. Sobre la base de dicha interpretación, las soluciones técnicas de la presente invención esencialmente, o la parte que contribuye a la técnica anterior, o algunas de las soluciones técnicas se pueden implementar en una forma de producto de *software*. El producto de *software* se almacena en un soporte de almacenamiento, e incluye diversas instrucciones para ordenar a un dispositivo de ordenador (el cual puede ser un ordenador personal, un servidor o un dispositivo de red) que lleve a cabo la totalidad o parte de las etapas de los métodos descritos en las realizaciones de la presente invención. El anterior soporte de almacenamiento incluye: cualquier soporte que pueda almacenar código de programa, tal como una unidad de almacenamiento *flash* USB, un disco duro extraíble, una memoria de solo lectura (ROM, Memoria de Solo Lectura), una memoria de acceso aleatorio (RAM, Memoria de Acceso Aleatorio), un disco magnético o un disco óptico.
- 10
- 15

La presente invención queda definida por las reivindicaciones independientes. En las reivindicaciones dependientes se definen aspectos preferidos.

REIVINDICACIONES

- 5 1. Un método de procesamiento de datos, aplicado a un sistema de almacenamiento, en donde el sistema de almacenamiento comprende un anfitrión, un controlador y un disco de estado sólido, SSD, el controlador está ubicado entre el anfitrión y el SSD, y como puente implementa un intercambio de datos entre el anfitrión y el SSD, y el método comprende:
- recibir, por parte del SSD, una solicitud de escritura del controlador, en donde la solicitud de escritura es portadora de datos destinados a escribirse;
 comprimir, por parte del SSD, los datos destinados a escribirse con el fin de obtener datos comprimidos;
 10 almacenar, por parte del SSD, los datos comprimidos; y
 enviar, por parte del SSD, una primera información de retroalimentación al controlador, en donde la primera información de retroalimentación indica la capacidad restante del SSD después de que se almacenen los datos comprimidos.
2. El método según la reivindicación 1, en donde el método comprende, además:
- 15 enviar, por parte del SSD, una segunda información de retroalimentación al controlador, en donde la segunda información de retroalimentación indica una capacidad lógica del SSD, y la capacidad lógica del SSD es la suma del volumen de datos sin comprimir de datos almacenados en el SSD y una capacidad de espacio físico libre del SSD.
3. Método según la reivindicación 1 ó 2, en donde, después de recibir, por parte del SSD, una solicitud de escritura del controlador, y antes de comprimir, por parte del SSD, los datos destinados a escribirse con el fin de obtener datos comprimidos, el método comprende, además:
- 20 asignar, por parte del SSD, un bloque lógico en blanco a los datos destinados a escribirse, en donde el bloque lógico en blanco es un bloque lógico con una dirección lógica libre, y cada bloque lógico se corresponde con un segmento de capacidad lógica del SSD;
 convertir, por parte del SSD, el bloque lógico en blanco asignado en un bloque lógico válido, en donde el bloque lógico válido es un bloque lógico cuya dirección lógica está ocupada por datos válidos; y
 25 consultar, por parte del SSD, una cantidad de bloques lógicos en blanco restantes en el SSD; y
 el envío, por parte del SSD, de la primera información de retroalimentación al controlador, comprende:
 enviar, por parte del SSD, la primera información de retroalimentación al controlador, en donde la primera información de retroalimentación comprende la cantidad de los bloques lógicos en blanco restantes o la capacidad restante, y la capacidad restante se determina sobre la base de la cantidad de los bloques lógicos en blanco restantes.
- 30 4. Método según la reivindicación 3, en donde la compresión, por parte del SSD, de los datos destinados a escribirse con el fin de obtener datos comprimidos, comprende:
- después de la asignación, por parte del SSD, de un bloque lógico en blanco a los datos destinados a escribirse,
 35 comprimir, por parte del SSD, los datos destinados a escribirse para obtener los datos comprimidos; y
 el método comprende, además:
 determinar, por parte del SSD de acuerdo con el tamaño de los datos destinados a escribirse y el tamaño de los datos comprimidos, una cantidad de bloques lógicos en blanco que es necesario añadir en el SSD.
5. El método según la reivindicación 4, en donde el método comprende, además:
- 40 añadir, por parte del SSD de acuerdo con la cantidad de los bloques lógicos en blanco que es necesario añadir en el SSD, entradas correspondientes a bloques lógicos en blanco nuevos a una tabla de mapeo del SSD.
6. El método según la reivindicación 4 ó 5, en donde el método comprende, además:
- 45 actualizar, por parte del SSD, un bloque lógico aislado en el SSD a un bloque lógico en blanco en función de la cantidad de los bloques lógicos en blanco que es necesario añadir en el SSD, en donde el bloque lógico aislado es un bloque lógico cuya dirección lógica no se puede usar en el SSD.
7. El método según una cualquiera de las reivindicaciones 4 a 6, en donde la cantidad M de bloques lógicos en blanco que es necesario añadir en el SSD es igual a un resultado, redondeado a la baja, de dividir D por L, en donde D indica una diferencia entre una longitud de los datos destinados a escribirse y una longitud de los datos comprimidos, y L indica una longitud de un bloque lógico en el SSD.
- 50 8. El método según una cualquiera de las reivindicaciones 3 a 7, en donde el método comprende, además:
- recibir, por parte del SSD, una orden del controlador, en donde la orden comprende un intervalo de direcciones lógicas, y la orden se usa para dar instrucciones con el fin de marcar datos válidos en el intervalo de direcciones lógicas como datos no válidos; y
 convertir, por parte del SSD, un bloque lógico válido del intervalo de direcciones lógicas en un bloque lógico aislado,

en donde el bloque lógico aislado es un bloque lógico cuya dirección lógica no se puede usar.

9. El método según la reivindicación 8, en donde el método comprende además:

determinar, por parte del SSD, el tamaño del espacio físico ocupado por datos en el intervalo de direcciones lógicas;

5 seleccionar, por parte del SSD, un espacio de sobredimensionamiento parcial del espacio de sobredimensionamiento, en donde un tamaño del espacio de sobredimensionamiento parcial es igual al tamaño del espacio físico ocupado por los datos en el intervalo de direcciones lógicas; y
usar, por parte del SSD, el espacio de sobredimensionamiento parcial como espacio físico libre del SSD.

10. El método según la reivindicación 1 ó 2, en donde antes del envío, por parte del SSD, de la primera información de retroalimentación al controlador, el método comprende, además:

consultar, por parte del SSD, el espacio físico libre del SSD; y
determinar, por parte del SSD, la capacidad del espacio físico libre como capacidad restante.

11. El método según una cualquiera de las reivindicaciones 1 a 10, en donde después de la recepción, por parte del SSD, de una solicitud de escritura de un controlador, el método comprende, además:

15 asignar, por parte del SSD, una dirección lógica a los datos destinados a escribirse; y
después del almacenamiento, por parte del SSD, de los datos comprimidos, el método comprende, además:
grabar, por parte del SSD, información de mapeo en una entrada que se encuentra en la tabla de mapeo y se
corresponde con la dirección lógica, en donde la información de mapeo comprende información sobre una página
física que almacena los datos comprimidos, una posición de inicio de los datos comprimidos en la página física, y
20 la longitud de los datos comprimidos.

12. El método según una cualquiera de las reivindicaciones 1 a 10, en donde después de la recepción, por parte del SSD, de una solicitud de escritura de un controlador, el método comprende, además:

asignar, por parte del SSD, una dirección lógica a los datos destinados a escribirse; y
después del almacenamiento, por parte del SSD, de los datos comprimidos, el método comprende, además:

25 grabar, por parte del SSD, información de mapeo en una entrada que se encuentra en la tabla de mapeo y se
corresponde con la dirección lógica, en donde la dirección de mapeo comprende información sobre una página
física que almacena los datos comprimidos; y
almacenar, por parte del SSD, metadatos de los datos comprimidos en la página física, en donde los metadatos
comprenden una posición de inicio de los datos comprimidos en la página física y la longitud de los datos
30 comprimidos.

13. El método según la reivindicación 11 ó 12, en donde la solicitud de escritura comprende la longitud de los datos destinados a escribirse, y la dirección lógica de los datos destinados a escribirse se asigna basándose en la longitud.

14. El método según una cualquiera de las reivindicaciones 1 a 13, en donde, después del almacenamiento, por parte del SSD, de los datos comprimidos, el método comprende, además:

35 enviar, por parte del SSD, la dirección lógica de los datos destinados a escribirse en el SSD al controlador.

15. El método según la reivindicación 14, en donde, después del almacenamiento, por parte del SSD, de los datos comprimidos, el método comprende, además:

recibir, por parte del SSD, una solicitud de lectura del controlador, en donde la solicitud de lectura comprende una dirección lógica de datos destinados a leerse;

40 leer, por parte del SSD, los datos objetivo en una dirección física correspondiente a la dirección lógica de los datos destinados a leerse;
descomprimir, por parte del SSD, los datos objetivo con el fin de obtener los datos destinados a leerse; y
enviar, por parte del SSD, los datos destinados a leerse al controlador.

16. Un aparato (1700) de almacenamiento, que comprende:

45 un primer módulo (1710) de recepción, configurado para recibir una solicitud de escritura de un controlador, en donde la solicitud de escritura es portadora de datos destinados a escribirse;
un módulo (1720) de compresión, configurado para comprimir los datos destinados a escribirse con el fin de obtener datos comprimidos;

50 un primer módulo (1730) de almacenamiento, configurado para almacenar los datos comprimidos; y
un primer módulo emisor (1740), configurado para enviar una primera información de retroalimentación al controlador, en donde la primera información de retroalimentación indica una capacidad restante del aparato de almacenamiento después de que se almacenen los datos comprimidos.

17. El aparato de almacenamiento según la reivindicación 16, en donde el aparato de almacenamiento comprende además:

5 un segundo módulo emisor, configurado para enviar una segunda información de retroalimentación al controlador, en donde la segunda información de retroalimentación indica la capacidad lógica del aparato de almacenamiento, y la capacidad lógica del aparato de almacenamiento es la suma de un volumen de datos sin comprimir de datos almacenados en el aparato de almacenamiento y una capacidad de espacio físico libre del aparato de almacenamiento.

18. El aparato de almacenamiento según la reivindicación 16 ó 17, en donde el aparato de almacenamiento comprende además:

10 un módulo de asignación, configurado para asignar un bloque lógico en blanco a los datos destinados a escribirse, en donde el bloque lógico en blanco es un bloque lógico con una dirección lógica libre, y cada bloque lógico se corresponde con un segmento de capacidad lógica del aparato de almacenamiento;
 un primer módulo de conversión, configurado para convertir el bloque lógico en blanco asignado en un bloque lógico válido, en donde el bloque lógico válido es un bloque lógico cuya dirección lógica está ocupada por datos válidos; y
 15 un primer módulo de consulta, configurado para consultar una cantidad de bloques lógicos en blanco restantes en el aparato de almacenamiento; en donde
 el primer módulo emisor (1740) está configurado específicamente para enviar la primera información de retroalimentación al controlador, en donde la primera información de retroalimentación comprende la cantidad de
 20 los bloques lógicos en blanco restantes o la capacidad restante, y la capacidad restante se determina basándose en la cantidad de los bloques lógicos en blanco restantes.

19. El aparato de almacenamiento según la reivindicación 18, en donde el módulo (1720) de compresión está configurado, específicamente, para: después de que el módulo de asignación asigne el bloque lógico en blanco a los datos destinados a escribirse, comprimir los datos destinados a escribirse con el fin de obtener los datos comprimidos;

25 y
 el aparato de almacenamiento comprende, además:
 un primer módulo de determinación, configurado para determinar, de acuerdo con un tamaño de los datos destinados a escribirse y un tamaño de los datos comprimidos, una cantidad de bloques lógicos en blanco que es necesario añadir en el aparato de almacenamiento.

30 20. El aparato de almacenamiento según la reivindicación 19, en donde el aparato de almacenamiento comprende además:

un módulo de adición, configurado para añadir, de acuerdo con la cantidad de los bloques lógicos en blanco que es necesario añadir en el aparato de almacenamiento, entradas correspondientes a bloques lógicos en blanco nuevos a una tabla de mapeo.

35 21. El aparato de almacenamiento según la reivindicación 19 ó 20, en donde el aparato de almacenamiento comprende además:

40 un módulo de actualización, configurado para actualizar un bloque lógico aislado en el aparato de almacenamiento a un bloque lógico en blanco de acuerdo con la cantidad de los bloques lógicos en blanco que es necesario añadir en el aparato de almacenamiento, en donde el bloque lógico aislado es un bloque lógico cuya dirección lógica no se puede usar en el aparato de almacenamiento.

22. El aparato de almacenamiento según una cualquiera de las reivindicaciones 19 a 21, en donde la cantidad M de bloques lógicos en blanco que es necesario añadir en el aparato de almacenamiento es igual a un resultado, redondeado a la baja, de dividir D por L, en donde D indica una diferencia entre una longitud de los datos destinados a escribirse y una longitud de los datos comprimidos, y L indica una longitud de un bloque lógico en el aparato de almacenamiento.

45 23. El aparato de almacenamiento según una cualquiera de las reivindicaciones 18 a 22, en donde el aparato de almacenamiento comprende además:

50 un segundo módulo de recepción, configurado para recibir una orden del controlador, en donde la orden incluye un intervalo de direcciones lógicas, y la orden se usa para dar instrucciones con el fin de marcar datos válidos en el intervalo de direcciones lógicas como datos no válidos; y
 un segundo módulo de conversión, configurado para convertir un bloque lógico válido del intervalo de direcciones lógicas en un bloque lógico aislado, en donde el bloque lógico aislado es un bloque lógico cuya dirección lógica no se puede usar.

55 24. El aparato de almacenamiento según la reivindicación 23, en donde el aparato de almacenamiento comprende además:

- un segundo módulo de determinación, configurado para determinar el tamaño del espacio físico ocupado por datos en el intervalo de direcciones lógicas;
 un módulo de selección, configurado para seleccionar un espacio de sobredimensionamiento parcial del espacio de sobredimensionamiento, en donde un tamaño del espacio de sobredimensionamiento parcial es igual al tamaño del espacio físico ocupado por los datos en el intervalo de direcciones lógicas; y
 un tercer módulo de determinación, configurado para usar el espacio de sobredimensionamiento parcial como el espacio físico libre del aparato de almacenamiento.
- 5
25. El aparato de almacenamiento según la reivindicación 16 ó 17, en donde el aparato de almacenamiento comprende además:
- 10 un segundo módulo de consulta, configurado para consultar el espacio físico libre del aparato de almacenamiento antes de que se envíe la primera información de retroalimentación al controlador; y
 un cuarto módulo de determinación, configurado para determinar la capacidad del espacio físico libre como la capacidad restante.
- 15 26. El aparato de almacenamiento según una cualquiera de las reivindicaciones 16 a 25, en donde el aparato de almacenamiento comprende además:
- un quinto módulo de determinación, configurado para asignar una dirección lógica a los datos destinados a escribirse después de que se reciba la solicitud de escritura desde el controlador; y
 un primer módulo de grabación, configurado para grabar información de mapeo en una entrada que está en la tabla de mapeo y se corresponde con la dirección lógica, en donde la información de mapeo comprende información sobre una página física que almacena los datos comprimidos, una posición de inicio de los datos comprimidos en la página física y la longitud de los datos comprimidos.
- 20
27. El aparato de almacenamiento según una cualquiera de las reivindicaciones 16 a 25, en donde el aparato de almacenamiento comprende además:
- un sexto módulo de determinación, configurado para asignar una dirección lógica a los datos destinados a escribirse después de que se reciba la solicitud de escritura del controlador;
 un segundo módulo de grabación, configurado para grabar información de mapeo en una entrada que está en la tabla de mapeo y se corresponde con la dirección lógica, en donde la información de mapeo comprende información sobre una página física que almacena los datos comprimidos; y
 un segundo módulo de almacenamiento, configurado para almacenar metadatos de los datos comprimidos en la página física, en donde los metadatos comprenden una posición de inicio de los datos comprimidos en la página física y la longitud de los datos comprimidos.
- 25
28. El aparato de almacenamiento según la reivindicación 26 ó 27, en donde la solicitud de escritura comprende la longitud de los datos destinados a escribirse, y la dirección lógica de los datos destinados a escribirse se asigna basándose en la longitud.
- 30
29. El aparato de almacenamiento según una cualquiera de las reivindicaciones 16 a 28, en donde el aparato de almacenamiento comprende además:
- un tercer módulo emisor, configurado para enviar la dirección lógica de los datos destinados a escribirse en el aparato de almacenamiento al controlador después de que se reciba la solicitud de escritura del controlador.
- 35
30. El aparato de almacenamiento según la reivindicación 29, en donde el aparato de almacenamiento comprende además:
- un tercer módulo de recepción, configurado para: recibir una solicitud de lectura del controlador, en donde la solicitud de lectura comprende una dirección lógica de datos destinados a leerse;
 un módulo de lectura, configurado para leer los datos objetivo en una dirección física correspondiente a la dirección lógica de los datos destinados a leerse;
 un módulo de descompresión, configurado para descomprimir los datos objetivo con el fin de obtener los datos destinados a leerse; y
 un cuarto módulo emisor, configurado para enviar los datos destinados a leerse al controlador.
- 40
31. Un sistema de almacenamiento, que comprende un controlador (1810) y el aparato (1700) de almacenamiento según una cualquiera de las reivindicaciones 16 a 30, en donde
- 45 el controlador (1810) está configurado para: generar una solicitud de escritura que comprende los datos destinados a escribirse; recibir un mensaje de respuesta para la solicitud de escritura del aparato (1700) de almacenamiento, en donde el mensaje de respuesta comprende una dirección lógica de los datos destinados a escribirse en el aparato (1700) de almacenamiento; y grabar una relación de mapeo entre la dirección lógica de los datos destinados a escribirse en una matriz de discos y la dirección lógica de los datos destinados a escribirse en el aparato (1700) de almacenamiento.
- 50
- 55

- 5 32. El sistema de almacenamiento según la reivindicación 31, en donde el controlador (1810) está configurado, además, para: recibir datos destinados a escribirse nuevos de un anfitrión; determinar una capacidad restante del aparato (1700) de almacenamiento de acuerdo con la primera información de retroalimentación; y cuando la capacidad restante del aparato (1700) de almacenamiento es mayor que un tamaño de los datos destinados a escribirse nuevos, escribir los datos destinados a escribirse nuevos en el aparato (1700) de almacenamiento.

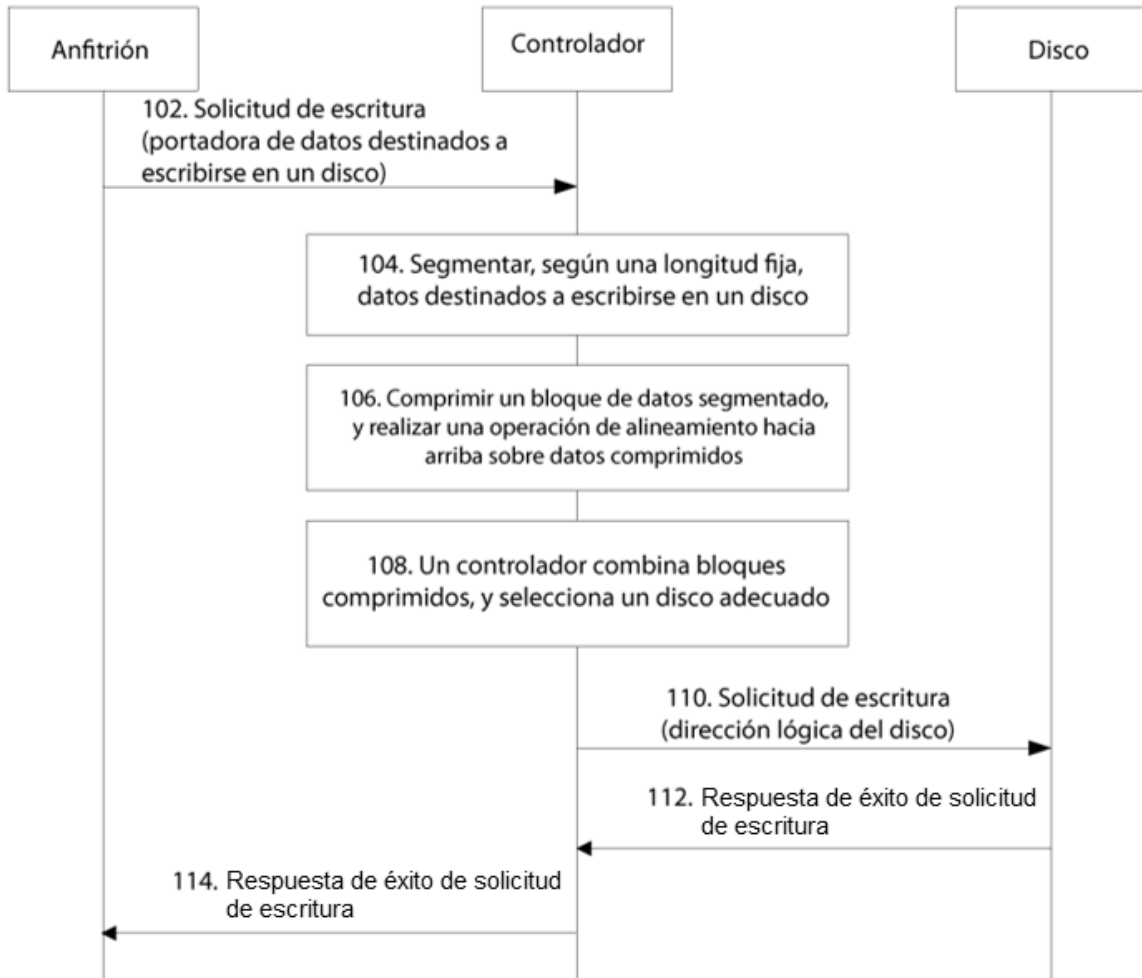


FIG. 1

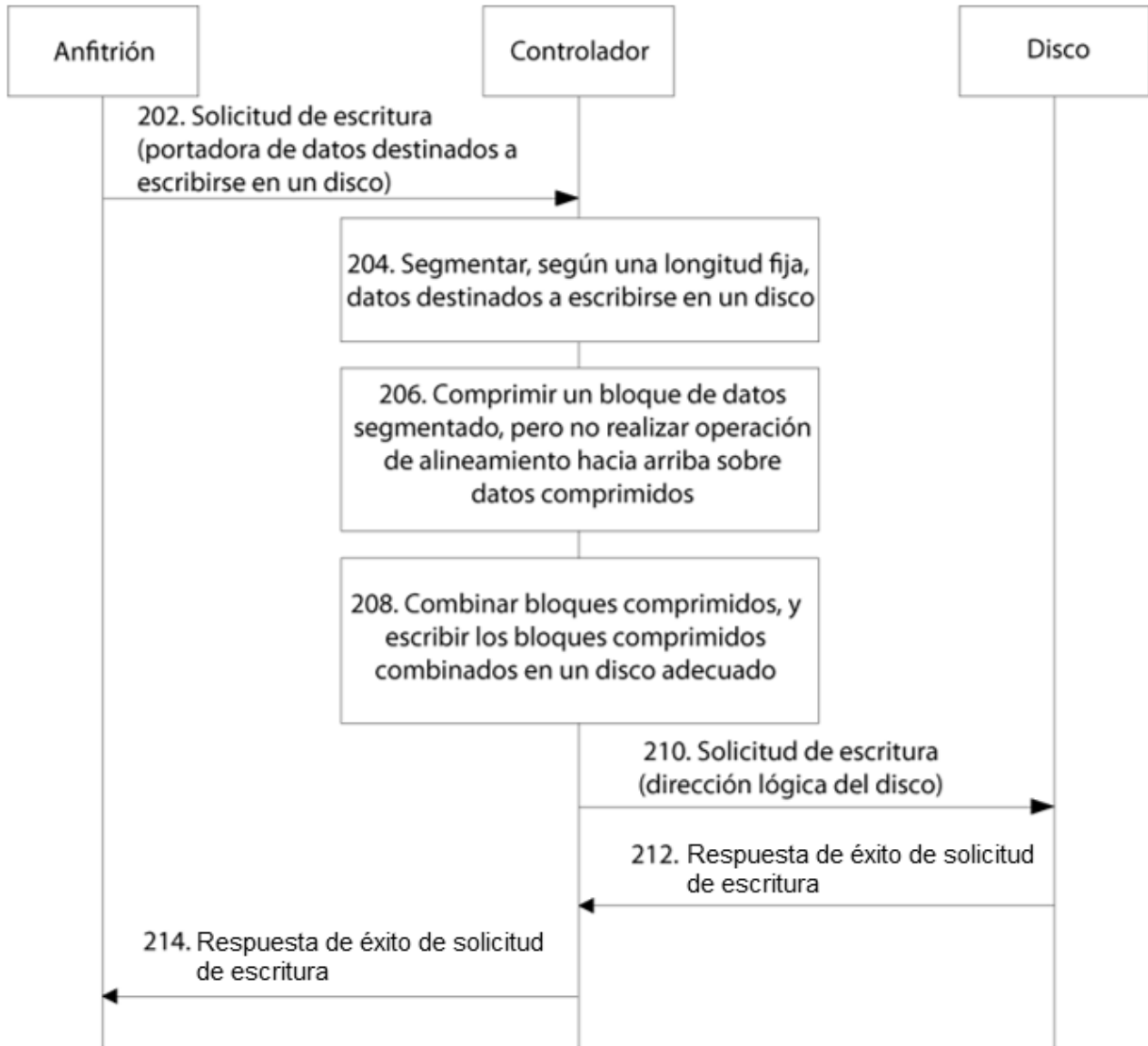


FIG. 2

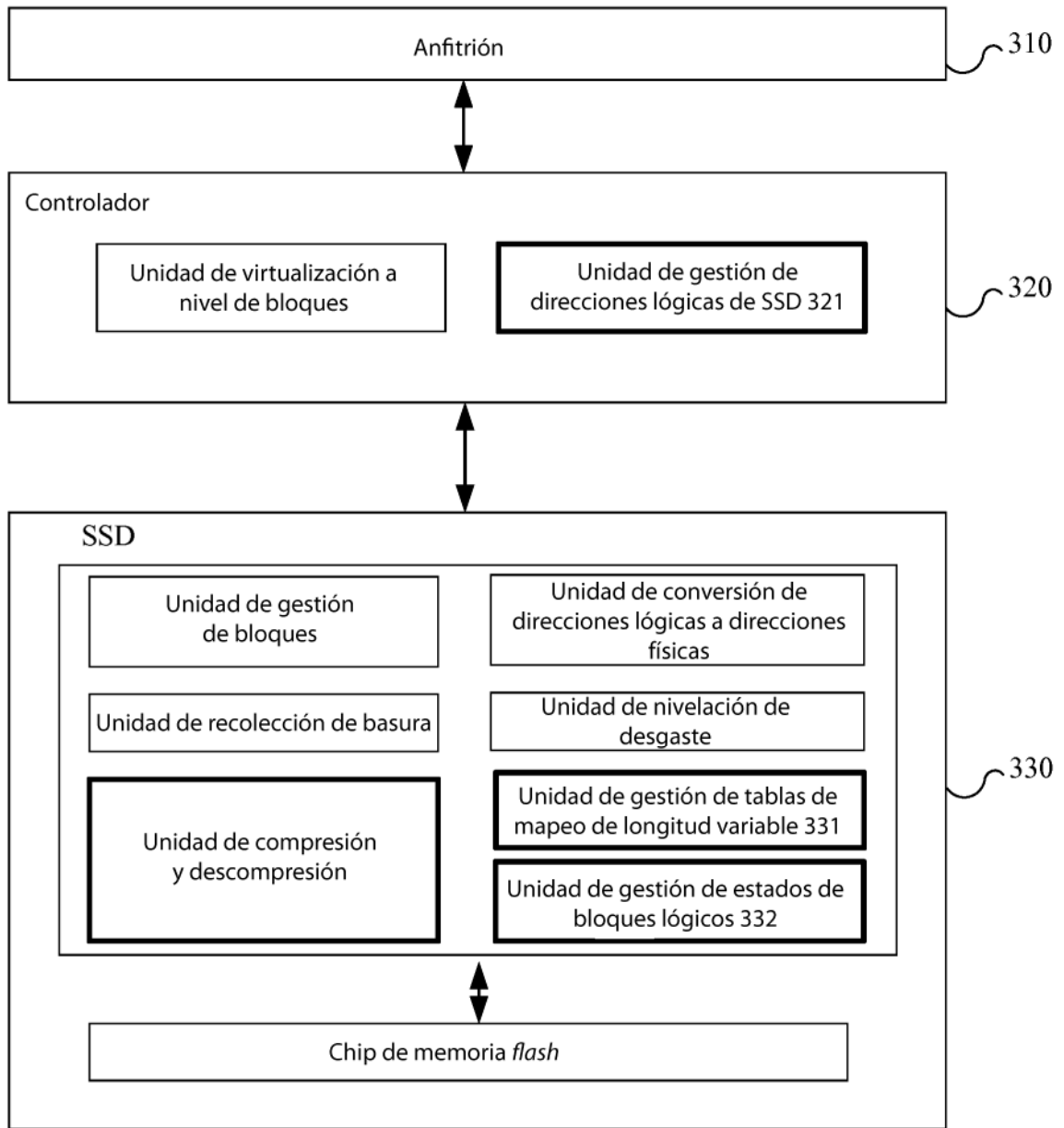


FIG. 3

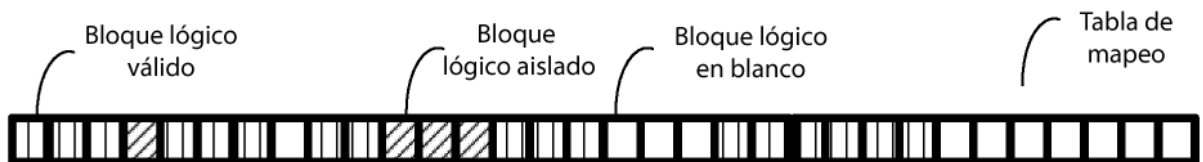


FIG. 4

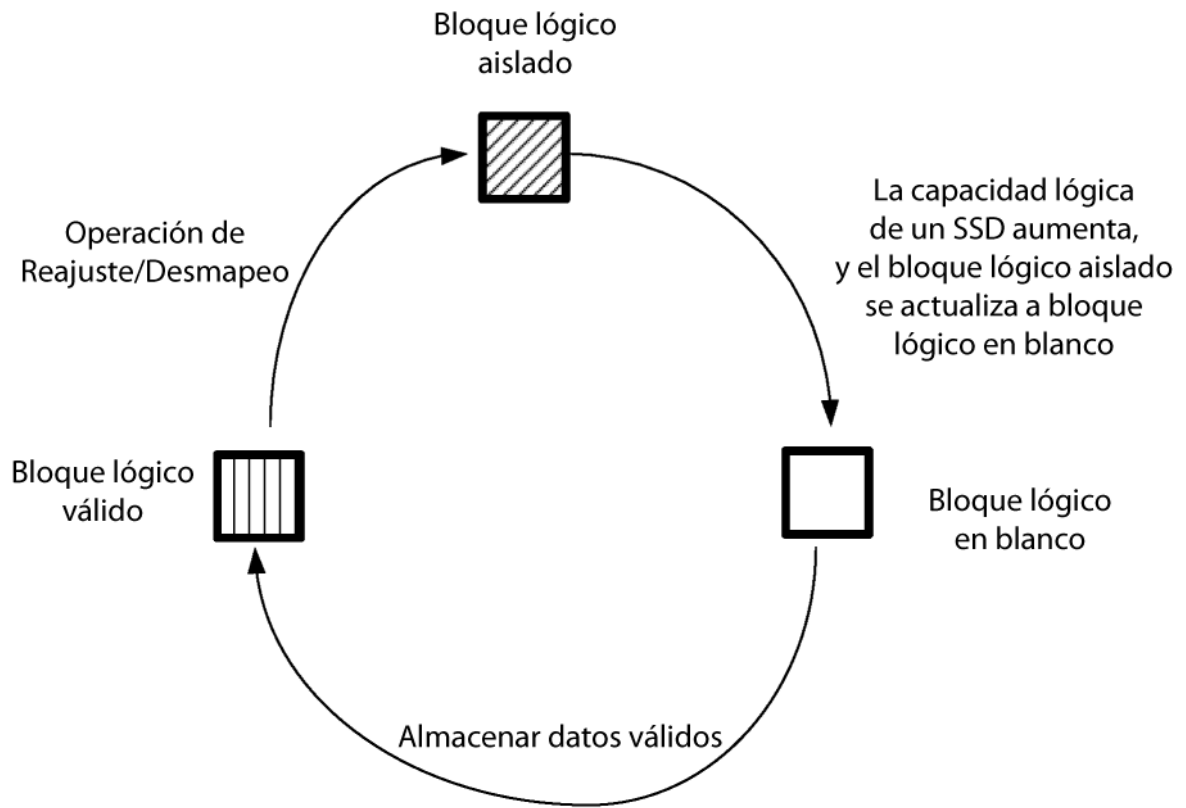


FIG. 5

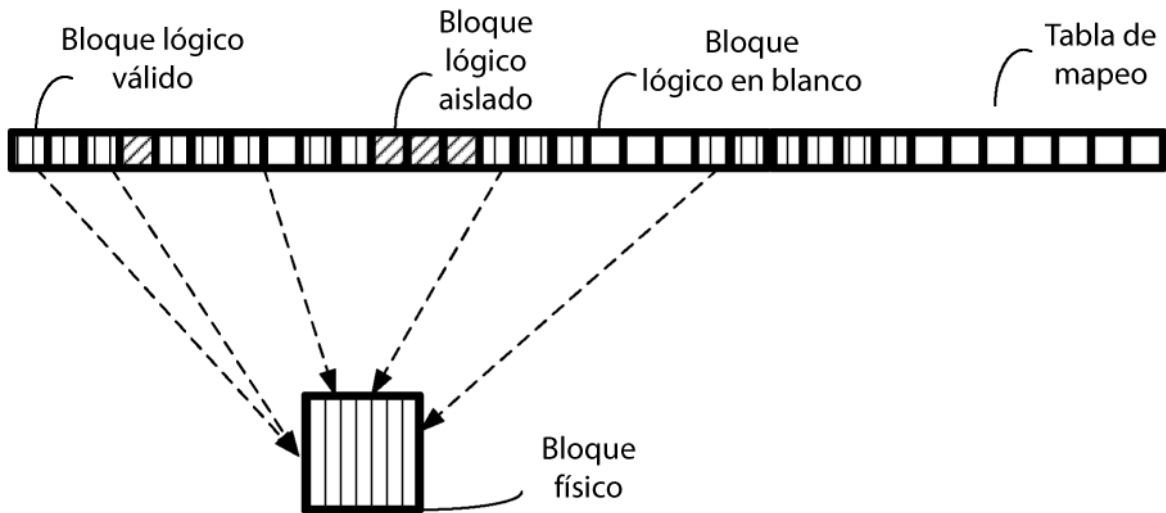


FIG. 6

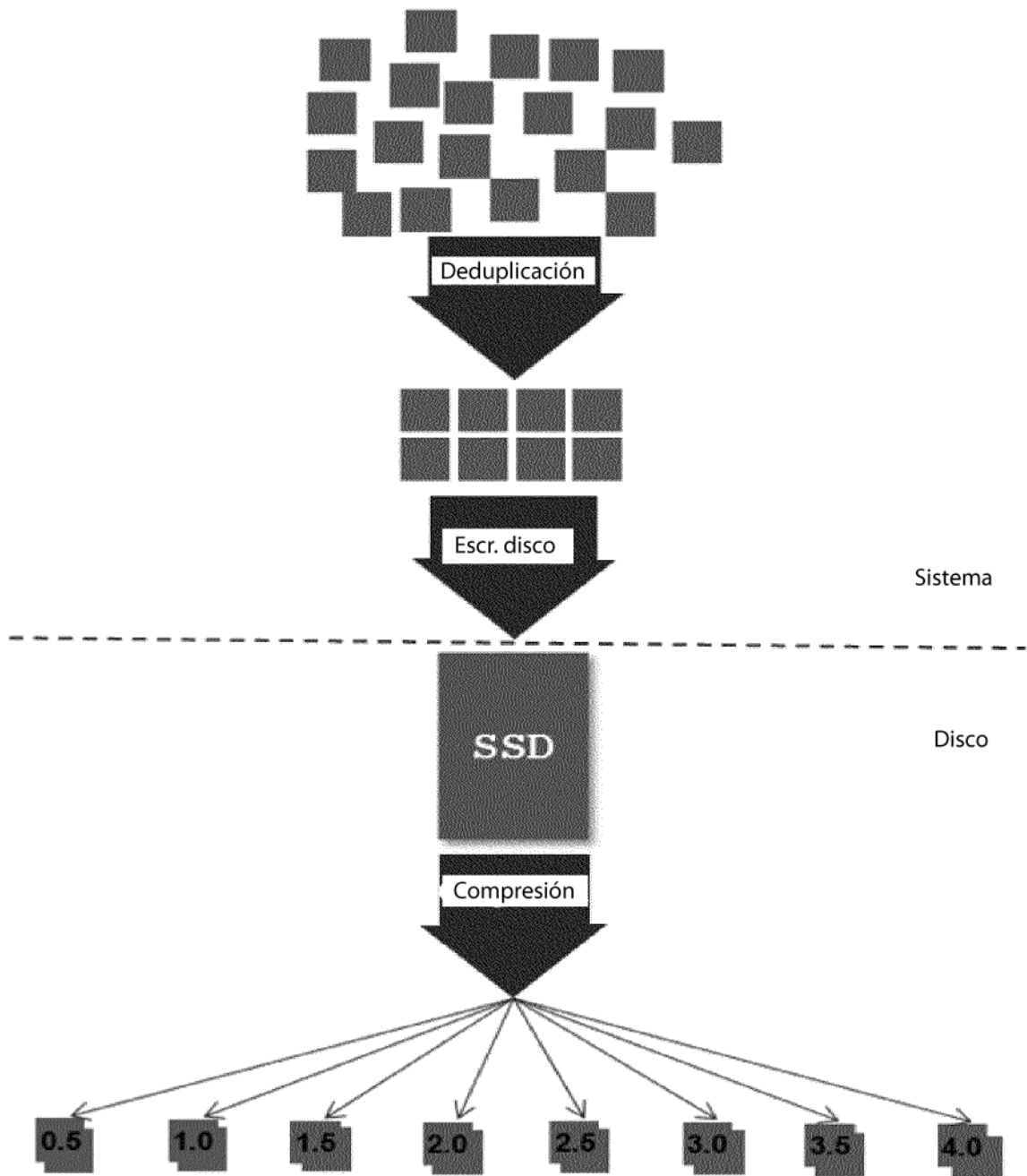


FIG. 7

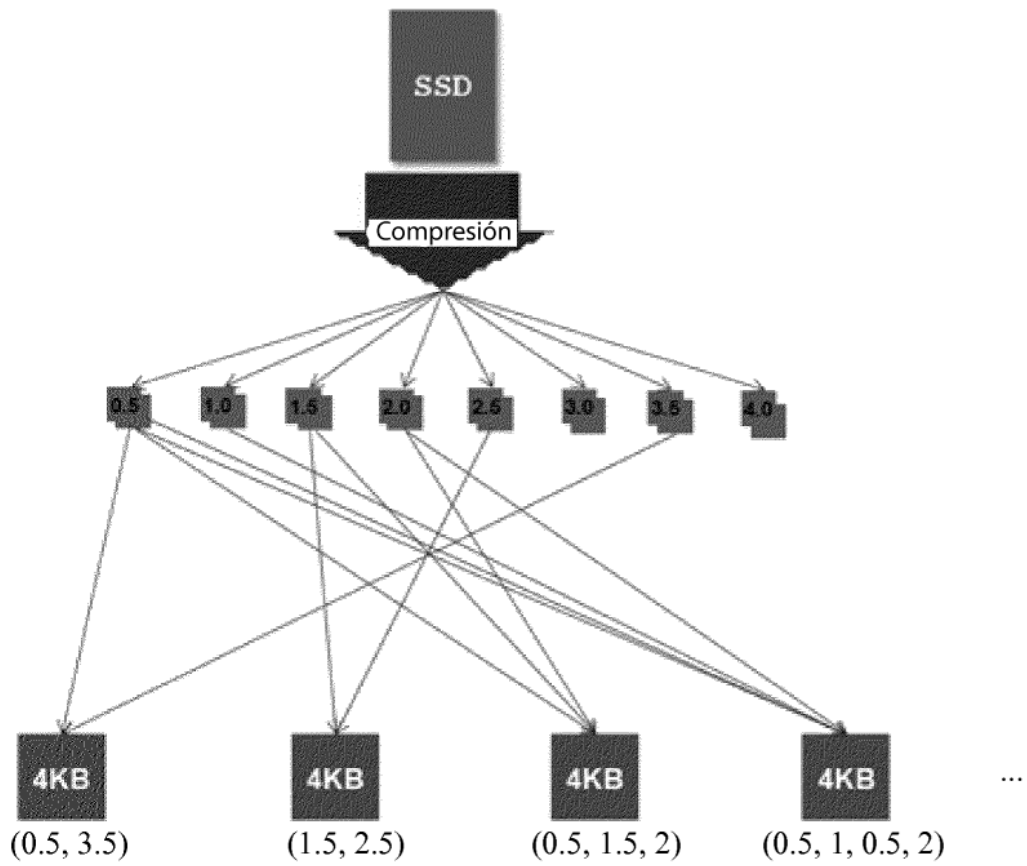


FIG. 8

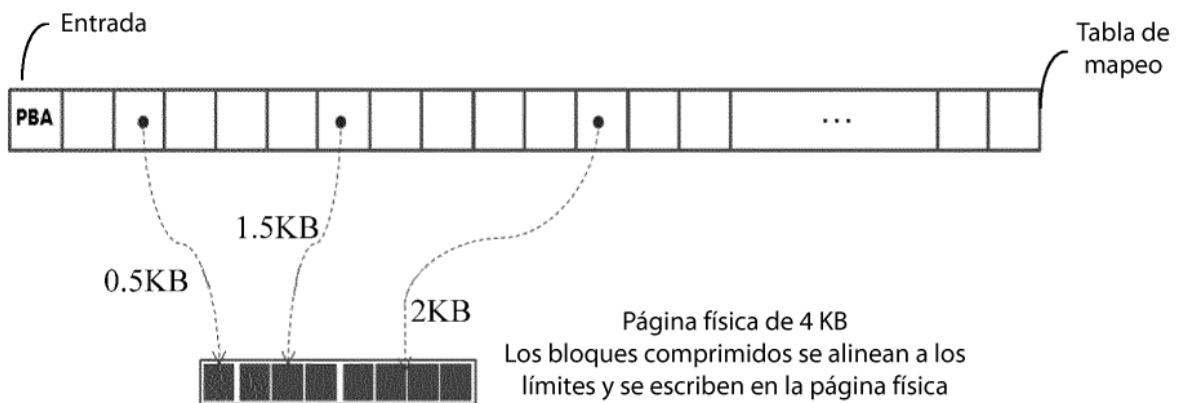


FIG. 9

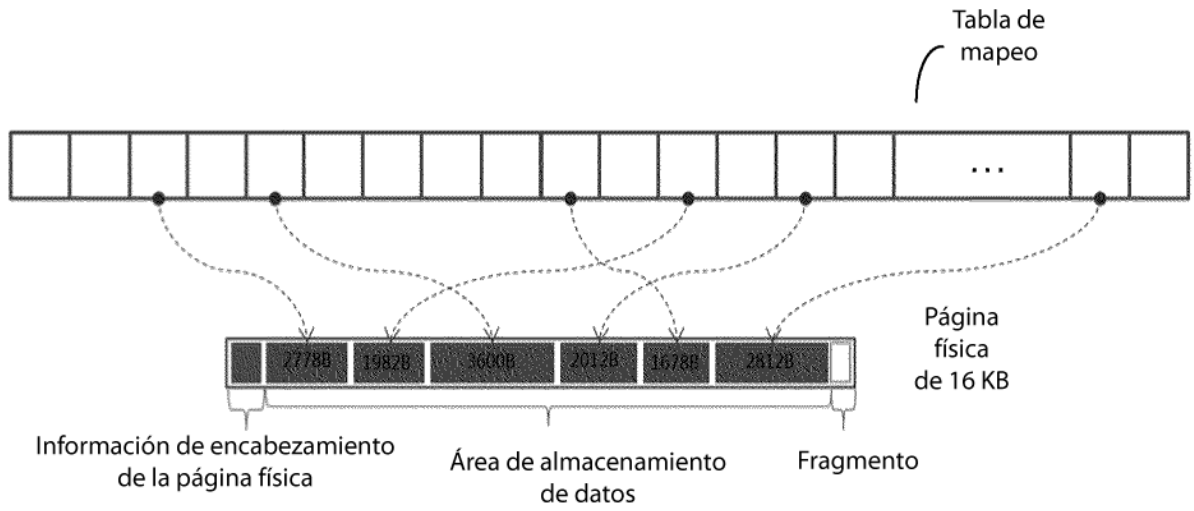


FIG. 10

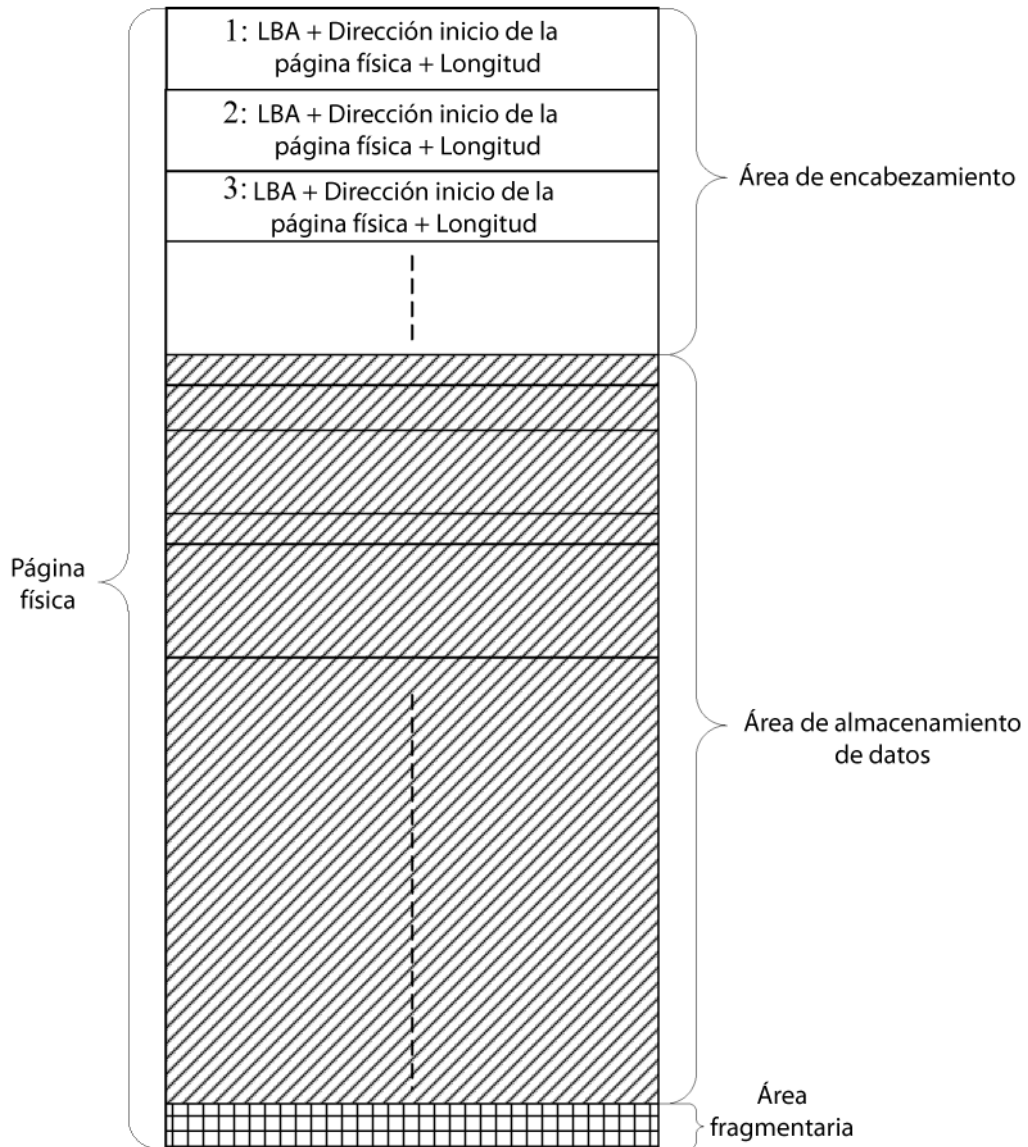


FIG. 11

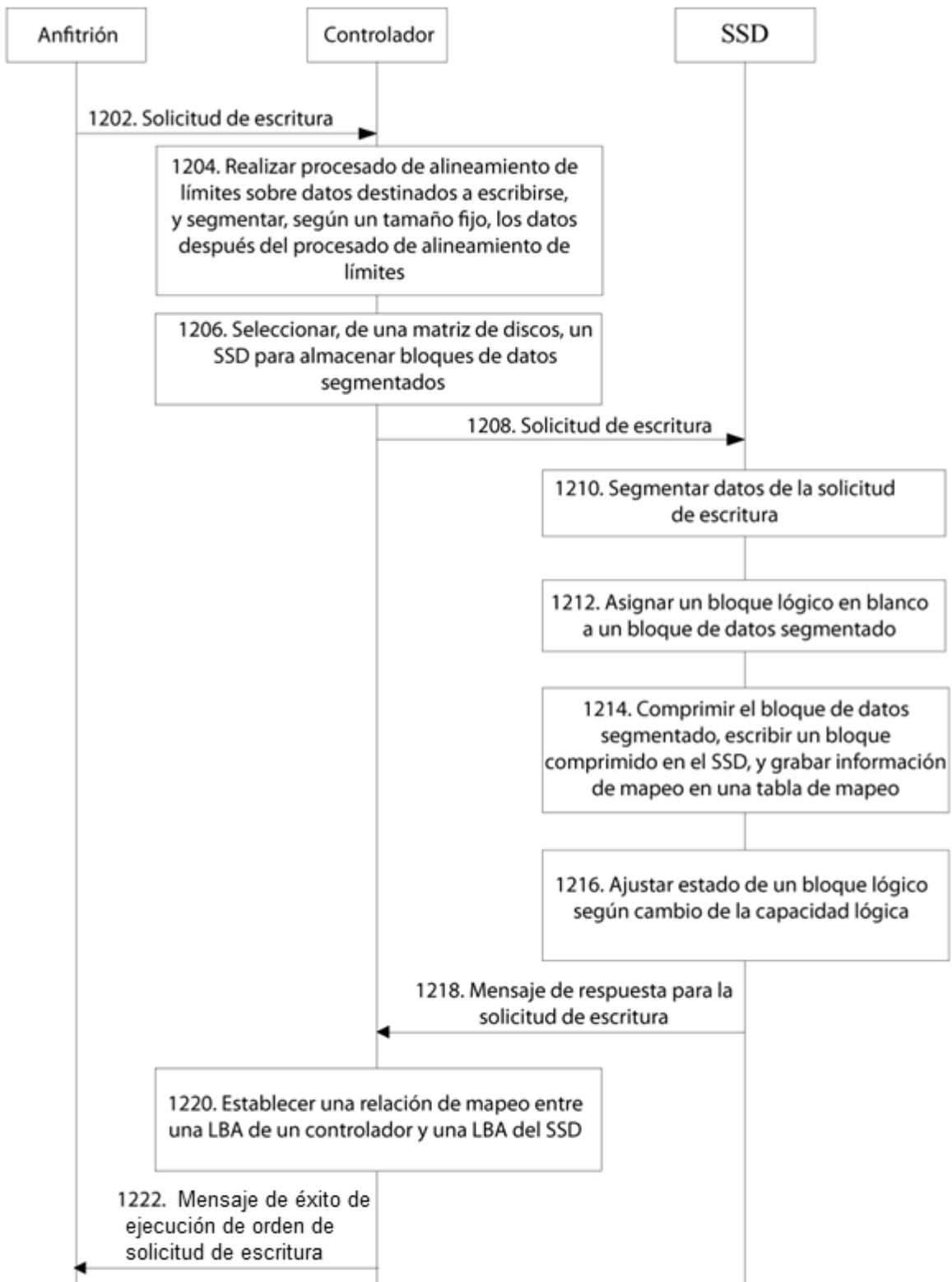


FIG. 12

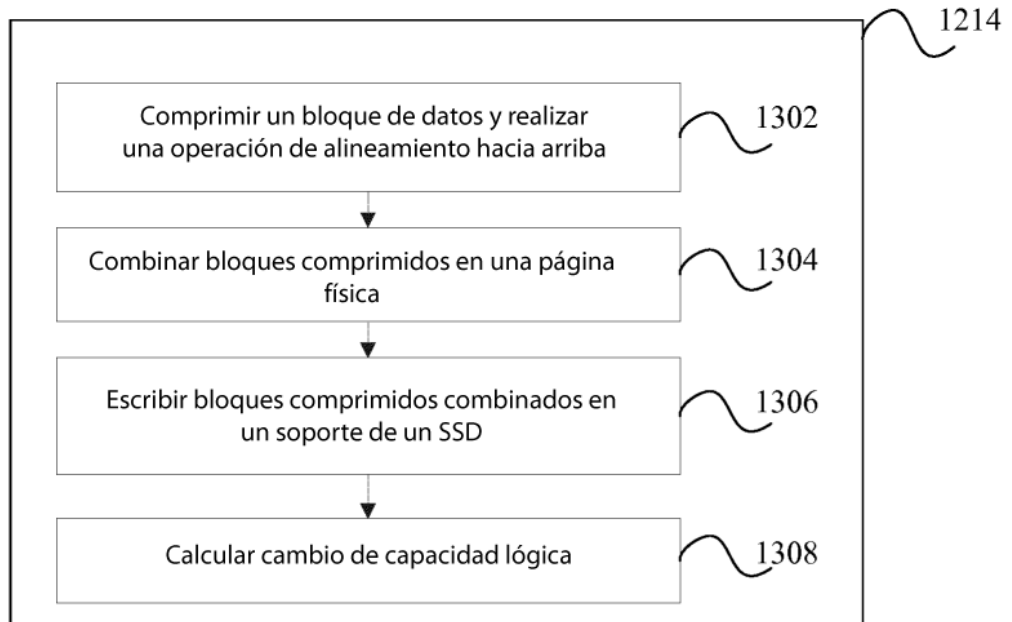


FIG. 13

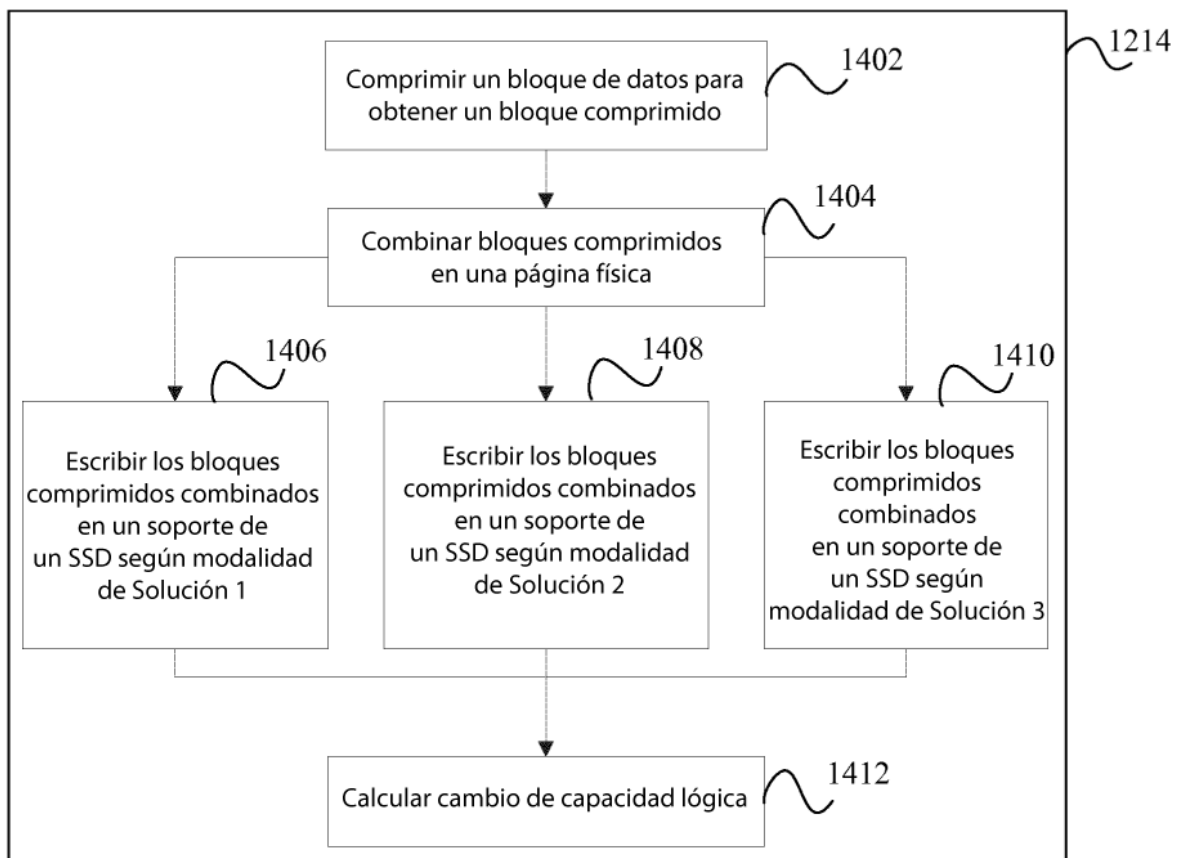


FIG. 14

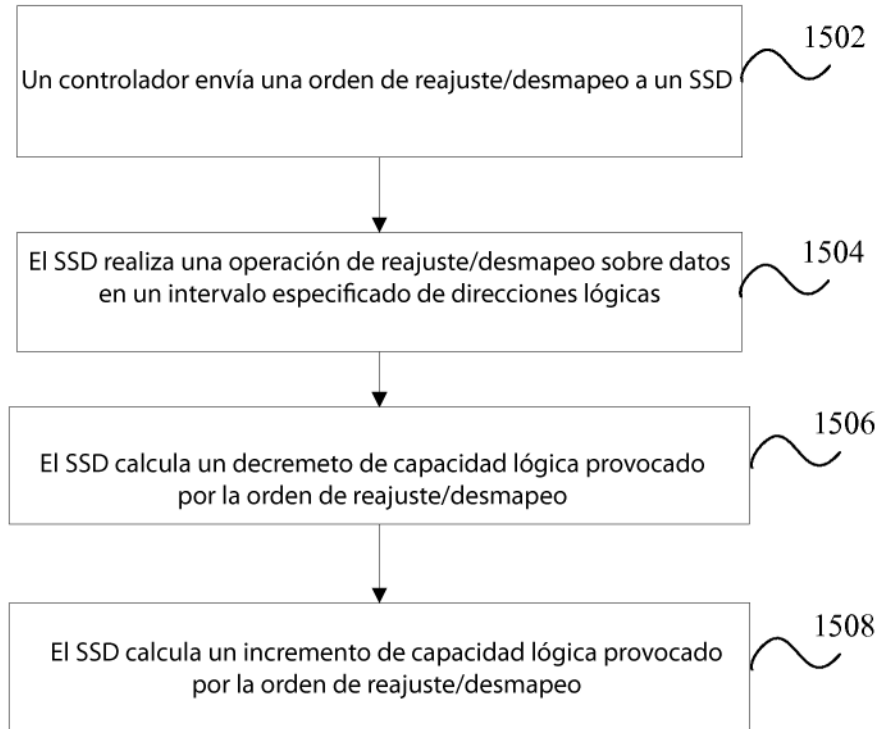


FIG. 15

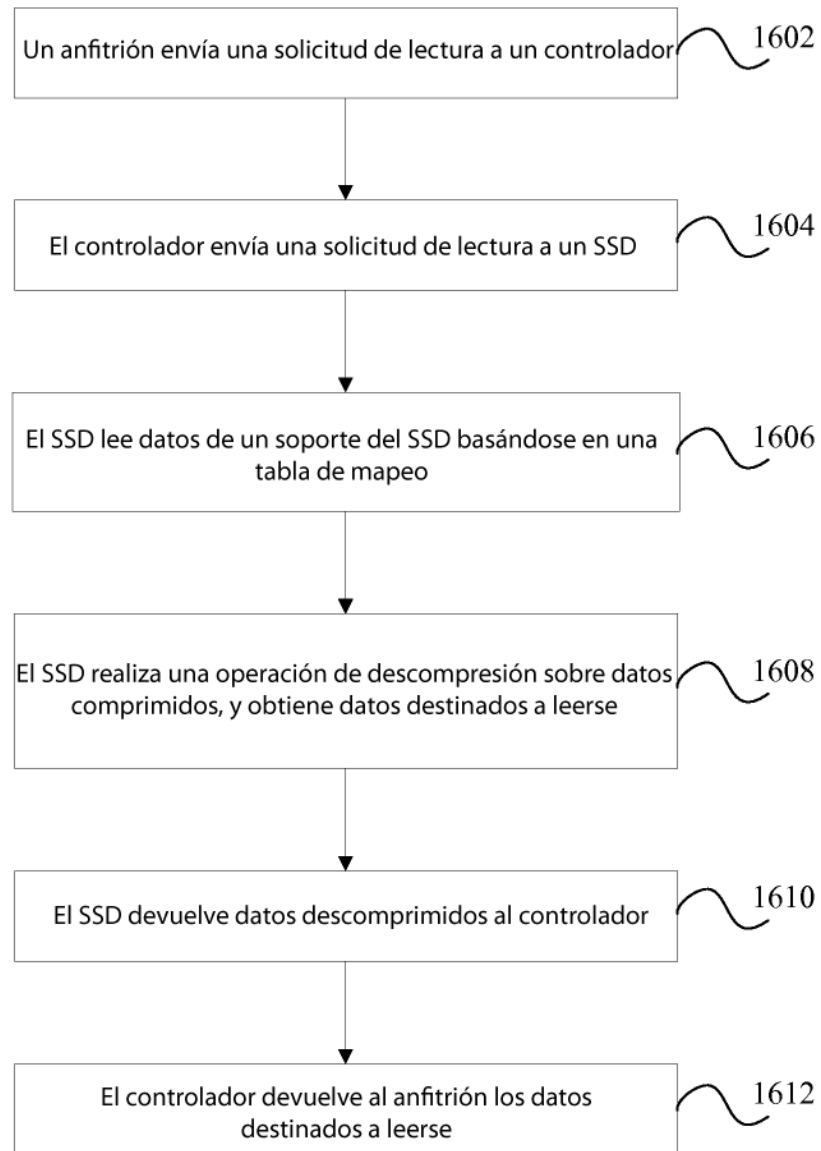


FIG. 16

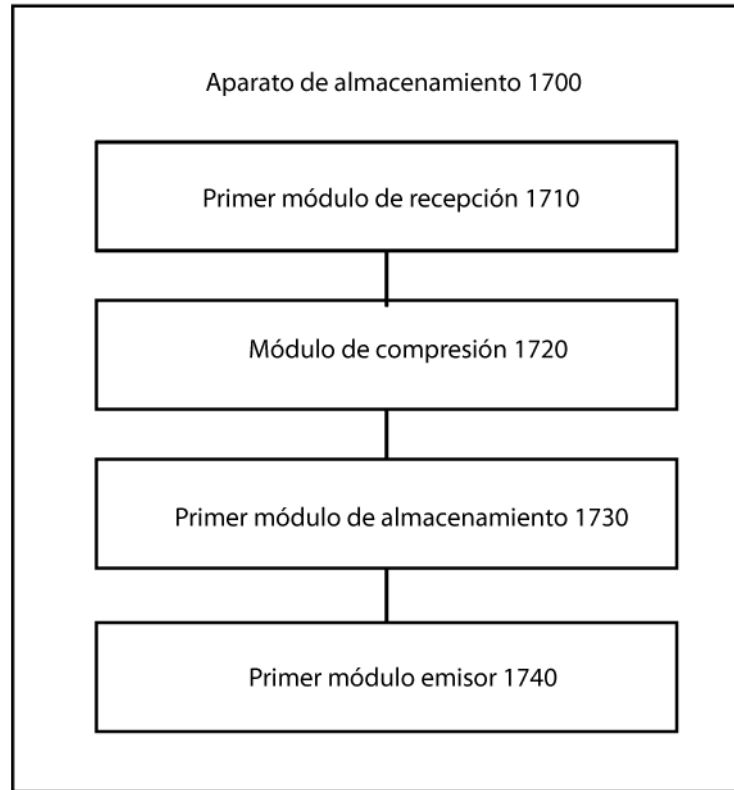


FIG. 17

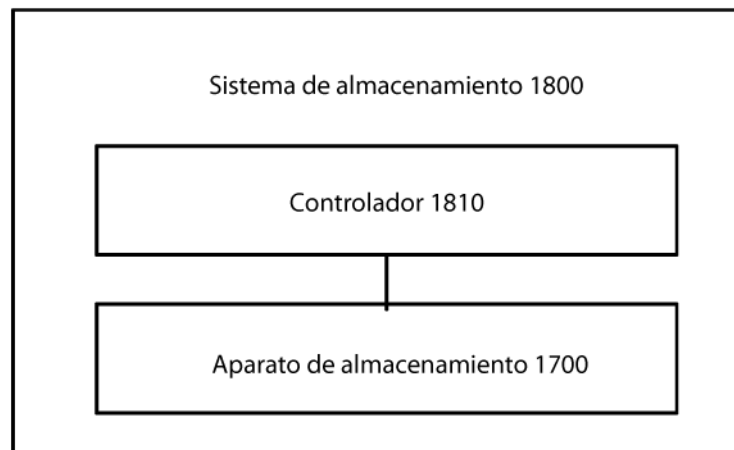


FIG. 18

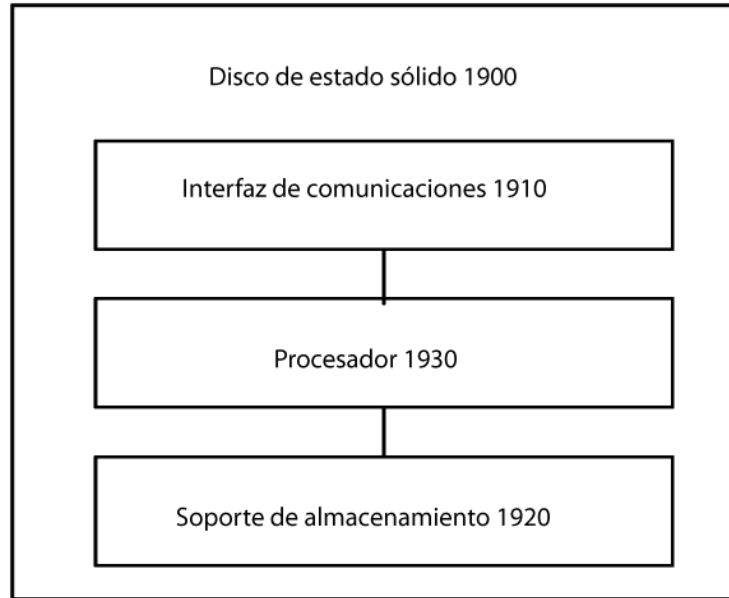


FIG. 19

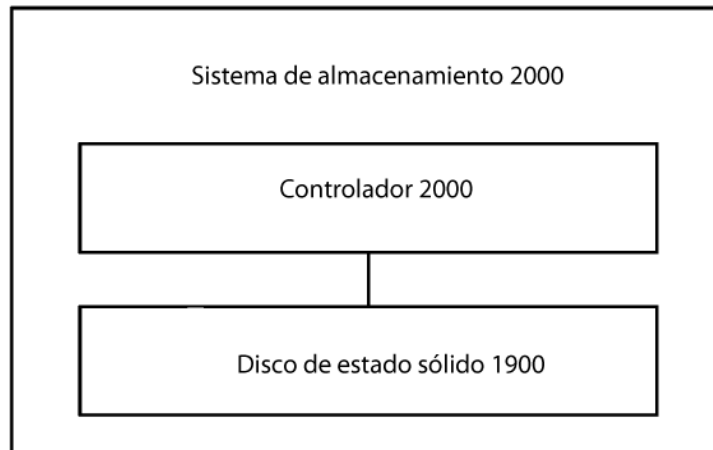


FIG. 20