Configuración del almacenamiento

Enterprise applications

NetApp
April 02, 2025

Tabla de contenidos

Configuración del almacenamiento

Descripción general

La combinación de las soluciones de almacenamiento de ONTAP y Microsoft SQL Server permite diseñar un almacenamiento de base de datos de nivel empresarial que puede satisfacer los requisitos de aplicaciones más exigentes de hoy en día.

La optimización de una solución SQL Server en ONTAP requiere comprender el patrón y las características de E/S de SQL Server. Una buena distribución del almacenamiento para una base de datos de SQL Server debe soportar los requisitos de rendimiento de SQL Server y proporcionar la máxima capacidad de gestión de la infraestructura en su conjunto. Una buena distribución de almacenamiento también permite que la puesta en marcha inicial tenga éxito y que el entorno crezca sin problemas con el tiempo a medida que crece el negocio.

Diseño de almacenamiento de datos

Para las bases de datos de SQL Server que no utilizan SnapCenter para realizar backups, Microsoft recomienda colocar los archivos de datos y de registro en unidades independientes. Para las aplicaciones que actualizan y solicitan datos simultáneamente, el archivo de registro tiene un gran consumo de escrituras y el archivo de datos (en función de la aplicación) tiene un gran volumen de lecturas y escrituras. Para la recuperación de datos, el archivo de registro no es necesario. Por lo tanto, las solicitudes de datos pueden satisfacerse desde el archivo de datos ubicado en su propia unidad.

Cuando se crea una nueva base de datos, Microsoft recomienda especificar unidades independientes para los datos y los registros. Para mover archivos después de crear la base de datos, ésta debe desconectarse. Para obtener más recomendaciones de Microsoft, consulte "Coloque los archivos de datos y de registro en unidades separadas".

Agregados

Los agregados son los contenedores de almacenamiento de nivel más bajo para las configuraciones de almacenamiento de NetApp. Existe cierta documentación heredada en Internet, que recomienda separar las operaciones de I/O en diferentes conjuntos de unidades subyacentes. No se recomienda con ONTAP. NetApp ha realizado distintas pruebas de caracterización de las cargas de trabajo de I/O utilizando agregados compartidos y dedicados con archivos de datos y archivos de registro de transacciones separados. Las pruebas muestran que un gran agregado con más grupos RAID y unidades optimiza y mejora el rendimiento del almacenamiento y tiene mayor facilidad de gestión para los administradores por dos motivos:

- Un gran agregado hace que las funcionalidades de I/O de todas las unidades estén disponibles para todos los archivos.
- Un agregado de gran tamaño permite hacer un uso más eficiente del espacio en disco.

Para alta disponibilidad (HA), colocar la réplica síncrona secundaria de SQL Server Always On Availability Group en una máquina virtual de almacenamiento (SVM) independiente del agregado. Para fines de recuperación ante desastres, coloque la réplica asíncrona en un agregado que forma parte de un clúster de almacenamiento separado en el sitio de recuperación ante desastres, con el contenido replicado mediante la tecnología SnapMirror de NetApp. NetApp recomienda tener al menos un 10% de espacio libre disponible en un agregado para un rendimiento del almacenamiento óptimo.

Volúmenes

los volúmenes se crean y residen dentro de los agregados. Este término a veces provoca confusión porque un volumen ONTAP no es una LUN. Un volumen ONTAP es un contenedor de gestión para datos. Un volumen puede contener archivos, LUN o incluso objetos S3. Un volumen no ocupa espacio, solo se utiliza para la gestión de los datos contenidos.

Consideraciones sobre el diseño del volumen

Antes de crear un diseño de volumen de base de datos, es importante comprender cómo los patrones de I/O de SQL Server y las características varían en función de la carga de trabajo y de los requisitos de backup y recuperación. Consulte las siguientes recomendaciones de NetApp para volúmenes flexibles:

- Evite compartir volúmenes entre hosts. Por ejemplo, aunque sería posible crear 2 LUN en un único volumen y compartir cada LUN en un host diferente, esto debe evitarse porque puede complicar la gestión. En el caso de ejecutar varias instancias de SQL Server en el mismo host, a menos que esté cerca del límite de volúmenes en un nodo, evite el uso compartido de volúmenes y disponga en su lugar de un volumen separado por instancia por host para facilitar la gestión de datos.
- Utilice puntos de montaje NTFS en lugar de letras de unidad para superar la limitación de 26 unidades en Windows. Cuando se usan puntos de montaje de volumen, se recomienda generalmente asignar a la etiqueta de volumen el mismo nombre que el punto de montaje.
- Cuando sea necesario, configure una política de tamaño automático de volúmenes para ayudar a evitar condiciones de falta de espacio.
- Si instala SQL Server en un recurso compartido de SMB, asegúrese de que Unicode esté habilitado en los volúmenes de SMB para crear carpetas.
- Establezca el valor de reserva de instantáneas en el volumen en cero para facilitar la supervisión desde una perspectiva operativa.
- Deshabilite las programaciones de Snapshot y las políticas de retención. En su lugar, utilice SnapCenter para coordinar las copias Snapshot de los volúmenes de datos de SQL Server.
- Coloque las bases de datos del sistema SQL Server en un volumen dedicado.
- Tempdb es una base de datos del sistema utilizada por SQL Server como espacio de trabajo temporal, especialmente para operaciones DBCC CHECKDB intensivas en E/S. Por lo tanto, coloque esta base de datos en un volumen dedicado con un conjunto de discos separado. En entornos grandes en los que el número de volúmenes es un reto, puede consolidar tempdb en menos volúmenes y almacenarlo en el mismo volumen que otras bases de datos del sistema tras una planificación cuidadosa. La protección de datos para tempdb no es una prioridad alta porque esta base de datos se vuelve a crear cada vez que se reinicia SQL Server.
- Colocar archivos de datos de usuario (`.mdf`) en volúmenes separados porque son cargas de trabajo de lectura/escritura aleatorias. Es común crear backups de registros de transacciones con más frecuencia que los backups de bases de datos. Por este motivo, coloque los archivos de registro de transacciones (`.ldf`) en un volumen o VMDK separados de los archivos de datos para que puedan crearse programaciones de backup independientes para cada uno. Esta separación también aísla la E/S de escritura secuencial de los archivos de registro de la E/S de lectura/escritura aleatoria de los archivos de datos y mejora significativamente el rendimiento de SQL Server.

LUN

- Asegúrese de que los archivos de la base de datos del usuario y el directorio de registro para almacenar backup de registros se encuentren en volúmenes independientes para evitar que la política de retención sobrescriba las snapshots cuando estas se usen con la tecnología SnapVault.

- No mezcle archivos de base de datos ni de otro tipo, como archivos relacionados con la búsqueda de texto completo, en el mismo LUN.
- La colocación de archivos secundarios de base de datos (como parte de un grupo de archivos) en volúmenes distintos mejora el rendimiento de la base de datos de SQL Server. Esta separación solo es válida si el archivo de la base de datos `.mdf` no comparte su LUN con ningún otro `.mdf` archivo.
- Si crea LUN con DiskManager u otras herramientas, asegúrese de que el tamaño de unidad de asignación esté establecido en 64K para las particiones al formatear las LUN.
- Consulte "Microsoft Windows e MPIO nativo bajo las prácticas recomendadas de ONTAP para SAN moderna" Para aplicar la compatibilidad con accesos múltiples en Windows a dispositivos iSCSI en las propiedades MPIO.

Archivos de base de datos y grupos de archivos

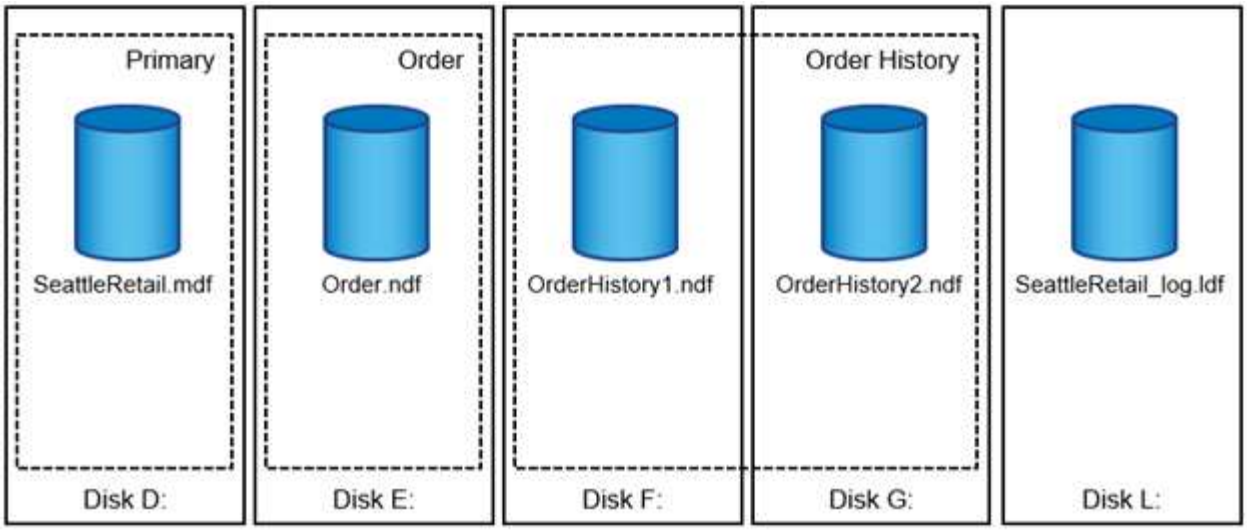
La ubicación correcta del archivo de la base de datos de SQL Server en ONTAP es crítica durante la etapa de la implementación inicial. Esto garantiza un rendimiento óptimo, gestión del espacio, tiempos de backup y restauración que pueden configurarse para que se ajusten a sus necesidades empresariales.

En teoría, SQL Server (64 bits) admite 32.767 bases de datos por instancia y 524.272TB GB de tamaño de base de datos, aunque la instalación típica suele tener varias bases de datos. Sin embargo, el número de bases de datos que SQL Server puede manejar depende de la carga y el hardware. No es raro ver instancias de SQL Server que alojan docenas, cientos o incluso miles de pequeñas bases de datos.

Cada base de datos consta de uno o más archivos de datos y uno o varios archivos de registro de transacciones. El log de transacciones almacena la información sobre las transacciones de la base de datos y todas las modificaciones de datos realizadas por cada sesión. Cada vez que se modifican los datos, SQL Server almacena suficiente información en el log de transacciones para deshacer (realizar rollback) o rehacer (reproducir) la acción. Un registro de transacciones de SQL Server es una parte esencial de la reputación de SQL Server en cuanto a integridad y solidez de los datos. El registro de transacciones es vital para las capacidades de atomicidad, consistencia, aislamiento y durabilidad (ACID) de SQL Server. SQL Server escribe en el registro de transacciones tan pronto como se producen cambios en la página de datos. Cada sentencia de lenguaje de manipulación de datos (DML) (por ejemplo, SELECT, INSERT, UPDATE o DELETE) es una transacción completa, y el registro de transacciones se asegura de que se realice toda la operación basada en juegos, asegurándose de la atomicidad de la transacción.

Cada base de datos tiene un archivo de datos primario, que, por defecto, tiene la extensión .mdf. Además, cada base de datos puede tener archivos de base de datos secundarios. Esos archivos, por defecto, tienen extensiones .ndf.

Todos los archivos de base de datos se agrupan en grupos de archivos. Un grupo de archivos es la unidad lógica, que simplifica la administración de la base de datos. Permiten la separación entre la ubicación de objetos lógicos y los archivos físicos de la base de datos. Al crear las tablas de objetos de base de datos, especifique en qué grupo de archivos se deben colocar sin preocuparse por la configuración del archivo de datos subyacente.



La capacidad de colocar varios archivos de datos dentro del grupo de archivos permite distribuir la carga entre diferentes dispositivos de almacenamiento, lo que ayuda a mejorar el rendimiento de I/O del sistema. Por el contrario, el registro de transacciones no se beneficia de los varios archivos, ya que SQL Server escribe en el registro de transacciones de forma secuencial.

La separación entre la ubicación de objetos lógicos en los grupos de archivos y los archivos físicos de la base de datos le permite ajustar el diseño del archivo de la base de datos, aprovechando al máximo el subsistema de almacenamiento. La cantidad de archivos de datos que soportan una carga de trabajo maestra se puede modificar según sea necesario para admitir los requisitos de I/O y la capacidad esperada sin afectar a la aplicación. Esas variaciones en el diseño de la base de datos son transparentes para los desarrolladores de aplicaciones, que colocan los objetos de la base de datos en los grupos de archivos en lugar de en los archivos de la base de datos.

NetApp recomienda evitar el uso del grupo de archivos primario para cualquier cosa excepto objetos del sistema. La creación de un grupo de archivos independiente o un conjunto de grupos de archivos para los objetos de usuario simplifica la administración de la base de datos y la recuperación ante desastres, especialmente en el caso de bases de datos grandes.

Puede especificar el tamaño inicial del archivo y los parámetros de crecimiento automático en el momento de crear la base de datos o agregar nuevos archivos a una base de datos existente. SQL Server utiliza un algoritmo de relleno proporcional al elegir en qué archivo de datos debe escribir los datos. Escribe una cantidad de datos proporcionalmente al espacio libre disponible en los archivos. Cuanto mayor sea el espacio libre del archivo, más escrituras gestionará.

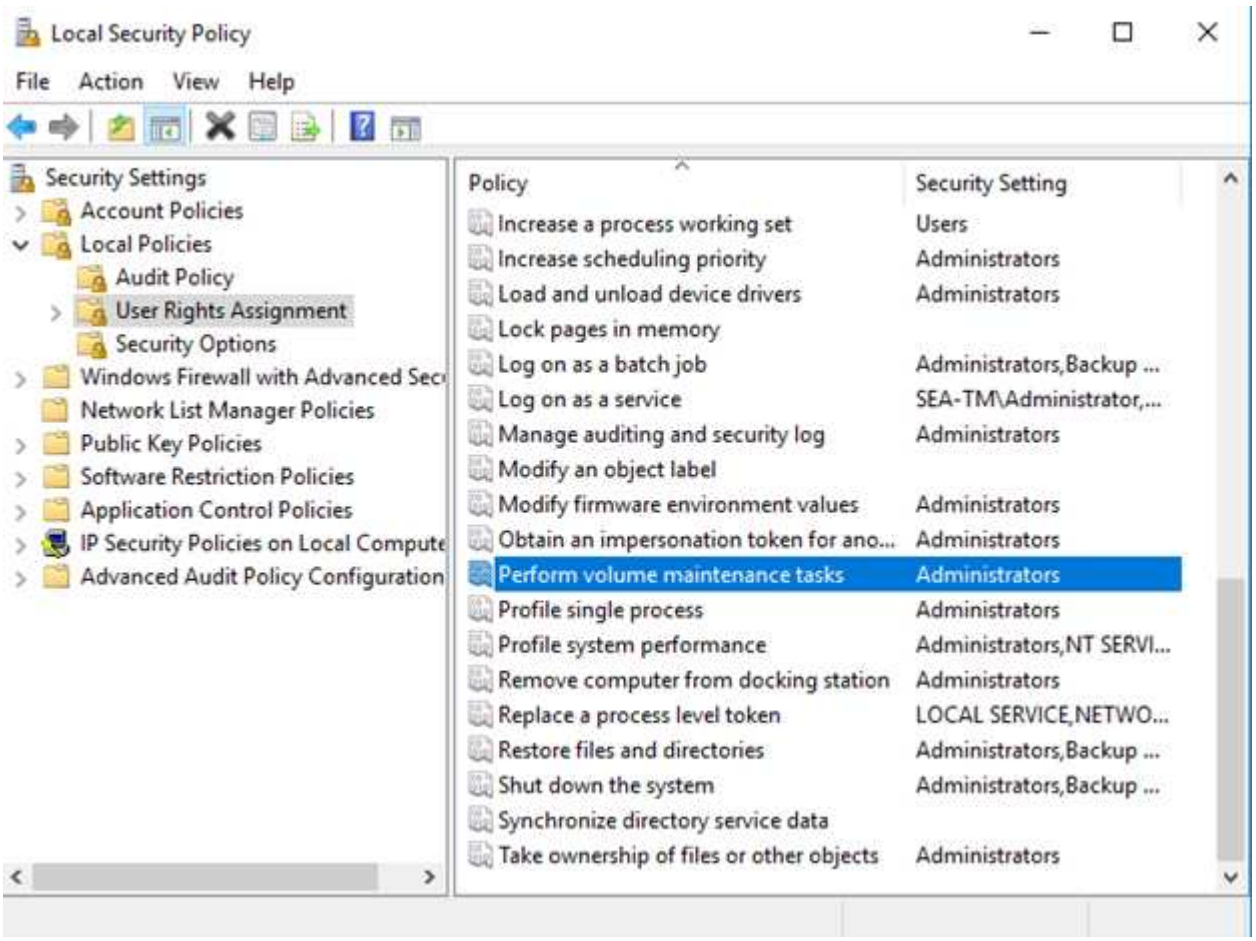
NetApp recomienda que todos los archivos en un solo grupo de archivos tengan los mismos parámetros iniciales de tamaño y crecimiento automático, con el tamaño de crecimiento definido en megabytes en lugar de porcentajes. Esto ayuda al algoritmo de relleno proporcional a equilibrar de forma uniforme las actividades de escritura en los archivos de datos.

Cada vez que SQL Server crece archivos, llena el espacio recién asignado con ceros. Este proceso bloquea todas las sesiones que necesitan escribir en el archivo correspondiente o, en caso de crecimiento del log de transacciones, generar registros de log de transacciones.

SQL Server siempre pone a cero el registro de transacciones y ese comportamiento no se puede cambiar. Sin embargo, puede controlar si los archivos de datos están a cero habilitando o deshabilitando la inicialización instantánea de archivos. La activación de la inicialización instantánea de archivos ayuda a acelerar el crecimiento de los archivos de datos y reduce el tiempo necesario para crear o restaurar la base de datos.

Un pequeño riesgo de seguridad está asociado con la inicialización instantánea de archivos. Cuando esta opción está activada, las partes no asignadas del archivo de datos pueden contener información de los archivos del sistema operativo eliminados anteriormente. Los administradores de bases de datos pueden examinar estos datos.

Puede activar la inicialización instantánea de archivos agregando el permiso SA_MANAGE_VOLUME_NAME, también conocido como “Realizar tarea de mantenimiento de volúmenes”, a la cuenta de inicio de SQL Server. Puede hacerlo en la aplicación de gestión de políticas de seguridad local (secpol.msc), como se muestra en la siguiente figura. Abra las propiedades del permiso “Realizar tarea de mantenimiento de volúmenes” y agregue la cuenta de inicio de SQL Server a la lista de usuarios allí.



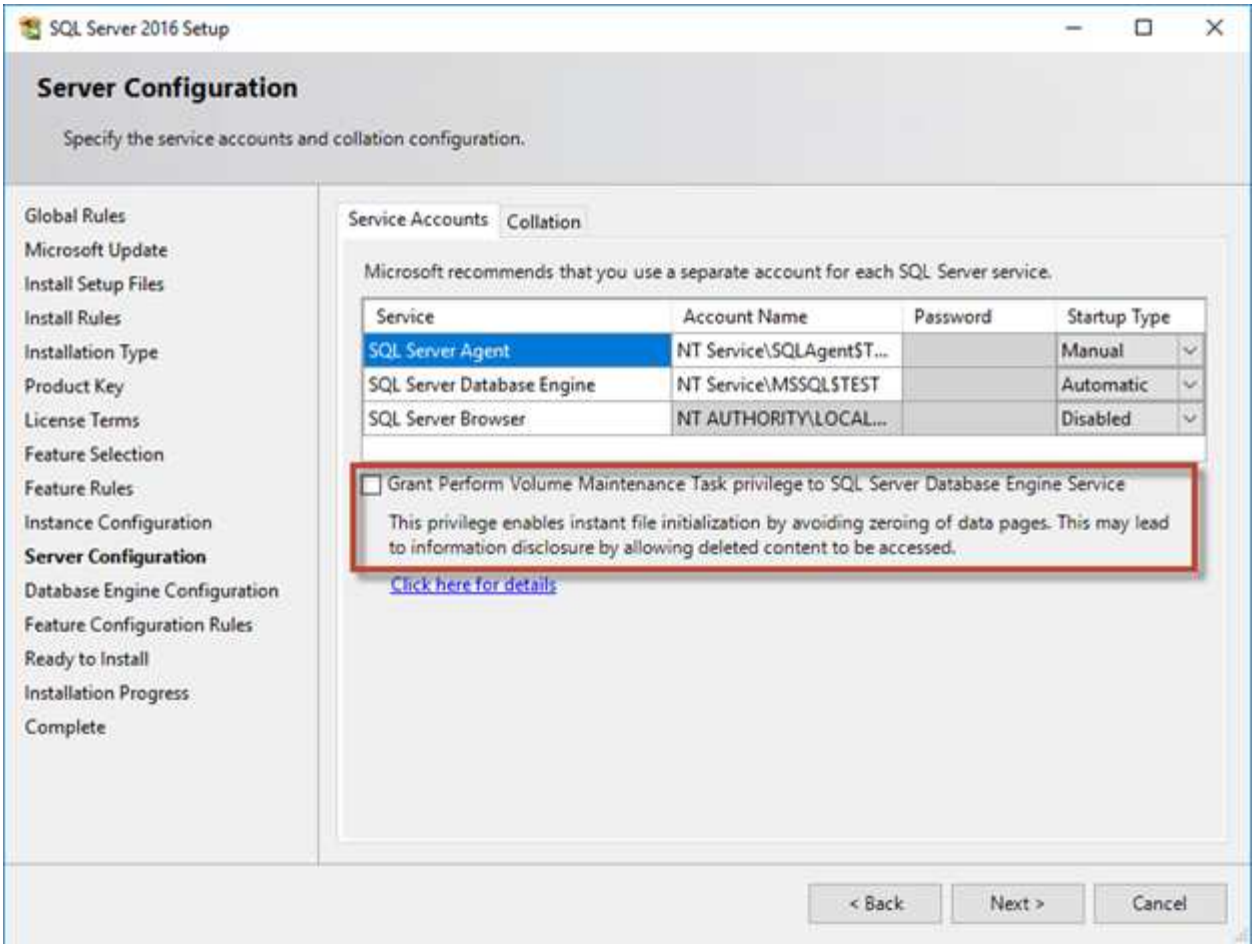
Para comprobar si el permiso está habilitado, puede utilizar el código del siguiente ejemplo. Este código establece dos indicadores de rastreo que obligan a SQL Server a escribir información adicional en el log de errores, crear una base de datos pequeña y leer el contenido del log.

```
DBCC TRACEON(3004,3605,-1)
GO
CREATE DATABASE DelMe
GO
EXECUTE sp_readerrorlog
GO
DROP DATABASE DelMe
GO
DBCC TRACEOFF(3004,3605,-1)
GO
```

Cuando la inicialización instantánea de archivos no está activada, el registro de errores de SQL Server muestra que SQL Server pone a cero el archivo de datos de mdf además de poner a cero el archivo log ldf, como se muestra en el siguiente ejemplo. Cuando se activa la inicialización instantánea del archivo, sólo se muestra la puesta a cero del archivo log.

	LogDate	ProcessInfo	Text
365	2017-02-09 08:10:07.660	spid53	Ckpt dbid 3 flush delta counts.
366	2017-02-09 08:10:07.660	spid53	Ckpt dbid 3 logging active xact info.
367	2017-02-09 08:10:07.750	spid53	Ckpt dbid 3 phase 1 ended (8)
368	2017-02-09 08:10:07.750	spid53	About to log Checkpoint end.
369	2017-02-09 08:10:07.880	spid53	Ckpt dbid 3 complete
370	2017-02-09 08:10:08.130	spid53	Starting up database 'DelMe'.
371	2017-02-09 08:10:08.150	spid53	FixupLogTail(progress) zeroing C:\Program Files\Micros
372	2017-02-09 08:10:08.160	spid53	Zeroing C:\Program Files\Microsoft SQL Server\MSSQ
373	2017-02-09 08:10:08.170	spid53	Zeroing completed on C:\Program Files\Microsoft SQL
374	2017-02-09 08:10:08.710	spid53	Ckpt dbid 6 started
375	2017-02-09 08:10:08.710	spid53	About to log Checkpoint begin.

La tarea Realizar mantenimiento de volumen se simplifica en SQL Server 2016 y, posteriormente, se proporciona como opción durante el proceso de instalación. Esta figura muestra la opción para otorgar al servicio del motor de base de datos SQL Server el privilegio para realizar la tarea de mantenimiento de volúmenes.



Otra opción importante de la base de datos que controla los tamaños de los archivos de la base de datos es la reducción automática. Cuando esta opción está habilitada, SQL Server reduce regularmente los archivos de la base de datos, reduce su tamaño y libera espacio al sistema operativo. Esta operación consume muchos recursos y rara vez es útil porque los archivos de la base de datos vuelven a crecer después de un tiempo cuando llegan nuevos datos al sistema. La reducción automática no debe estar activada en la base de datos.

Directorio de registro

El directorio de registro se especifica en SQL Server para almacenar datos de backup de registros de transacciones en el nivel de host. Si utiliza SnapCenter para realizar backup de archivos de registro, cada host SQL Server que utiliza SnapCenter debe tener un directorio de registro de host configurado para realizar backups de registros. SnapCenter tiene un repositorio de base de datos, por lo que los metadatos relacionados con las operaciones de backup, restauración o clonado se almacenan en un repositorio de base de datos central.

Los tamaños del directorio de registro de host se calculan de la siguiente manera:
Tamaño del directorio de registro de host = ((tamaño máximo de LDF de base de datos x tasa de cambio de registro diario %) x (retención de instantánea) ÷ (1: Porcentaje de espacio de sobrecarga de LUN)
La fórmula de ajuste de tamaño del directorio de registro del host asume una sobrecarga del 10% de las LUN

Coloque el directorio de registro en un volumen o LUN dedicados. La cantidad de datos en el directorio de registro del host depende del tamaño de los backups y de la cantidad de días que se retienen los backups. SnapCenter solo permite un directorio de registro de host por cada host SQL Server. Puede configurar los

directorios de registro de host en SnapCenter -→ Host -→ Configurar el plugin.

NetApp recomienda lo siguiente para un directorio de registro de host:

- Asegúrese de que el directorio de registro de host no esté compartido por ningún otro tipo de datos que pueda dañar los datos de la instantánea de backup.
- No coloque bases de datos de usuario ni bases de datos del sistema en un LUN que aloje puntos de montaje.
- Cree el directorio de registro de host en un volumen dedicado al cual SnapCenter copia los registros de transacciones.
- Utilice los asistentes de SnapCenter para migrar bases de datos al almacenamiento NetApp de modo que las bases de datos se almacenen en ubicaciones válidas, lo que permite realizar correctamente las operaciones de backup y restauración de SnapCenter. Tenga en cuenta que el proceso de migración es disruptivo y puede provocar que las bases de datos se desconecten mientras se realiza la migración.
- Deben establecerse las siguientes condiciones para las instancias de clúster de conmutación por error (FCI) de SQL Server:
  - Si va a utilizar una instancia de clúster de conmutación al nodo de respaldo, el LUN del directorio de registro de host debe ser un recurso de disco de clúster en el mismo grupo de clústeres que la instancia de SQL Server que se va a realizar el backup de SnapCenter.
  - Si utiliza una instancia de clúster de conmutación al nodo de respaldo, las bases de datos de usuario deben colocarse en LUN compartidos que sean recursos de clúster de discos físicos asignados al grupo de clústeres asociado con la instancia de SQL Server.

Archivos tempdb

La base de datos Tempdb se puede utilizar en gran medida. Además de la ubicación óptima de los archivos de base de datos de usuario en ONTAP, la ubicación de los archivos de datos tempdb también es crítica para reducir la contención de asignación. Tempdb debe colocarse en un disco independiente y no compartirse con los archivos de datos de usuario.

La contención de página se puede producir en las páginas de mapa de asignación global (GAM), mapa de asignación global compartida (SGAM) o espacio libre de página (PFS) cuando SQL Server debe escribir en páginas especiales del sistema para asignar nuevos objetos. Los pestillos bloquean estas páginas en la memoria. En una instancia de SQL Server ocupada, puede tardar mucho tiempo en obtener un bloqueo en una página del sistema en tempdb. Esto da como resultado tiempos de ejecución de consultas más lentos y se conoce como contención de bloqueo interno. Consulte las siguientes prácticas recomendadas para crear archivos de datos tempdb:

- Para < or = a 8 núcleos: Archivos de datos tempdb = número de núcleos
- Para > 8 núcleos: 8 archivos de datos tempdb
- El archivo de datos tempdb se debe crear con el mismo tamaño

El siguiente script de ejemplo modifica tempdb creando ocho archivos tempdb de igual tamaño y moviendo tempdb al punto de montaje `C:\MSSQL\tempdb` para SQL Server 2012 y posterior.

```
use master

go

-- Change logical tempdb file name first since SQL Server shipped with
logical file name called tempdev

alter database tempdb modify file (name = 'tempdev', newname =
'tempdev01');



-- Change location of tempdev01 and log file

alter database tempdb modify file (name = 'tempdev01', filename =
'C:\MSSQL\tempdb\tempdev01.mdf');

alter database tempdb modify file (name = 'templog', filename =
'C:\MSSQL\tempdb\templog.ldf');

GO

-- Assign proper size for tempdev01

ALTER DATABASE [tempdb] MODIFY FILE ( NAME = N'tempdev01', SIZE = 10GB );

ALTER DATABASE [tempdb] MODIFY FILE ( NAME = N'templog', SIZE = 10GB );

GO

-- Add more tempdb files

ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdev02', FILENAME =
N'C:\MSSQL\tempdb\tempdev02.ndf' , SIZE = 10GB , FILEGROWTH = 10%);

ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdev03', FILENAME =
N'C:\MSSQL\tempdb\tempdev03.ndf' , SIZE = 10GB , FILEGROWTH = 10%);

ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdev04', FILENAME =
N'C:\MSSQL\tempdb\tempdev04.ndf' , SIZE = 10GB , FILEGROWTH = 10%);

ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdev05', FILENAME =
N'C:\MSSQL\tempdb\tempdev05.ndf' , SIZE = 10GB , FILEGROWTH = 10%);

ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdev06', FILENAME =
N'C:\MSSQL\tempdb\tempdev06.ndf' , SIZE = 10GB , FILEGROWTH = 10%);
```

```
ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdev07', FILENAME =
N'C:\MSSQL\tempdb\tempdev07.ndf' , SIZE = 10GB , FILEGROWTH = 10%);

ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdev08', FILENAME =
N'C:\MSSQL\tempdb\tempdev08.ndf' , SIZE = 10GB , FILEGROWTH = 10%);

GO
```

A partir de SQL Server 2016, el número de núcleos de CPU visibles para el sistema operativo se detecta automáticamente durante la instalación y, en función de ese número, SQL Server calcula y configura el número de archivos tempdb necesarios para un rendimiento óptimo.

Eficiencia del almacenamiento

La eficiencia del almacenamiento de ONTAP está optimizada para almacenar y gestionar datos de SQL Server de una manera que consuma la menor cantidad de espacio de almacenamiento sin que ello afecte al rendimiento.

Las funciones de eficiencia del espacio, como la compresión, la compactación y la deduplicación están diseñadas para aumentar la cantidad de datos lógicos que se adaptan a una determinada cantidad de almacenamiento físico. El resultado es una reducción de los costes y los gastos generales de gestión.

En un nivel superior, la compresión es un proceso matemático por el cual los patrones en los datos se detectan y codifican de manera que reducen los requisitos de espacio. Por el contrario, la deduplicación detecta bloques de datos repetidos y elimina las copias externas. La compactación permite que varios bloques lógicos de datos compartan el mismo bloque físico en medios.

> Consulte las siguientes secciones sobre thin provisioning para obtener una explicación de la interacción entre la eficiencia del almacenamiento y la reserva fraccionaria.

Compresión

Antes de la disponibilidad de sistemas de almacenamiento all-flash, la compresión basada en cabinas era de un valor limitado debido a que la mayoría de las cargas de trabajo con un gran volumen de I/O requerían un gran número de discos para proporcionar un rendimiento aceptable. Los sistemas de almacenamiento contenían invariablemente mucha más capacidad de la necesaria como efecto secundario al gran número de unidades. La situación ha cambiado con el aumento del almacenamiento de estado sólido. Ya no es necesario sobreaprovisionar enormemente las unidades solo para obtener un buen rendimiento. El espacio de las unidades de un sistema de almacenamiento puede coincidir con las necesidades de capacidad reales.

La mayor funcionalidad de IOPS de las unidades de estado sólido (SSD) casi siempre genera ahorro de costes en comparación con las unidades giratorias, pero la compresión puede conseguir un mayor ahorro al aumentar la capacidad efectiva de los medios de estado sólido.

Existen varias formas de comprimir datos. Muchas bases de datos incluyen sus propias funcionalidades de compresión, pero esto se observa muy rara vez en los entornos del cliente. La razón suele ser la penalización de rendimiento para un **cambio** a los datos comprimidos, además con algunas aplicaciones hay altos costos de licencia para la compresión a nivel de base de datos. Por último, existen las consecuencias de rendimiento generales para las operaciones de base de datos. Tiene poco sentido pagar un alto coste de licencia por CPU por una CPU que realiza compresión y descompresión de datos en lugar de trabajo real de base de datos. Una mejor opción es descargar el trabajo de compresión en el sistema de almacenamiento.

Compresión adaptativa

La compresión adaptativa se ha probado minuciosamente en cargas de trabajo empresariales sin que ello afecte al rendimiento, incluso en un entorno all-flash en el que la latencia se mide en microsegundos. Algunos clientes incluso han informado de un aumento del rendimiento con el uso de la compresión, ya que los datos siguen comprimidos en la caché, lo que aumenta efectivamente la cantidad de caché disponible en una controladora.

ONTAP gestiona bloques físicos en 4KB unidades. La compresión adaptativa usa un tamaño de bloque de compresión predeterminado de 8KB KB, lo que significa que los datos se comprimen en 8KB unidades. Esto coincide con el tamaño de bloque de 8KB KB que suelen utilizar las bases de datos relacionales. Los algoritmos de compresión son más eficientes a medida que se comprimen más datos como una sola unidad. Un tamaño de bloque de compresión de 32KB KB haría más eficiente el espacio que una unidad de bloques de compresión de 8KB KB. Esto significa que la compresión adaptativa con el tamaño de bloque de 8KB KB predeterminado conduce a tasas de eficiencia ligeramente más bajas, pero también ofrece una ventaja significativa si se usa un tamaño de bloque de compresión más pequeño. Las cargas de trabajo de bases de datos incluyen una gran cantidad de actividad de sobrescritura. Para sobrescribir un bloque de datos de 8KB GB de 32KB comprimido, es necesario volver a leer los 32KB TB completos de datos lógicos, descomprimirlos, actualizar la región de 8KB requerida, recomprimir y, a continuación, volver a escribir todo el 32KB en las unidades. Esta es una operación muy cara para un sistema de almacenamiento y es el motivo por el que algunas cabinas de almacenamiento de la competencia basadas en bloques de compresión más grandes también incurren en un impacto significativo en el rendimiento con las cargas de trabajo de base de datos.

> El tamaño de los bloques utilizado por la compresión adaptativa se puede aumentar hasta 32KB KB. Esto puede mejorar la eficiencia del almacenamiento y debe considerarse en el caso de archivos inactivos, como registros de transacciones y archivos de backup, cuando se almacena una cantidad sustancial de dichos datos en la cabina. En algunas situaciones, las bases de datos activas que usan un tamaño de bloque de 16KB KB o de 32KB KB también pueden beneficiarse de aumentar el tamaño de bloque de la compresión adaptativa para que coincida. Consulte a un representante de NetApp o de su partner para obtener orientación sobre si esto es adecuado para su carga de trabajo.

> Los bloques de compresión superiores a los 8KB MB no se deben usar junto a la deduplicación en destinos de backup en streaming. El motivo es que los pequeños cambios en los datos de backup afectan a la ventana de compresión de 32KB:1. Si la ventana cambia, los datos comprimidos resultantes difieren en todo el archivo. La deduplicación ocurre después de la compresión, lo que significa que el motor de deduplicación ve cada backup comprimido de forma diferente. Si se requiere la deduplicación de backups en streaming, solo deberá usarse la compresión adaptativa de 8KB bloques. Es preferible recurrir a la compresión adaptativa, ya que funciona con un tamaño de bloque más pequeño y no interrumpe la eficiencia de la deduplicación. Por motivos similares, la compresión en el lado del host también interfiere con la eficiencia de la deduplicación.

Alineación de la compresión

La compresión adaptativa en un entorno de base de datos requiere tener en cuenta algún tipo de aspecto en la alineación de bloques de compresión. Hacerlo solo es una preocupación para los datos sujetos a sobrescrituras aleatorias de bloques muy específicos. Este enfoque es similar en concepto a la alineación general del sistema de archivos, donde el inicio de un sistema de archivos debe alinearse con un límite de dispositivo 4K y el tamaño de bloque de un sistema de archivos debe ser un múltiplo de 4K.

Por ejemplo, una escritura 8KB en un archivo se comprime solo si se alinea con un límite de 8KB KB en el propio sistema de archivos. Este punto significa que debe caer en los primeros 8KB del archivo, el segundo

8KB del archivo, y así sucesivamente. La forma más sencilla de garantizar una alineación correcta es utilizar el tipo de LUN correcto, cualquier partición creada debe tener un desplazamiento desde el inicio del dispositivo que sea un múltiplo de 8K y usar un tamaño de bloque del sistema de archivos que sea un múltiplo del tamaño del bloque de la base de datos.

Los datos como los backups o los registros de transacciones son operaciones escritas secuencialmente que abarcan varios bloques, todos ellos comprimidos. Por lo tanto, no hay necesidad de considerar la alineación. El único patrón de E/S preocupante es la sobrescritura aleatoria de archivos.

Compactación de datos

La compactación de datos es una tecnología que mejora la eficiencia de la compresión. Como se ha indicado anteriormente, la compresión adaptativa por sí sola puede proporcionar un ahorro de 2:1 KB, ya que se limita a almacenar una I/O de 8KB KB en un bloque de 4KB WAFL. Los métodos de compresión con tamaños de bloque más grandes ofrecen una mejor eficiencia. Sin embargo, no son adecuados para datos sujetos a sobrescrituras de bloques pequeños. La descompresión de 32KB unidades de datos, la actualización de una parte de 8KB, la recompresión y la escritura en las unidades genera una sobrecarga.

La compactación de datos permite almacenar varios bloques lógicos en bloques físicos. Por ejemplo, una base de datos con datos altamente comprimibles, como texto o bloques parcialmente completos, puede comprimirse de 8KB a 1KB. Sin compactación, esos 1KB TB de datos seguirían ocupando un bloque completo de 4KB KB. La compactación de datos inline permite almacenar 1KB TB de datos comprimidos en solo 1KB GB de espacio físico junto con otros datos comprimidos. No es una tecnología de compresión; simplemente es una forma más eficaz de asignar espacio en las unidades y, por tanto, no debe crear un efecto de rendimiento detectable.

El grado de ahorro obtenido varía. Por lo general, los datos que ya están comprimidos o cifrados no se pueden comprimir aún más y, por lo tanto, estos conjuntos de datos no se benefician de la compactación. Por el contrario, los archivos de datos recién inicializados que contienen poco más que metadatos de bloques y ceros se comprimen hasta 80:1.

Eficiencia de almacenamiento sensible a la temperatura

La eficiencia de almacenamiento sensible a la temperatura (TSSE) está disponible en ONTAP 9, e.8 y posteriores. Se basa en mapas de calor de acceso a bloques para identificar los bloques a los que se accede con poca frecuencia y comprimirlos con mayor eficiencia.

Deduplicación

La deduplicación es eliminar los tamaños de bloques duplicados de un conjunto de datos. Por ejemplo, si existiera el mismo bloque de 4KB KB en 10 archivos diferentes, la deduplicación redirigiría ese bloque de 4KB KB en los 10 archivos al mismo bloque físico de 4KB KB. El resultado sería una mejora de 10:1 veces en eficiencia en esos datos.

Los datos, como las LUN de arranque invitado de VMware, suelen deduplicar muy bien porque constan de varias copias de los mismos archivos del sistema operativo. Se ha observado una eficiencia de 100:1 y mayor.

Algunos datos no contienen datos duplicados. Por ejemplo, un bloque de Oracle contiene una cabecera que es única globalmente para la base de datos y un cola que es casi único. Como resultado, la deduplicación de una base de datos de Oracle rara vez produce un ahorro superior al 1%. La deduplicación con bases de datos de MS SQL es ligeramente mejor, pero los metadatos únicos a nivel de bloque siguen siendo una limitación.

En pocos casos, se ha observado un ahorro de espacio de hasta un 15 % en bases de datos con 16KB KB y tamaños de bloque grandes. El primer 4KB de cada bloque contiene el encabezado único a nivel mundial, y el

último bloque de 4KB contiene el remolque casi único. Los bloques internos pueden optar a la deduplicación, aunque en la práctica esto se atribuye casi por completo a la deduplicación de datos puestos a cero.

Muchas cabinas de la competencia afirman la capacidad de deduplicar bases de datos basándose en la presunción de que una base de datos se copia varias veces. En este sentido, la deduplicación de NetApp también podría utilizarse, pero ONTAP ofrece una opción mejor: La tecnología FlexClone de NetApp. El resultado final es el mismo; se crean varias copias de una base de datos que comparten la mayoría de los bloques físicos subyacentes. El uso de FlexClone es mucho más eficiente que tomarse tiempo para copiar archivos de base de datos y después deduplicarlos. Es, de hecho, la no duplicación en lugar de la deduplicación, porque nunca se crea un duplicado.

Eficiencia y thin provisioning

Las funciones de eficiencia son formas de thin provisioning. Por ejemplo, una LUN de 100GB GB que ocupa un volumen de 100GB GB podría comprimirse hasta 50GB 000. Todavía no hay ahorros reales realizados porque el volumen sigue siendo de 100GB GB. Primero se debe reducir el volumen para que el espacio ahorrado se pueda usar en cualquier otro lugar del sistema. Si los cambios realizados en la LUN de 100GB TB más adelante hacen que los datos se puedan comprimir menos, el tamaño de la LUN aumentará y el volumen podría llenarse.

Se recomienda encarecidamente el aprovisionamiento ligero porque puede simplificar la gestión y, al mismo tiempo, proporcionar una mejora considerable en la capacidad utilizable con un ahorro de costes asociado. La razón es simple: Los entornos de bases de datos suelen incluir una gran cantidad de espacio vacío, un gran número de volúmenes y LUN, y datos comprimibles. El aprovisionamiento grueso provoca la reserva de espacio en el almacenamiento para volúmenes y LUN por si en algún momento llegan a estar llenos un 100 % y contienen un 100 % de datos que no se pueden comprimir. Es poco probable que esto ocurra. El thin provisioning permite reclamar y utilizar ese espacio en otra parte, y permite que la gestión de la capacidad se base en el propio sistema de almacenamiento en lugar de muchos volúmenes y LUN más pequeños.

Algunos clientes prefieren utilizar el aprovisionamiento pesado, ya sea para cargas de trabajo específicas o, por lo general, basándose en prácticas operativas y de adquisición establecidas.

> Si un volumen se aprovisiona en exceso, debe tenerse cuidado desactivar por completo todas las funciones de eficiencia de ese volumen, incluida la descompresión y la eliminación de la deduplicación con `sis undo` el comando. El volumen no debe aparecer en `volume efficiency show` la salida. Si lo hace, el volumen sigue estando parcialmente configurado para las funciones de eficiencia. Como resultado, la sobrescritura garantiza un funcionamiento diferente, lo que aumenta la posibilidad de que las sobretensiones de la configuración hagan que el volumen se quede sin espacio inesperadamente, lo que producirá errores de I/O de la base de datos.

Mejores prácticas de eficiencia

NetApp recomienda lo siguiente:

Valores predeterminados de AFF

Los volúmenes creados en ONTAP en un sistema AFF all-flash son thin provisioning, con todas las funciones de eficiencia inline habilitadas. Aunque por lo general, las bases de datos no se benefician de la deduplicación y pueden incluir datos que no se pueden comprimir, la configuración predeterminada es adecuada para casi todas las cargas de trabajo. ONTAP está diseñado para procesar eficientemente todo tipo de datos y patrones de I/O, independientemente de que generen o no ahorros. Los valores predeterminados solo se deben cambiar si los motivos se entienden por completo y existe un beneficio para desviarse.

Recomendaciones generales

- Si los volúmenes o LUN no son con thin provisioning, debe deshabilitar todas las configuraciones de eficiencia, ya que el uso de estas funciones no proporciona ahorro y la combinación de aprovisionamiento grueso con la eficiencia de espacio habilitada puede provocar un comportamiento inesperado, incluidos errores de falta de espacio.
- Si los datos no están sujetos a sobrescrituras, como con backups o registros de transacciones de base de datos, puede lograr una mayor eficiencia habilitando TSSE con un bajo período de enfriamiento.
- Es posible que algunos archivos contengan una cantidad significativa de datos que no se puedan comprimir, por ejemplo, cuando la compresión ya está activada en el nivel de aplicación de los archivos está cifrada. Si se da alguna de estas situaciones, considere la posibilidad de deshabilitar la compresión para permitir un funcionamiento más eficiente en otros volúmenes que contengan datos comprimibles.
- No utilice la compresión 32KB ni la deduplicación con backups de bases de datos. Consulte la sección Compresión adaptativa para obtener más detalles.

Compresión de base de datos

SQL Server en sí también tiene funciones para comprimir y gestionar datos de forma eficiente. SQL Server soporta actualmente dos tipos de compresión de datos: Compresión de filas y compresión de páginas.

La compresión de filas cambia el formato de almacenamiento de datos. Por ejemplo, cambia los enteros y decimales al formato de longitud variable en lugar de su formato nativo de longitud fija. También cambia las cadenas de caracteres de longitud fija al formato de longitud variable eliminando espacios en blanco. La compresión de páginas implementa la compresión de filas y otras dos estrategias de compresión (compresión de prefijo y compresión de diccionario). Puede encontrar más detalles sobre la compresión de páginas en "Implantación de Compresión de Página".

Actualmente, la compresión de datos es compatible en las ediciones Enterprise, Developer y Evaluation de SQL Server 2008 y versiones posteriores. Aunque la propia base de datos puede realizar la compresión, esto rara vez se observa en un entorno de SQL Server.

Aquí están las recomendaciones para administrar el espacio para los archivos de datos de SQL Server

- Use thin provisioning en los entornos SQL Server para mejorar el aprovechamiento del espacio y reducir los requisitos generales de almacenamiento cuando se utilice la funcionalidad de garantía de espacio.
  - Use el crecimiento automático para las configuraciones de puesta en marcha más comunes porque el administrador de almacenamiento solo necesita supervisar el uso de espacio en el agregado.
- No active la deduplicación en ningún volumen que contenga archivos de datos de SQL Server a menos que se sepa que el volumen contenga varias copias de los mismos datos, como restaurar la base de datos a partir de backups en un único volumen.

Recuperación de espacio

La recuperación de espacio se puede iniciar periódicamente para recuperar el espacio no utilizado en una LUN. Con SnapCenter, puede utilizar el siguiente comando de PowerShell para iniciar la recuperación de espacio.

```
Invoke-SdHostVolumeSpaceReclaim -Path drive_path
```

Si necesita ejecutar la recuperación de espacio, este proceso debe ejecutarse en períodos de baja actividad

porque inicialmente consume ciclos en el host.