



UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO



SISTEMAS OPERATIVOS

Deduplicación de Datos

Nombre:

Jessica Zepeda Baeza

Profesor:

Gunnar Wolf

Semestre:

2022-2

Fecha:

28 de abril de 2022

1. ¿Qué es la deduplicación?

La deduplicación o deduplicación de datos es el proceso que reconoce la repetición de datos y la elimina o sustituye por una referencia a los datos. Este proceso se realiza en almacenamiento a largo plazo como pueden ser discos o unidades SSD. La implementación de este proceso se ha realizado empleando distintas técnicas y algoritmos que han permitido que la deduplicación se divida en distintas categorías. No obstante, todas las técnicas hacen uso de algún tipo de almacenamiento donde se guardan segmentos de datos únicos así como un índice que incluye referencias para localizar estos datos. (Patricio, 2009) La deduplicación puede implementarse como software y como hardware; en el primero principalmente se utiliza en aplicaciones de backups de datos y en el segundo se implementa en los sistemas de archivos. (IONOS, 2021)

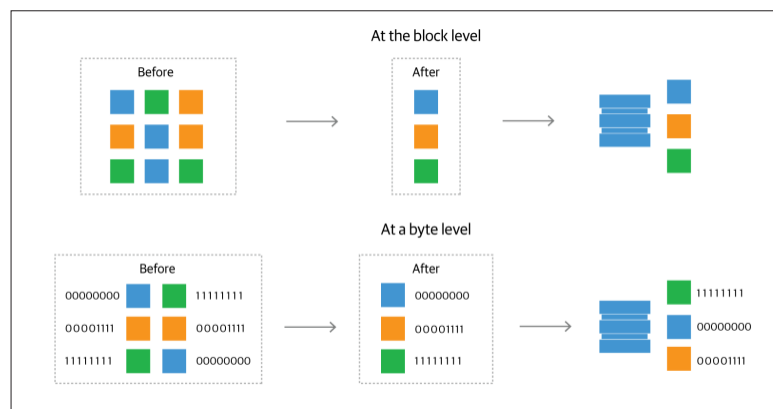


Figura 1: Proceso de Deduplicación de datos

La deduplicación depende en gran medida de los datos y es por ello que el proceso de deduplicación siempre procede un proceso previo de segmentación de los datos conocido como "Chunking". Este proceso divide en bloques los datos a deduplicar, de manera que al tener bloques de menor tamaño, la deduplicación tiene un impacto mayor. Esto último porque es mucho más probable que un conjunto pequeño de datos se repita que un conjunto grande. Sin embargo, tener bloques muy pequeños podría afectar el desempeño tanto del proceso de deduplicación como el de los recursos en la computadora.

La principal ventaja que ha impulsado este proceso a hacerse más utilizado y popular es que aumenta el tiempo de vida así como mejora la eficiencia de espacio en disco al eliminar datos que no sirven o que no se utilizan. También puede reducir el número de operaciones a disco y en el recolector de basura. (An y Shin, 2013)

2. Clasificaciones de la Deduplicación

Aplicar la deduplicación correctamente trae muchos beneficios; sin embargo para ello es necesario un análisis de cómo funciona el sistema a deduplicar para así decidir tanto qué algoritmo usar como qué enfoque implementar.

2.1. En origen / En destino

Esta clasificación se hace de acuerdo al agente de duplicación, donde se tienen los tipos en origen o en destino. (Patricio, 2009)

2.1.1. En origen

La deduplicación en origen se realiza cuando se tiene un sistema de origen y uno de destino como puede ser una arquitectura cliente servidor. En esta clasificación se realiza la deduplicación en el sistema de origen y después se transmite la información al sistema de destino. De esta manera, se evita el paso de datos duplicados al sistema destino y además se hace un ahorro en el volumen enviado.

2.1.2. En destino

La deduplicación en destino se realiza en el sistema de destino o aquel con el almacenamiento en disco. Existe otra clasificación dentro de esta: la deduplicación después del hecho o en vivo, es decir de acuerdo al instante de deduplicación.

2.2. En vivo / Después de hecho

Se hace una clasificación de acuerdo a cuándo se realiza la deduplicación de datos una vez que se encuentran en el sistema de destino. (Shin, 2015)

2.2.1. En vivo

La deduplicación en vivo, comúnmente conocida como inline u online se realiza al recibir los datos, es decir antes de escribir los datos en el disco. De esta forma se previenen la escritura de datos duplicados aunque se aumenta el tiempo de escritura al tener que revisar si estos datos son únicos. En otras palabras se tiene una entrada a escribir y la salida tardará más que si fuera inmediata debido a las comparaciones a hacer.

2.2.2. Después del hecho

La deduplicación después del hecho, offline o post process realiza la escritura de datos en el disco al recibir los datos y una vez que se tiene algún tiempo de inactividad se realiza la deduplicación. Esta técnica permite el paso de datos duplicados al disco; sin embargo, puede ayudar a reducir las operaciones de copia de páginas o referencias realizadas durante la recolección de basura. Se utiliza principalmente en sistemas de archivos debido a que hay un menor porcentaje de datos duplicados.

2.3. Con / Sin conocimiento del contenido

Esta clasificación se realiza dependiendo de si se conoce o no la estructura de los datos. En las técnicas donde se sabe el contenido o Content-Aware se tiene acceso a los metadatos que describen los datos almacenados de manera que al conocer la estructura de un dato es posible la identificación de objetos con la misma estructura y así identificar de forma más rápida aquellos datos duplicados. Por otro lado, en las técnicas sin conocimiento del contenido o agnósticas no se conoce esta estructura por lo que se aplican independientemente

del contenido de los datos y por consiguiente se generan muchos más posibles datos de entrada. (Patricio, 2009)

2.4. Por archivo / Por bloque

En esta clasificación depende del nivel de análisis es decir si se realiza a nivel de archivo (File) analizando conjuntos de datos o a nivel inferior, por bloque (Subfile) donde se divide el conjunto de datos en bloques de tamaño fijo y predecible. (Patricio, 2009) Estos bloques dependen de compromiso entre la organización física del disco, el aprovechamiento del espacio en disco y el ritmo de transferencia de datos. El sistema de archivos es el encargado de asignar un número m de bloques a cada archivo. (Guirado, s.f.) La deduplicación de bloque tiene una sobrecarga mayor a la deduplicación por archivo.

2.5. Por algoritmo

2.5.1. Hashing

Una vez que se hace la segmentación de los datos, se le aplica una función hash criptográfica a cada segmento con lo que se busca obtener un valor único. En caso de que no se obtenga dicho valor se sabe que existe una duplicación de datos. Las funciones hash criptográficas son funciones matemáticas unidireccionales que devuelven un valor único de acuerdo a una cierta entrada. Las características de estas funciones son:

- Son deterministas. Una entrada siempre devolverá el mismo valor hash.
- Son rápidas.
- No deben permitir colisiones (dos entradas con el mismo valor hash).
- Cambios leves en una entrada deben cambiar ampliamente el valor del hash de manera que no exista relación alguna entre ambos valores. (de soporte de SSL, s.f.)

Las funciones hash más utilizadas para la deduplicación SHA-1, SHA-2 y MDA5. SHA significa Secure Hash Algorithm y mientras SHA-1 permite convertir una cadena de datos de longitud variable en una cadena de 160 bits (número hexadecimal de 40 caracteres), SHA-2 la convierte en una cadena de 224, 256, 384 o hasta 512 bits. Actualmente, SHA-1 ya no se considera una función segura debido a el descubrimiento de la generación de colisiones en esta función. Este es un caso en el que el hashing podría no ser suficiente para identificar la duplicación de datos y es por ello que existen otros métodos como los que se mencionarán a continuación.

Existe un algoritmo llamado **Whole File Hashing (WFH)** donde se le aplica una función hash a todo un archivo (entran en la clasificación de deduplicación File) y si se encuentra un hash idéntico ya almacenado se comparan byte a byte estos archivos y se evita caer en un error de deduplicación debido a las colisiones de la función hash. (Mandagere, Zhou, Smith, y Uttamchandani, 2008)

También se tienen algoritmos que dividen dicho archivo en “chunks” o bloques y les aplican la función hash (deduplicación Subfile). En ese último caso se tienen dos tipos de

algoritmos: **Fixed Block Hashing (FBH)** y **Variable Block Hashing (VBH)** donde en el primero se divide un archivo en bloques de tamaño fijo y en el segundo en bloques de tamaño variable, de igual manera se les aplica la función hash y se compara byte a byte. (Mandagere y cols., 2008)

2.5.2. Delta Encoding (DE)

Por último se tiene la función **Data Encoding (DE)** donde se genera una delta o patch file entre dos archivos donde esta delta contiene la diferencia entre ambos archivos. De esta forma es sencillo localizar aquellos pedazos repetidos. En un sistema que usa DE se debe elegir cuidadosamente el archivo contra el cual generar la delta cuando se recibe un nuevo archivo. (Mandagere y cols., 2008)

3. ¿Cómo funciona la deduplicación?

Hablar más a detalle de la deduplicación de datos depende en gran medida de la configuración de este proceso; como se explicó, puede tener muchas variantes que laboran internamente de distinta manera. Sin embargo, en lo que involucra a los sistemas de archivos y al almacenamiento en disco generalmente se realiza una deduplicación offline utilizando funciones hash, esta técnica es la que se explicará.

Primero se identifican aquellos datos que se van a deduplicar. Después se dividen estos datos en chunks (ya sea de tamaño variable o fijo) y se identifican los segmentos únicos (por las técnicas de hashing ya mencionadas). Se almacenan los segmentos únicos en un espacio designado en el disco y se reemplazan los segmentos de los datos originales por apuntadores que indican la localidad del segmento que pertenece a esos datos. (Microsoft, 2022)

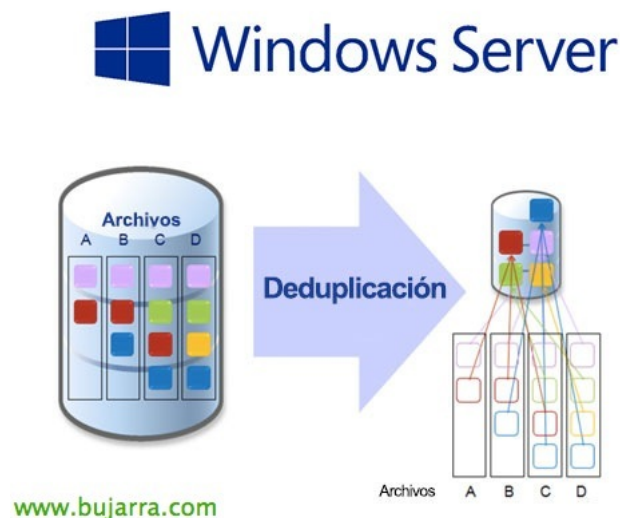


Figura 2: Deduplicación en el disco.

Se han encontrado técnicas para que la deduplicación sea más eficiente en cuanto a tiempo y espacio mezclando la deduplicación online y offline, así como usando hashing tanto criptográfico como no criptográfico. Un ejemplo de esta técnica se llama **Block Separation** y se

explicará a continuación. El principal objetivo de esto es identificar bloques que posiblemente estén duplicados durante el proceso inline (antes de escribirlos en el disco) para así dividir los datos en dos: Datos Únicos y Datos No Determinados. (An y Shin, 2013)

El primer paso para esta técnica es identificar los datos a deduplicar y dividirlos en chunks de tamaño fijo durante la parte inline, este tamaño es el mismo que el de una página en disco. Para cada segmento se obtiene una función hash que no es libre de colisiones como CRC32. Se revisa si el hash generado ya existe en una tabla con los valores hash CRC32 de los chunks ya escritos; si no existe significa que se trata de un segmento único que se debe almacenar en un bloque de la región de Datos Únicos y se almacena este hash en la tabla. En caso de que se repita el hash, se puede tratar de un dato duplicado por lo que se almacena en un bloque de la región de Datos No Determinados. (An y Shin, 2013)

A los datos de la región No Determinada se les aplica una función hash criptográfica como SHA-2 durante el proceso offline y se revisa otra vez en la tabla de hash SHA-2 si ya existe este hash. En caso de que exista, se invalida la página del chunk y se actualiza la tabla de mapeo para que todas las páginas lógicas inválidas compartan una página física. Si el hash no existe tan sólo se valida la página y se puede copiar en la región Única. Es por ello que tanto la región Única como la No Determinada tienen tamaño variable. Durante el proceso de offline también se obtienen los hashes SHA-2 de los chunks Únicos; sin embargo, no es necesario hacer la comparación para encontrar duplicados lo que ahorra operaciones. (An y Shin, 2013)

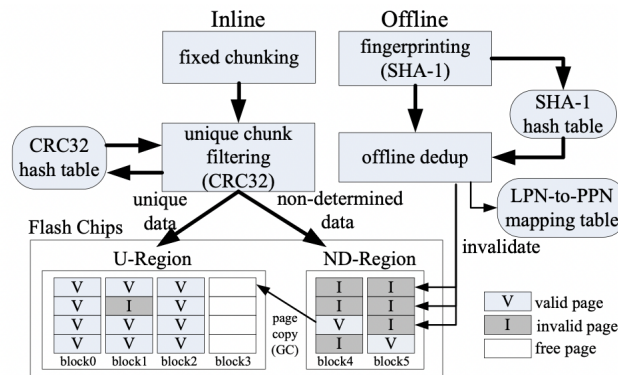


Figura 3: Técnica de Block Separation para la deduplicación de datos.

La región No Determinada tendrá mucha duplicación de datos por la naturaleza de los hashes lo que significa que habrá muchas páginas inválidas. Esto último le permite al recolector de basura identificar fácilmente a una víctima y borrar un bloque. Además al invalidar páginas duplicadas, las páginas válidas tendrán un menor tamaño y se reduce el ratio de amplificación de escritura. (An y Shin, 2013)

4. Implementaciones Actuales

La deduplicación de datos ha adquirido su mayor aplicación en sistemas de backups ya que estos, en cada backup, contienen casi la misma información por lo que se genera mucha

duplicación de datos y por consiguiente se ocupa mucho espacio. Los sistemas de virtualización se benefician en gran medida de la deduplicación ya que comparten bloques de datos muy grandes con el sistema operativo que no es necesario almacenar por cada máquina virtual, especialmente si las máquinas virtuales emplean el mismo sistema operativo. (33HOPS, 2015) Esto debido a que la imagen de una máquina virtual en su mayoría, duplica datos. (Solís, 2015)

El término **overcommitment** o sobrecompromiso habla sobre la asignación de almacenamiento persistente a una máquina virtual mayor al almacenamiento que se tiene. (Oracle, s.f.) La deduplicación en todo caso evitaría este problema ya que proporcionaría más espacio al eliminar copias redundantes. En otras palabras, la deduplicación permite sobrecomprometer el almacenamiento de la máquina ya que optimiza el espacio. Como ejemplos de aplicaciones o sistemas operativos virtuales u open source que implementan la deduplicación existen: Windows Server desde 2016, Azure Stack HCI, VeeamBackup, BorgBackup, 33HOPS.

5. Compresión de datos y Single Instance Store (SIS)

La compresión de datos es otra técnica que permite decodificar la información de ciertos archivos a menos bits para reducir el espacio de almacenamiento así como también se optimiza la duplicación de patrones dentro de un archivo y entre archivos. Sin embargo, no aporta los mismos resultados que la deduplicación ya que para acceder a un archivo se necesita realizar una descompresión lo que podría incluir un cambio en el archivo. (IONOS, 2021)

Por otra parte se tiene la técnica Single Instance Store o SIS que significa almacenamiento de instancia única. Esta técnica también puede parecer similar a la deduplicación ya que optimizaba almacenamiento al identificar archivos totalmente idénticos a otro ya almacenado y reemplazar su almacenamiento por una referencia a este archivo. En realidad, SIS es el predecesor de la deduplicación y ahora, un tipo de deduplicación: por archivo; pero esta última tiene la principal ventaja de no solo comparar archivos completos sino también bloques dentro de estos que contienen patrones repetidos en varios archivos, reduciendo aún más el posible almacenamiento. (Microsoft, 2022) De hecho, la primera aparición de la deduplicación en un software fue en 1997 en Exchange Server 5.5 de Microsoft como SIS; sin embargo, fue hasta 2007 cuando se empezó a desarrollar y dar a conocer esta tecnología. (Solís, 2015)

6. Puntos en contra

A pesar de proporcionar en su mayoría ventajas, la deduplicación también genera puntos en contra al ser susceptible a errores; estos errores incluyen el rendimiento y la confiabilidad.

El problema de rendimiento se da en el evento en que un archivo esté duplicado en su mayoría lo que significaría que acceder a él, involucraría lecturas de varios bloques que muy probablemente no estén contiguos. Esto último implica que la cabeza lectora se desplace más y por consiguiente que se tenga un acceso o recuperación de archivos más tardado (menos rendimiento).

La confiabilidad se refiere a el proceso de garantizar la exactitud de la información. En este caso hace referencia a las pérdidas de datos que puede sufrir el sistema de archivos y cómo afecta el acceso. Es decir, si se pierde o daña un bloque que contiene los datos que son referenciados por más bloques se tendrá un daño en cada archivo que ocupe ese bloque. Es un impacto mucho mayor a aquel que ocurre cuando no existe deduplicación ya que sólo se dañaría un archivo y no “propagaría” este daño a más.

Referencias

- 33HOPS. (2015). Deduplicación de datos a nivel de bloque. *33HOPS*. Descargado de <https://33hops.com/deduplicacion-de-datos.html>
- An, J., y Shin, D. (2013). Offline deduplication-aware block separation for solid state disk. Descargado de https://www.usenix.org/system/files/fastpw13-paper6_0.pdf
- de soporte de SSL, E. (s.f.). ¿qué es una función criptográfica de hash? *SSL*. Descargado de <https://www.ssl.com/es/preguntas-frecuentes/Qu-es-una-funcin-hash-criptogrifica/%3F/>
- Guirado, R. (s.f.). Tema 7: El sistema de archivos. *Universidad de Almería*. Descargado de <https://w3.ual.es/~rguirado/so/tema7>
- IONOS. (2021). Reducción de datos por deduplicación o por compresión. *IONOS Digital Guide*. Descargado de <https://www.ionos.mx/digitalguide/servidores/know-how/reduccion-de-datos-deduplicacion-o-compresion/>
- Mandagere, N., Zhou, P., Smith, M. A., y Uttamchandani, S. (2008). Demystifying data deduplication. Descargado de <http://cs.brown.edu/courses/cs227/archives/2016/papers/p12-mandagere.pdf#page9>
- Microsoft. (2022). Understanding data deduplication. Descargado de <https://docs.microsoft.com/en-us/windows-server/storage/data-deduplication/understand>
- Oracle. (s.f.). 4.10. memory overcommitment. *Oracle*. Descargado de <https://docs.oracle.com/en/virtualization/virtualbox/6.0/user/guestadd-memory-usage.html>
- Patricio, F. J. J. (2009). Técnicas de deduplicación de datos y aplicación en librerías virtuales de cintas. *Lenguajes y Sistemas Informáticos e Ingeniería del Software*. Descargado de https://oa.upm.es/1803/1/PFC_FRANCISCO_JAVIER_JIMENEZ_PATRICIO.pdf
- Shin, D. (2015). Offline deduplication with lightweight hash for solid state disk. *The 22nd Korean Conference on Semiconductors*. Descargado de http://nyx.skku.ac.kr/wp-content/uploads/2014/07/offline_deduplication_with_lightweight_hash_for_ssd.pdf
- Solís, D. G. M. (2015). Análisis de métodos de deduplicación de datos aplicados en repositorios linux para la facultad de ingeniería en sistemas electrónica e industrial. *UNIVERSIDAD TÉCNICA DE AMBATO*. Descargado de http://repositorio.uta.edu.ec/bitstream/123456789/19367/1/Tesis_t1081si.pdf