



Configuración de la base de datos

Enterprise applications

NetApp
May 09, 2024

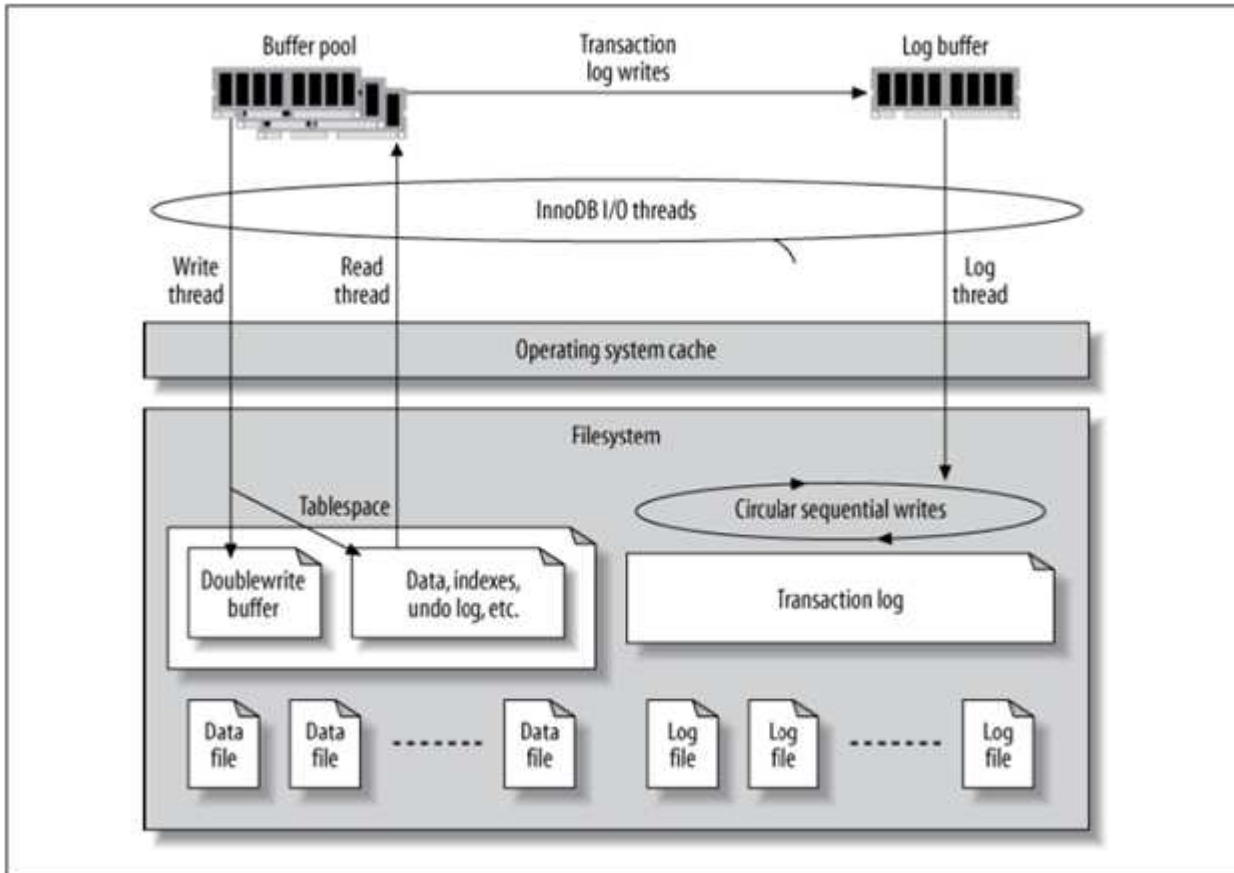
Tabla de contenidos

- Configuración de la base de datos 1
 - MySQL e InnoDB 1
 - Parámetros de configuración de MySQL 3
 - innodb_log_file_size 4
 - innodb_flush_log_at_trx_commit 4
 - innodb_doublewrite 5
 - innodb_buffer_pool_size 5
 - innodb_flush_method 5
 - innodb_io_capacity 6
 - innodb_lru_scan_profund 7
 - open_file_limits 7

Configuración de la base de datos

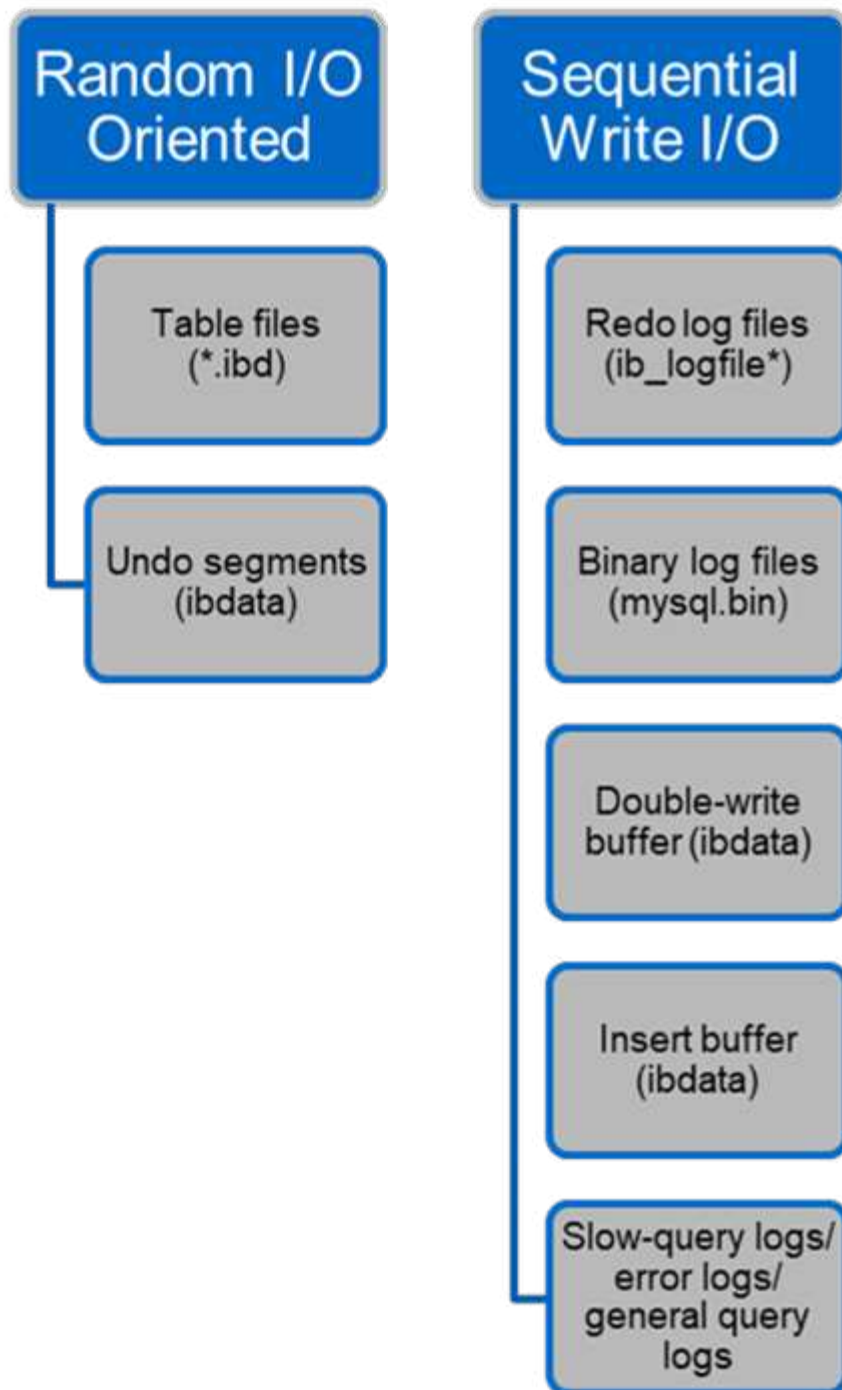
MySQL e InnoDB

InnoDB actúa como la capa media entre el almacenamiento y el servidor MySQL, almacena los datos en las unidades.



MySQL I/O se clasifica en dos tipos:

- I/O de archivo aleatoria
- E/S de archivo secuencial



Los archivos de datos se leen y se sobrescriben aleatoriamente, lo que da como resultado un elevado número de IOPS. Por lo tanto, se recomienda el almacenamiento SSD.

Los archivos de registro de recuperación y los archivos de registro binarios son registros transaccionales. Se escriben secuencialmente, por lo que puede obtener un buen rendimiento en HDD con la caché de escritura. Se produce una lectura secuencial en la recuperación, pero rara vez causa un problema de rendimiento, porque el tamaño de los archivos de registro suele ser menor que el de los archivos de datos y las lecturas secuenciales son más rápidas que las lecturas aleatorias (se producen en archivos de datos).

El buffer de doble escritura es una característica especial de InnoDB. En primer lugar, InnoDB escribe las páginas vaciadas en el buffer de doble escritura y, a continuación, escribe las páginas en sus posiciones

correctas en los archivos de datos. Este proceso evita la corrupción de páginas. Sin el búfer de doble escritura, la página puede dañarse si se produce un fallo de alimentación durante el proceso de escritura en unidades. Como la escritura en el búfer de doble escritura es secuencial, está muy optimizada para HDD. Las lecturas secuenciales se producen en la recuperación.

Como la NVRAM de ONTAP ya proporciona protección de escritura, no es necesario el almacenamiento en búfer de doble escritura. MySQL tiene un parámetro, `skip_innodb_doublewrite`, para desactivar el buffer de doble escritura. Esta característica puede mejorar sustancialmente el rendimiento.

El buffer INSERT también es una característica especial de InnoDB. Si los bloques de índice secundarios no únicos no están en la memoria, InnoDB inserta entradas en el buffer de inserción para evitar operaciones de E/S aleatorias. Periódicamente, el buffer de inserción se fusiona en los árboles de índice secundarios de la base de datos. El buffer de inserción reduce el número de operaciones de E/S fusionando las solicitudes de E/S en el mismo bloque; las operaciones de E/S aleatorias pueden ser secuenciales. El búfer de inserción también está altamente optimizado para HDD. Tanto las escrituras secuenciales como las lecturas se producen durante las operaciones normales.

Los segmentos de deshacer están orientados a E/S aleatorias. Para garantizar la concurrencia multiversión (MVCC), InnoDB debe registrar imágenes antiguas en los segmentos de deshacer. La lectura de imágenes anteriores de los segmentos de deshacer requiere lecturas aleatorias. Si ejecuta una transacción larga con lecturas repetibles (como `mysqldump`, transacción única) o ejecuta una consulta larga, pueden producirse lecturas aleatorias. Por lo tanto, almacenar segmentos de deshacer en SSD es mejor en esta instancia. Si ejecuta sólo transacciones o consultas cortas, las lecturas aleatorias no suponen un problema.

NetApp recomienda el siguiente diseño de almacenamiento debido a las características de E/S de InnoDB.



- Un volumen para almacenar archivos de MySQL orientados a I/O aleatorios y secuenciales
- Otro volumen para almacenar archivos de MySQL orientados a I/O puramente secuenciales

Este diseño también le ayuda a diseñar políticas y estrategias de protección de datos.

Parámetros de configuración de MySQL

NetApp recomienda algunos parámetros de configuración MySQL importantes para obtener un rendimiento óptimo.

Parámetros	Valores
<code>innodb_log_file_size</code>	256M
<code>innodb_flush_log_at_trx_commit</code>	2
<code>innodb_doublewrite</code>	0
<code>innodb_flush_method</code>	<code>fsync</code>
<code>innodb_buffer_pool_size</code>	11G
<code>innodb_io_capacity</code>	8192
<code>innodb_buffer_pool_instances</code>	8
<code>innodb_lru_scan_profund</code>	8192
<code>open_file_limit</code>	65535

Para establecer los parámetros descritos en esta sección, debe cambiarlos en el archivo de configuración de MySQL (my.cnf). Las mejores prácticas de NetApp se deben a las pruebas que se realizan internamente.

innodb_log_file_size

Seleccionar el tamaño correcto para el tamaño del archivo de registro InnoDB es importante para las operaciones de escritura y para tener un tiempo de recuperación decente después de un fallo del servidor.

Dado que hay tantas transacciones registradas en el archivo, el tamaño del archivo de registro es importante para las operaciones de escritura. Cuando se modifican los registros, el cambio no se vuelve a escribir inmediatamente en el tablespace. En su lugar, el cambio se registra al final del archivo de registro y la página se marca como sucia. InnoDB utiliza su registro para convertir las operaciones de I/O aleatorias en operaciones de I/O secuenciales

Cuando el log está lleno, la página sucia se escribe en el tablespace en secuencia para liberar espacio en el archivo log. Por ejemplo, supongamos que un servidor se bloquea en medio de una transacción y que las operaciones de escritura solo se registran en el archivo de registro. Antes de que el servidor pueda volver a activarse, debe pasar por una fase de recuperación en la que se reproduzcan los cambios registrados en el archivo de registro. Cuantas más entradas haya en el archivo de registro, más tiempo tardará el servidor en recuperarse.

En este ejemplo, el tamaño del archivo log afecta tanto al tiempo de recuperación como al rendimiento de escritura. Al elegir el número correcto para el tamaño del archivo log, equilibre el tiempo de recuperación con respecto al rendimiento de escritura. Normalmente, cualquier cosa entre 128M y 512M es una buena relación calidad-precio.

innodb_flush_log_at_trx_commit

Cuando se produce un cambio en los datos, este no se escribe inmediatamente en el almacenamiento.

En su lugar, los datos se registran en un buffer de log, que es una parte de la memoria que InnoDB asigna a los cambios de buffer que se registran en el archivo log. InnoDB vacía el buffer en el archivo log cuando se confirma una transacción, cuando el buffer se llena, o una vez por segundo, lo que ocurra primero. La variable de configuración que controla este proceso es `innodb_flush_log_at_trx_commit`. Las opciones de valor incluyen:

- Cuando lo ajuste `innodb_flush_log_trx_at_commit=0`, InnoDB escribe los datos modificados (en el grupo de buffers de InnoDB) en el archivo log (`ib_logfile`) y vacía el archivo log (escribir en almacenamiento) cada segundo. Sin embargo, no hace nada cuando se confirma la transacción. Si hay un fallo de alimentación o un bloqueo del sistema, ninguno de los datos sin vaciar es recuperable, ya que no se escribe en el archivo de registro o en las unidades.
- Cuando lo ajuste `innodb_flush_log_trx_commit=1`, InnoDB escribe el buffer de log en el log de transacciones y se vacía en un almacenamiento duradero para cada transacción. Por ejemplo, para todas las confirmaciones de transacciones, InnoDB escribe en el log y, a continuación, escribe en el almacenamiento. Un almacenamiento más lento afecta negativamente al rendimiento; por ejemplo, se reduce el número de transacciones InnoDB por segundo.
- Cuando lo ajuste `innodb_flush_log_trx_commit=2`, InnoDB escribe el buffer de log en el archivo log en cada confirmación; sin embargo, no escribe datos en el almacenamiento. InnoDB vacía los datos una vez cada segundo. Incluso si hay un fallo de alimentación o un fallo del sistema, los datos de la opción 2

están disponibles en el archivo de registro y son recuperables.

Si el rendimiento es el objetivo principal, establezca el valor en 2. Como InnoDB escribe en las unidades una vez por segundo, no por cada confirmación de transacción, el rendimiento mejora drásticamente. Si se produce un fallo en el suministro eléctrico o un fallo, los datos se pueden recuperar del registro de transacciones.

Si la seguridad de los datos es el objetivo principal, establezca el valor en 1 para que, para cada confirmación de transacción, InnoDB se vacíe en las unidades. Sin embargo, el rendimiento puede verse afectado.



NetApp recomienda Establecer el valor `innodb_flush_log_trx_commit` en 2 para un mejor rendimiento.

innodb_doublewrite

Cuando `innodb_doublewrite` Está activado (por defecto), InnoDB almacena todos los datos dos veces: Primero en el buffer de doble escritura y luego en los archivos de datos reales.

Es posible desactivar este parámetro con `--skip-innodb_doublewrite` para pruebas de rendimiento o cuando le preocupa más el rendimiento superior que la integridad de los datos o posibles fallos. InnoDB utiliza una técnica de vaciado de archivos denominada doble escritura. Antes de escribir páginas en los archivos de datos, InnoDB las escribe en un área contigua denominada buffer de doble escritura. Una vez que se hayan completado las operaciones de escritura y vaciado en el buffer de doble escritura, InnoDB escribe las páginas en sus posiciones adecuadas en el archivo de datos. Si el sistema operativo o un proceso `mysqld` se bloquea durante la escritura de una página, InnoDB puede encontrar más tarde una buena copia de la página desde el buffer de doble escritura durante la recuperación de fallos.



NetApp recomienda deshabilitar el buffer de doble escritura. NVRAM de ONTAP sirve la misma función. El almacenamiento en búfer doble dañará innecesariamente el rendimiento.

innodb_buffer_pool_size

El pool de buffers de InnoDB es la parte más importante de cualquier actividad de ajuste.

InnoDB depende en gran medida del pool de buffers para almacenar en caché los índices y remar los datos, el índice hash adaptativo, el buffer INSERT y muchas otras estructuras de datos utilizadas internamente. El pool de búfer también almacena en búfer los cambios en los datos para que las operaciones de escritura no sean necesarias inmediatamente en el almacenamiento, lo que mejora el rendimiento. El pool de buffers es una parte integral de InnoDB y su tamaño debe ajustarse en consecuencia. Tenga en cuenta los siguientes factores al definir el tamaño del pool de buffers:

- Para una máquina exclusiva de InnoDB, establezca el tamaño del pool de buffers en 80% o más de RAM disponible.
- Si no es un servidor dedicado de MySQL, establezca el tamaño en 50% de RAM.

innodb_flush_method

El parámetro `innodb_flush_method` especifica cómo InnoDB abre y vacía los archivos log y de datos.

Optimizaciones

En la optimización de InnoDB, la configuración de este parámetro modifica el rendimiento de la base de datos cuando es aplicable.

Las siguientes opciones son para vaciar los archivos a través de InnoDB:

- `fsync`. InnoDB utiliza el `fsync()` llamada del sistema para vaciar los archivos de datos y de registro. Esta opción es el valor predeterminado.
- `O_DSYNC`. InnoDB utiliza el `O_DSYNC` opción para abrir y vaciar los archivos de registro y `fsync()` para vaciar los archivos de datos. InnoDB no utiliza `O_DSYNC` Directamente, porque ha habido problemas con él en muchas variedades de UNIX.
- `O_DIRECT`. InnoDB utiliza el `O_DIRECT` opción (`o_directio()` En Solaris) para abrir los archivos de datos y usos `fsync()` para vaciar los archivos de datos y de registro. Esta opción está disponible en algunas versiones de GNU/Linux, FreeBSD y Solaris.
- `O_DIRECT_NO_FSYNC`. InnoDB utiliza el `O_DIRECT` Durante el vaciado de I/O; sin embargo, omite el `fsync()` llamada del sistema posterior. Esta opción no es adecuada para algunos tipos de sistemas de archivos (por ejemplo, XFS). Si no está seguro de si su sistema de archivos requiere un `fsync()` llamada al sistema (por ejemplo, para conservar todos los metadatos del archivo), utilice el `O_DIRECT` en su lugar.

Observación

En las pruebas de laboratorio de NetApp, el `fsync` La opción predeterminada se utilizó en NFS y SAN, y fue un gran improvisador de rendimiento `O_DIRECT`. Mientras se utiliza el método de vaciado como `O_DIRECT` Con ONTAP, hemos observado que el cliente escribe muchas escrituras de un solo byte en el borde del bloque de 4096 KB de forma en serie. Estas escrituras aumentan la latencia en la red y el rendimiento disminuye.

innodb_io_capacity

En el plug-in InnoDB, se agregó un nuevo parámetro llamado `innodb_io_capacity` desde MySQL 5.7.

Controla el número máximo de IOPS que realiza InnoDB (lo que incluye la tasa de vaciado de páginas desfasadas y el tamaño de lote [`ibuf`] del buffer de inserción). El parámetro `innodb_io_capacity` define un límite superior de IOPS por tareas en segundo plano de InnoDB, como vaciar páginas del pool de buffers y fusionar datos del buffer de cambios.

Defina el parámetro `innodb_io_capacity` en el número aproximado de operaciones de E/S que el sistema puede realizar por segundo. Lo ideal es mantener la configuración lo más baja posible, pero no tan baja que las actividades en segundo plano se ralenticen. Si el valor es demasiado alto, los datos se eliminan del pool de buffers e insertan el buffer demasiado rápido para que el almacenamiento en caché proporcione una ventaja significativa.



NetApp recomienda que si utiliza esta configuración en NFS, analice el resultado de la prueba de IOPS (SysBench/fio) y establezca el parámetro en consecuencia. Utilice el valor más pequeño posible para vaciar y depurar para mantener el ritmo a menos que vea más páginas modificadas o sucias de las que desee en el pool de buffers de InnoDB.



No utilice valores extremos como 20.000 o más a menos que haya demostrado que los valores más bajos no son suficientes para su carga de trabajo.

El parámetro `InnoDB_IO_CAPACITY` regula las tasas de vaciado y la E/S relacionada



Puede dañar seriamente el rendimiento si se configura este parámetro o el parámetro `innodb_io_capacity_max` demasiado alto y se desperdician las operaciones de I/O con vaciado prematuro.

`innodb_lru_scan_profund`

La `innodb_lru_scan_depth` El parámetro influye en los algoritmos y la heurística de la operación de vaciado para el pool de buffers de InnoDB.

Este parámetro es principalmente interesante para los expertos en rendimiento que ajustan las cargas de trabajo con un gran volumen de I/O. Para cada instancia de pool de buffers, este parámetro especifica hasta qué punto en la lista de páginas de uso menos reciente (LRU) el thread del limpiador de páginas debe continuar escaneando, buscando páginas sucias para vaciar. Esta operación en segundo plano se realiza una vez por segundo.

Puede ajustar el valor hacia arriba o hacia abajo para minimizar el número de páginas libres. No establezca el valor mucho más alto de lo necesario, ya que los escaneos pueden tener un costo de rendimiento significativo. Además, considere ajustar este parámetro al cambiar el Núm. De instancias del pool de buffers, porque `innodb_lru_scan_depth * innodb_buffer_pool_instances` define la cantidad de trabajo que realiza el thread de limpieza de páginas cada segundo.

Una configuración inferior a la predeterminada es adecuada para la mayoría de las cargas de trabajo. Considere aumentar el valor solo si cuenta con capacidad de I/O de reserva con una carga de trabajo típica. Por el contrario, si una carga de trabajo con gran cantidad de escritura satura la capacidad de E/S, disminuya el valor, especialmente si tiene un pool de buffers grande.

`open_file_limits`

La `open_file_limits` el parámetro determina el número de archivos que el sistema operativo permite que `mysqld` abra.

El valor de este parámetro en tiempo de ejecución es el valor real permitido por el sistema y puede ser diferente del valor especificado al iniciar el servidor. El valor es 0 en sistemas donde MySQL no puede cambiar el número de archivos abiertos. Eficaz `open_files_limit` el valor se basa en el valor especificado en el inicio del sistema (si lo hay) y en los valores de `max_connections` y `table_open_cache` mediante el uso de estas fórmulas:

- $10 + \text{max_connections} + (\text{table_open_cache} \cdot 2)$
- `max_connections` 5
- Límite del sistema operativo si es positivo
- Si el límite del sistema operativo es infinito: `open_files_limit` el valor se especifica en el inicio; 5.000 si no hay ninguno

El servidor intenta obtener el número de descriptores de archivo utilizando el máximo de estos cuatro valores.

Si no se pueden obtener muchos descriptores, el servidor intenta obtener tantos como el sistema permita.

Información de copyright

Copyright © 2024 NetApp, Inc. Todos los derechos reservados. Imprimido en EE. UU. No se puede reproducir este documento protegido por copyright ni parte del mismo de ninguna forma ni por ningún medio (gráfico, electrónico o mecánico, incluidas fotocopias, grabaciones o almacenamiento en un sistema de recuperación electrónico) sin la autorización previa y por escrito del propietario del copyright.

El software derivado del material de NetApp con copyright está sujeto a la siguiente licencia y exención de responsabilidad:

ESTE SOFTWARE LO PROPORCIONA NETAPP «TAL CUAL» Y SIN NINGUNA GARANTÍA EXPRESA O IMPLÍCITA, INCLUYENDO, SIN LIMITAR, LAS GARANTÍAS IMPLÍCITAS DE COMERCIALIZACIÓN O IDONEIDAD PARA UN FIN CONCRETO, CUYA RESPONSABILIDAD QUEDA EXIMIDA POR EL PRESENTE DOCUMENTO. EN NINGÚN CASO NETAPP SERÁ RESPONSABLE DE NINGÚN DAÑO DIRECTO, INDIRECTO, ESPECIAL, EJEMPLAR O RESULTANTE (INCLUYENDO, ENTRE OTROS, LA OBTENCIÓN DE BIENES O SERVICIOS SUSTITUTIVOS, PÉRDIDA DE USO, DE DATOS O DE BENEFICIOS, O INTERRUPTIÓN DE LA ACTIVIDAD EMPRESARIAL) CUALQUIERA SEA EL MODO EN EL QUE SE PRODUJERON Y LA TEORÍA DE RESPONSABILIDAD QUE SE APLIQUE, YA SEA EN CONTRATO, RESPONSABILIDAD OBJETIVA O AGRAVIO (INCLUIDA LA NEGLIGENCIA U OTRO TIPO), QUE SURJAN DE ALGÚN MODO DEL USO DE ESTE SOFTWARE, INCLUSO SI HUBIEREN SIDO ADVERTIDOS DE LA POSIBILIDAD DE TALES DAÑOS.

NetApp se reserva el derecho de modificar cualquiera de los productos aquí descritos en cualquier momento y sin aviso previo. NetApp no asume ningún tipo de responsabilidad que surja del uso de los productos aquí descritos, excepto aquello expresamente acordado por escrito por parte de NetApp. El uso o adquisición de este producto no lleva implícita ninguna licencia con derechos de patente, de marcas comerciales o cualquier otro derecho de propiedad intelectual de NetApp.

Es posible que el producto que se describe en este manual esté protegido por una o más patentes de EE. UU., patentes extranjeras o solicitudes pendientes.

LEYENDA DE DERECHOS LIMITADOS: el uso, la copia o la divulgación por parte del gobierno están sujetos a las restricciones establecidas en el subpárrafo (b)(3) de los derechos de datos técnicos y productos no comerciales de DFARS 252.227-7013 (FEB de 2014) y FAR 52.227-19 (DIC de 2007).

Los datos aquí contenidos pertenecen a un producto comercial o servicio comercial (como se define en FAR 2.101) y son propiedad de NetApp, Inc. Todos los datos técnicos y el software informático de NetApp que se proporcionan en este Acuerdo tienen una naturaleza comercial y se han desarrollado exclusivamente con fondos privados. El Gobierno de EE. UU. tiene una licencia limitada, irrevocable, no exclusiva, no transferible, no sublicenciable y de alcance mundial para utilizar los Datos en relación con el contrato del Gobierno de los Estados Unidos bajo el cual se proporcionaron los Datos. Excepto que aquí se disponga lo contrario, los Datos no se pueden utilizar, desvelar, reproducir, modificar, interpretar o mostrar sin la previa aprobación por escrito de NetApp, Inc. Los derechos de licencia del Gobierno de los Estados Unidos de América y su Departamento de Defensa se limitan a los derechos identificados en la cláusula 252.227-7015(b) de la sección DFARS (FEB de 2014).

Información de la marca comercial

NETAPP, el logotipo de NETAPP y las marcas que constan en <http://www.netapp.com/TM> son marcas comerciales de NetApp, Inc. El resto de nombres de empresa y de producto pueden ser marcas comerciales de sus respectivos propietarios.