



Guía Práctica de SQL Server 2008 para profesionales de Oracle

Artículo técnico de SQL Server

Publicado: Octubre 2008

Se aplica a: SQL Server 2008

Resumen: en este whitepaper se revisan muchas de las principales comparaciones entre los entornos de base de datos SQL Server 2008 y Oracle. Destaca las arquitecturas respectivas de ambos productos y ofrece información sobre las últimas novedades introducidas en SQL Server 2008. Este documento va dirigido a los profesionales de Oracle que quieren ampliar sus conocimientos y aplicarlos a la gestión de SQL Server. Entre los temas abordados se incluyen el backup, la seguridad de la base de datos, opciones de gestión y alta disponibilidad.

Copyright

Este es un documento preliminar y puede cambiar notablemente antes de la aparición de la versión comercial del producto descrito en él.

La información contenida en este documento representa la visión actual de Microsoft Corporation acerca de los temas analizados en la fecha de publicación. Debido a que Microsoft debe responder a condiciones de mercado cambiantes, no debe interpretarse como un compromiso por parte de Microsoft, y Microsoft no puede garantizar la exactitud de ninguna información aquí mostrada después de la fecha de su publicación. Este documento se publica con fines informativos exclusivamente. MICROSOFT NO OFRECE GARANTIAS, EXPRESAS, IMPLICITAS NI ESTATUTARIAS SOBRE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO

El cumplimiento con todas las leyes aplicables de copyright es responsabilidad del usuario. Sin que suponga limitación a los derechos de copyright, ninguna porción de este documento puede reproducirse, almacenarse o introducirse en un sistema de recuperación o transmitirse de forma alguna o por medio alguno (electrónico, mecánico, fotocopia, grabación o cualquier otro), ni para fin alguno sin el consentimiento expreso y por escrito de Microsoft Corporation.

Microsoft puede tener patentes, aplicaciones de patentes, marcas registradas, copyrights u otras modalidades de propiedad intelectual protegiendo los asuntos analizados en este documento. Excepto cuando así se prevea en los contratos de licencia por escrito de Microsoft, la modificación de este documento no le otorga ningún tipo de licencia sobre dichas patentes, marcas, copyrights o derechos de propiedad intelectual.

© 2008 Microsoft Corporation. Todos los derechos reservados.

Microsoft, Active Directory, SQL Server, Windows, y Windows Server son marcas registradas del grupo de empresas de Microsoft

Otros nombres de empresas y productos mencionados aquí pueden ser marcas registradas de sus respectivos propietarios.

Índice

Introducción	4
Implementación de Oracle de Instancias y Bases de Datos	4
Implementación de Instancias y bases de datos en SQL Server	5
Arquitectura del almacenamiento	6
Mapa de Base de Datos	7
Modelo de log	9
Backup y restauración.....	11
Backups lógicos.....	11
Backups físicos.....	12
Backups en frío / consistentes	12
Backup online / inconsistente	12
Backup incremental	13
Seguridad de la base de datos.....	14
Cifrado de datos transparente	15
Cuentas de usuario	16
Autenticación	17
Passwords.....	18
Escalabilidad y Alta Disponibilidad.....	19
Gestión dinámica de la memoria	19
Ejecución de sentencias SQL en paralelo.....	19
Inserciones Direct-Path.....	20
Procesos de servidor compartidos (antes denominado MTS)	20
Asignación de espacio recuperable	20
Métodos de acceso	20
Gestión de datos agregados.....	21
Tamaños de bloque múltiples.....	21
Tablas externas.....	21
RAC	22
Particionado de tablas.....	22
Replicación de datos.....	22
Base de datos en standby.....	23
Soporte para AWE.....	23
Soporte para arquitecturas de 64 Bits	23
NUMA	24
SOA	24
Broker de Servicio de SQL Server	24
Oracle Streams Advanced Queuing.....	25
Conclusión.....	25

Introducción

Tanto Oracle como Microsoft utilizan los conceptos definidos de instancias y bases de datos. Comparten muchas características similares, pero conservan sus propias estrategias exclusivas. Puesto que ambos productos llevan muchos años en el mercado y continúan introduciendo mejoras, algunos términos (“servidor”, “sistemas”, “instancias” y “base de datos”) se han utilizado para describir a uno en comparación con el otro. Hay que reconocer que incluso los propios fabricantes tienen su parte de responsabilidad en esta confusión. La Base de datos de Oracle es algo más que una serie de archivos de bases de datos para su uso en una instancia concreta. De igual forma, el nombre de producto Microsoft® SQL Server® no se limita a una única instancia que se ejecuta en el servidor físico.

Implementación de Oracle de Instancias y Bases de Datos

Una implementación de la Base de Datos Oracle consiste en una serie de archivos binarios. Entre ellos aparecen los archivos ejecutables del paquete de software, archivos de la red y herramientas de administración. También se instalan los “archivos de Base de Datos”, que constan de archivos de control, archivos de tablespace y archivos de log.

Cuando se arranca una instancia de Oracle en un servidor Windows®, accede a los archivos de inicialización (SPFILE o init.ora) para determinar los parámetros necesarios para el arranque del servicio. Oracle reserva después una cierta cantidad de memoria RAM del sistema operativo dentro de un buffer especial llamado SGA (System Global Area) y a continuación, administra los procesos en background. Estos procesos utilizan threads de procesamiento de Windows para ejecutar el código en los procesadores del sistema y establecer el nivel de aislamiento requerido con respecto a las demás aplicaciones que se ejecutan en el sistema. Una vez en marcha la instancia, se puede montar la base de datos leyendo información desde el archivo de control de dicha base de datos. Finalmente, la base de datos queda accesible a las peticiones de los clientes. Una base de datos puede no estar abierta, incluso puede estar sin montar. La instancia permanece en ejecución y puede escuchar las peticiones procedentes de la red.

Una instancia puede asociarse con una sola base de datos. En algunos entornos el diseño se puede configurar de forma que una misma base de datos quede montada en más de una instancia. Dada la potencia de los ordenadores actuales, muchas veces es posible y tiene sentido instalar varias instancias dentro del mismo sistema. Todas ellas se configuran como procesos y regiones de memoria independientes para el sistema operativo, a fin de aprovechar las ventajas en seguridad y aislamiento que permite esta arquitectura. EL número máximo de instancias que se pueden ejecutar en el mismo sistema depende de la potencia del procesador y la cantidad de memoria del equipo. Obviamente, debe aplicarse el sentido común a la hora de asignar los recursos. Los mismos programas ejecutables que se emplean para arrancar una sola instancia pueden arrancar instancias adicionales en el ordenador. Se pueden también utilizar varias instalaciones para independizar el acceso a archivos binarios y versiones concretas del producto dentro de un mismo sistema.

Una instancia única puede soportar las transacciones de más de una aplicación externa. Esto es importante para introducirnos en algunos de los aspectos que diferencian una solución Oracle de otra basada en SQL Server. A fin de disponer de soporte para múltiples usuarios y aplicaciones. Oracle incorpora esquemas dentro de su arquitectura. Un esquema es simplemente un conjunto de objetos que sirven a un mismo propósito. En el caso de las bases de datos, consiste en una colección de estructuras lógicas de datos que soportan una operación o un usuario. Los productos disponen también de opciones de gestión y seguridad. La mayoría de las veces un esquema, que va asociado con una cuenta de usuario concreta, se utiliza para independizar aplicaciones distintas unas de otras. Los objetos del esquema son muchos y muy variados en Oracle.

El objeto más evidente y más utilizado por los administradores de bases de datos (DBAs) y desarrolladores es el objeto bidimensional llamado "tabla". Otros objetos generalmente soportan el uso de tablas. La tabla representa una entidad única descrita por sus atributos (o columnas). Además su contenido viene dado por instancias independientes de la entidad, cada una de ellas con información diferente asociada a sus atributos abstractos. A menudo se las conoce con el nombre de filas.

Implementación de Instancias y bases de datos en SQL Server

Una instancia creada con SQL Server es un conjunto de archivos binarios que aíslan los servicios del Motor de Base de Datos de SQL Server de las demás aplicaciones. También se utilizan archivos independientes de base de datos. Entre los archivos binarios hay ejecutables, archivos de red y herramientas de administración, necesarias para ejecutar una base de datos sobre un sistema Windows. Cuando se ejecuta una instancia, obtiene todos los parámetros de configuración y reserva áreas de memoria (buffers) dentro de la RAM, para después generar los threads que se encargan de la ejecución del código en la CPU.

Un mismo ordenador puede ejecutar varias instancias de SQL Server. EL objetivo principal que persigue la arquitectura de instancia múltiple es el de aislar los recursos empleados por las distintos tipos de actividades de base de datos que se ejecutan dentro del mismo servidor físico. Cada instancia utiliza de forma independiente los recursos de memoria y procesador disponibles. SQL Server soporta hasta 50 instancias en las ediciones Estándar y Enterprise.

¿Le suena familiar? Efectivamente, Oracle y SQL Server son dos sistemas de gestión de bases de datos relacionales (RDBMSs), una variación del modelo de soluciones DBMS. Pero independientemente de las semejanzas que existen entre ambos productos tal y como se han descrito anteriormente, las diferencias aparecen en seguida. Si, por ejemplo, en Oracle los mismos archivos binarios pueden emplearse para la ejecución de múltiples instancias, en SQL Server cada instancia requiere la instalación de sus propios ejecutables. Microsoft ha decidido proteger no solo los procesos y la memoria de cada instancia de forma independiente, sino también los archivos físicos, de manera que se comportan como entidades aisladas en caso de surgir problemas de acceso al espacio de almacenamiento donde se guarda cada grupo de archivos binarios. Otro nivel de protección se aplica a las instancias que pueden necesitar una actualización o la copia de un archivo concreto en ellas. Esto puede hacerse en una instancia sin que afecte a las demás y, por consiguiente, a aquellas aplicaciones que aún no han sido probadas con la actualización o que simplemente no necesitan este cambio.

Una instancia no se asocia necesariamente con una sola base de datos. En realidad, cada instancia instalada contiene su propio conjunto de cuatro “bases de datos del sistema” donde se guardan los parámetros de configuración específicos de dicha instancia. Estas bases de datos tienen una misión similar a la de los archivos de control de Oracle. Pueden considerarse como “contenedores” de una o más bases de datos. ¿Le suena familiar también? Si está empezando a concebir las bases de datos como algo equivalente a los esquemas, en cierto sentido es así.

Los DBAs de SQL Server instalan una instancia en un equipo. Pueden después crear una base de datos, independiente de las bases de datos del sistema antes mencionadas. Esta base de datos contendrá una serie de objetos, como por ejemplo tablas. Estos objetos soportarán algunas funciones concretas. El aislamiento de la actividad ofrece ciertas ventajas en términos de administración y seguridad. El aislamiento no solamente se produce a nivel de archivos de datos, sino también a nivel de las transacciones en cada base de datos. Oracle utiliza los logs “redo” (“rehacer”) para capturar las actividades transaccionales de la instancia y su base de datos. En SQL Server, igual que en Oracle, cada base de datos dispone de un archivo propio, el archivo de log, para el seguimiento y administración de las transacciones. Así pues, cada base de datos dispone de un sistema de log de transacciones individual.

Arquitectura del almacenamiento

La introducción a los conceptos de bloques, extents y segmentos conduce a dos preguntas ¿qué consecuencias tiene el emplear un tamaño de bloque fijo de 8 Kbytes y qué efecto puede ocasionar el uso de un tamaño fijo de extent de 64 Kb con respecto a la fragmentación interna que puede producirse a consecuencia del uso de extents de tamaño variable? Este tema es uno de los que más interesan y preocupan a los DBAs.

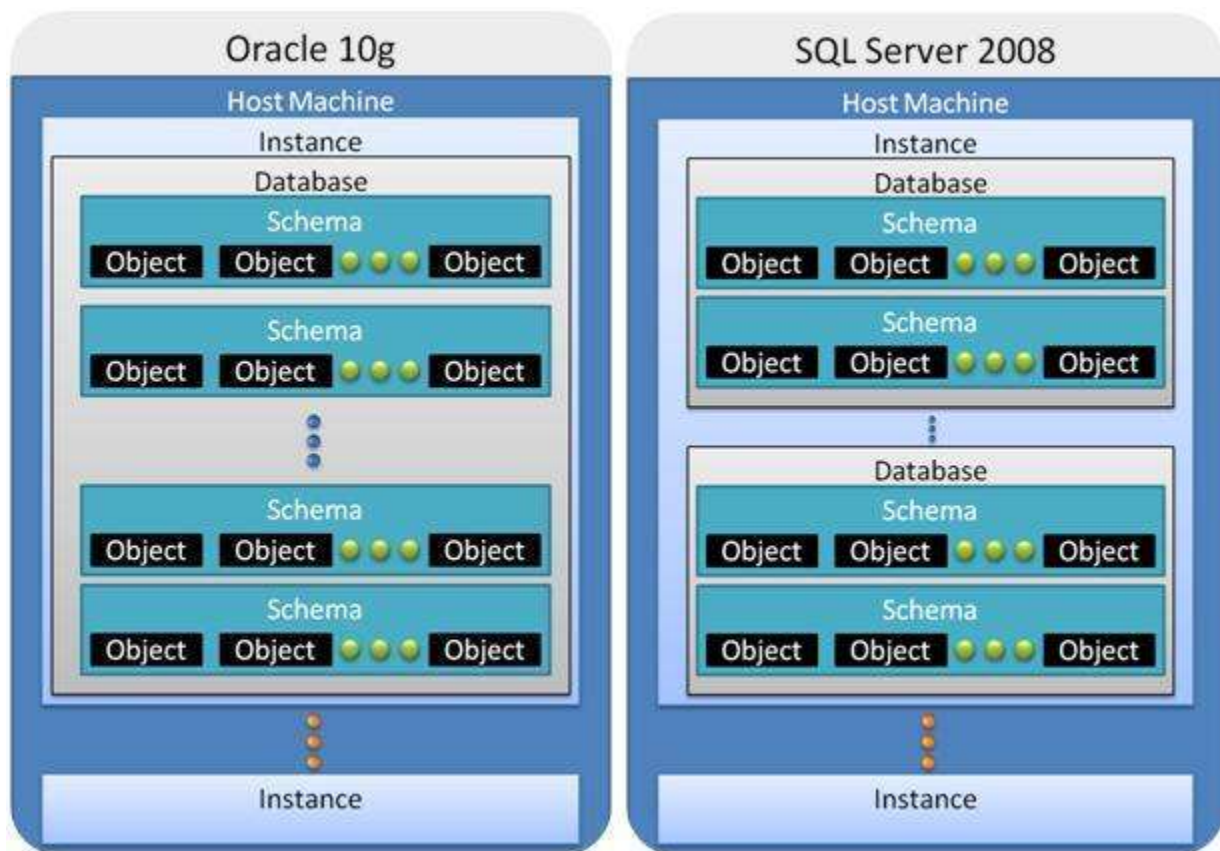


Figura 1 Organización del motor de base de datos de Oracle comparado con SQL Server

En Oracle, puesto que los extents pueden ser de tamaño variable, los datos se pueden escribir en ocasiones de manera no contigua. En SQL Server, al utilizar extents de un tamaño fijo (64 Kb), los datos pueden escribirse y recuperarse de un lugar del disco en vez de leer y escribir en un número elevado de posiciones no contiguas. La fragmentación interna del tablespace (que se denomina “filegroup” en SQL Server) es un asunto muy debatido en el mundo Oracle que en SQL Server se evita por completo al utilizar extents de tamaño fijo.

Oracle ha introducido el ASM (Automatic Storage Management), una tecnología que permite la gestión del disco físico. ASM permite el uso de sistemas de archivo en modo “raw” (no estructurado por el propio S.O.), y automatiza la gestión de archivos desde la base de datos Oracle. ASM soporta la gestión del espacio del disco y también facilita la incorporación o eliminación de discos, encargándose Oracle de reequilibrar los diferentes almacenamientos de datos a medida que se introducen cambios.

Tanto en Oracle como SQL Server, el espacio necesario para la definición y utilización de muchos de los objetos de esquema (triggers, procedimientos, etc.) y objetos de nivel de sistema (usuarios, roles, etc.) se sitúa fuera del diccionario de datos.

Mapa de Base de Datos

Cada instancia de SQL Server incluye dos estructuras de base de datos, **model** y **msdb**, que no tienen un equivalente en Oracle. La base de datos **model** forma la plantilla que se utiliza

para generar nuevas bases de datos. Es a grandes rasgos, lo mismo que generar un esquema de inicio utilizando la herramienta de configuración de base de datos en Oracle. La base de datos **msdb** la utiliza la aplicación SQL Server Agent para almacenar información sobre trabajos, eventos, planificaciones, operadores y demás. Esta base de datos combina las funciones que realizan en Oracle los RDBMS_JOBS (DBMS_JOB y DBMS_SCHEDULER) y el repositorio Intelligent Agent.

En la siguiente tabla se muestran los tablespaces de Oracle y las estructuras de almacenamiento equivalentes de SQL Server en su caso.

Tabla 1 Tablespaces de Oracle y estructuras de almacenamiento de SQL Server Storage Structures con funciones equivalentes:

SQL Server	Oracle
Base de datos master	Tablespace System
Base de datos master	Tablespace Sysaux
Base de datos de recursos	Tablespace System/Sysaux
Base de datos tempdb	Tablespace temporal
Log de transacciones	Tablespace Undo (Rollback)
Log de transacciones	Log Redo online
BD "Aplicación" filegroup -"Datos"	Tablespace "Bigfile de Datos de Aplicación"
BD "Aplicación" filegroup -"Datos"	Tablespace "Datos de Aplicación"
BD "Aplicación" filegroup -"Datos"	Tablespace "Índice de Aplicación"
Base de datos model	No disponible en Oracle
Base de datos msdb	No disponible en Oracle

No es posible categorizar cada una de las bases de datos dentro de una instancia de SQL Server como integral o "autocontenida" en el mismo sentido que tienen estos términos en las bases de datos Oracle. Estos son algunos ejemplos de autonomía y de dependencia sobre la estructura central:

- En SQL Server cada base de datos dispone de sus propios archivos de log de transacciones que agrupan y combinan las funciones de los logs redo online de Oracle y los segmentos de rollback/undo.
- Cada base de datos de SQL Server tiene sus propias estructuras de seguridad, como usuarios, privilegios (permisos) y roles.
- Cada base de datos SQL Server dispone de sus propios roles administrativos que asignan privilegios únicamente a la base de datos a la cual pertenecen.
- En SQL Server el Catálogo del Sistema, que es análogo al diccionario de datos de Oracle, se copia en cada una de las bases de datos individuales, la base de datos **master** y la base de datos Resource (oculta y de solo lectura).
- El espacio temporal (llamado **tempdb** en SQL Server y tablespace temporal en Oracle) es único y común para toda la instancia.

Los tablespaces y filegroups (denominados como “grupos de archivos” en la versión en español de SQL Server) dan la opción de optimizar la distribución de los datos repartiéndolos entre múltiples archivos, para mejorar el rendimiento. El agrupamiento además contribuye a facilitar otras actividades administrativas como la copia de seguridad y restauración, el mantenimiento (pudiendo cambiar el estado online/offline para tablespaces determinados) y permite aplicar otras técnicas, como el particionado de tablas.

Modelo de log

Los logs redo online se usan en Oracle para registrar los cambios transaccionales (DML y DDL) realizados sobre la base de datos antes de que dichos cambios se hagan efectivos en los archivos de base de datos. En SQL Server los logs redo reciben el nombre de logs de transacciones. Oracle utiliza también los tablespaces “undo” para capturar una imagen de los datos previa al cambio para facilitar la operación de reversión (rollback), la restauración de datos y consistencia en la lectura (conurrencia de múltiples versiones).

En las primeras versiones de BD Oracle se utilizaban los segmentos de rollback para almacenar la información de undo. A partir de Oracle 9i se introduce la gestión automática de undo, que simplifica la gestión del espacio de undo al eliminar las complejidades asociadas con la gestión del segmento de rollback.

Oracle utiliza dos o más logs redo online de tamaño fijo. Cuando un log redo se llena se establece un punto de comprobación o “checkpoint” y la base de datos comienza a escribir los registros de transacciones en el siguiente log redo online. Entre tanto, todos los buffers de base de datos “sucios” relacionados con los registros en el log completado se aplican a la base de datos por medio de un proceso de escritura (DBWR). Además se puede hacer una copia del log redo cuando se utiliza el modo **archivelog**, que puede utilizarse para una restauración posterior de la base de datos.

El log de transacciones de SQL Server combina la funcionalidad de los logs redo de Oracle y los segmentos undo. Cada base de datos de SQL Server dispone de uno o más archivos de log de transacciones. El log de transacciones es un archivo de contenidos empaquetados. Si el log está distribuido entre varios archivos físicos, se organizan secciones dentro de cada uno de ellos y se va accediendo a cada sección de forma cíclica (“round robin”) para registrar las transacciones y posteriormente se empaqueta todo y se devuelve al inicio del primer archivo físico de log.

SQL Server segmenta cada archivo de log de la base de datos internamente en un cierto número de “archivos de log virtuales”. Los archivos de log virtuales varían en tamaño y no hay un número predefinido de archivos de log virtuales por cada archivo físico. SQL Server elige el tamaño de los archivos de log virtuales de forma dinámica cuando se genera o amplía el espacio de log. SQL Server trata de mantener un número reducido de archivos virtuales. El tamaño o número de los archivos de log virtuales no puede configurarse ni modificarse desde las herramientas de administración: es el propio SQL Server quien lo establece internamente y de manera dinámica.

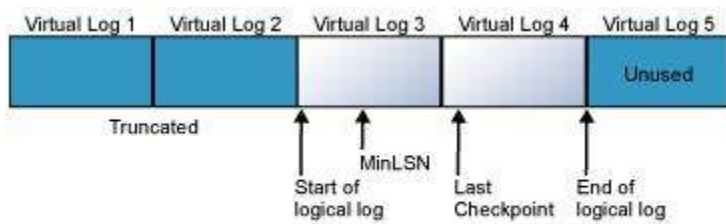


Figura 2 Arquitectura del log de transacciones de SQL Server

Los nuevos registros de log, identificados con un número de secuencia de log (LSN), se van añadiendo al final del log de transacciones. Los registros de log correspondientes a modificaciones de los datos registran bien la operación lógica realizada o bien las imágenes anterior y posterior de los datos modificados. Los registros de log se almacenan de forma secuenciada conforme se van generando. Cada registro de log lleva marcado el ID de la transacción a la cual pertenece. Para cada transacción, todos los registros de log asociados con dicha transacción se encadenan de forma simple utilizando punteros hacia el registro previo que aceleran el proceso de rollback.

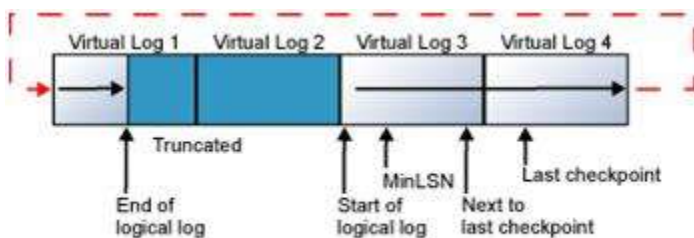


Figura 3 Empaquetado del log de transacciones de SQL Server

De manera parecida a los segmentos undo de Oracle, las entradas de log virtuales pueden sobrescribirse si se ejecuta la operación de *commit* de sus correspondientes transacciones. La función y los aspectos administrativos relativos a los logs de transacciones en SQL Server son una mezcla de los aspectos y funciones asociados con los segmentos de undo y los logs redo de Oracle. Cuando el log de actividad se empaqueta al principio del log, el espacio puede reutilizarse únicamente si las transacciones ya se han confirmado (commit), -como ocurre con los segmentos undo-, si los checkpoints se han establecido correctamente -como con los logs redo- y los datos se han eliminado de los archivos de log virtuales mediante una actividad de Log de Backup -parecida a un conmutador-. La función de Log de Backup trunca los datos confirmados. En cuanto se produce el truncamiento, se eliminan los registros de los logs virtuales con LSN anterior al del primer registro de imagen del log o el número de secuencia de log de restauración más bajo (MinLSN). Una correcta planificación de los backups (que puede hacerse programando tareas de archivo de log) pueden servir para efectuar las operaciones de limpieza y empaquetado de manera automática.

En SQL Server es una buena práctica dimensionar adecuadamente el log de transacciones. La funcionalidad de crecimiento automático de SQL Server tiene que estar habilitada para permitir que los archivos de log puedan ampliarse automáticamente en caso de que el espacio de log

definido se llene por completo, y así se consigue que la base de datos siga funcionando en momentos de actividad transaccional muy elevada o inesperada entre copias de seguridad en el caso de que el DBA no haya dimensionado el log correctamente. La función de crecimiento automático únicamente se inicia en el momento en que el log se queda sin espacio libre, por lo que es mejor monitorizar el espacio disponible y ampliarlo antes de que sea realmente necesario. La opción de auto-compresión en condiciones normales no será necesario utilizarla. Hace que los espacios de log se compriman pero solo para que la función de crecimiento automático vuelva a aumentarlos después. Estas opciones tienen una función similar a las “autoextend” y “shrink” del segmento undo en Oracle.

Backup y restauración

En Oracle los métodos de backup pueden clasificarse en dos grandes categorías: backups físicos y lógicos. Hay dos maneras de realizar en Oracle las operaciones de backup y restauración: con el Recovery Manager (RMAN) y mediante un proceso gestionado por el usuario. Oracle ofrece distintas opciones para la restauración completa después de un fallo grave en una instancia o disco físico: el log redo, los registros undo, un archivo de control, y los backups de base de datos.

RMAN se utiliza para hacer el backup, restaurar y recuperar bases de datos.

Otra posibilidad es la de la operación manual, ejecutada desde el sistema operativo y en este caso se utiliza SQL*Plus para la restauración. Oracle distribuye sus backups en estados consistente e inconsistente, y pueden asimilarse a un backup en frío o en caliente.

SQL Server ofrece backups completos, diferenciales, parciales, del log de transacciones y backups de cola, que permiten reconstruir una base de datos completamente en caso de fallo del disco, del servidor o de una instancia. Hay distintos backups, en frío y en caliente, que se pueden utilizar con SQL Server para adaptarse a cualquier escenario de uso. Las bases de datos de SQL Server pueden también desconectarse de forma rápida y copiar sus archivos físicos, los cuales después pueden asociarse a cualquier otra instancia.

Backups lógicos

El objetivo de un backup lógico es el de poder recuperar datos a nivel de objeto individual del esquema. En Oracle los backups lógicos se realizan principalmente con la utilidad Export. Esta utilidad exporta los objetos del esquema a un archivo binario que solamente puede ser leído desde la utilidad Import, que incorpora los objetos del esquema dentro de la base de datos.

En SQL Server los objetos individuales del esquema pueden copiarse a archivos de disco en cualquiera de los formatos soportados. Posteriormente estos archivos planos pueden restaurarse utilizando herramientas como la utilidad **bcp** (una herramienta de línea de comandos que utiliza el API del programa de copias masivas Bulk Copy Program, o BCP), el comando BULK INSERT, el asistente de Importación y Exportación o las herramientas de SQL Server Integration Services.

Backups físicos

Los backups físicos son copias de los archivos de físicos de la base de datos. En Oracle estos archivos incluyen archivos de datos, archivos de control y si la base de datos está en modo ARCHIVELOG MODE, los archivos de log redo ya marcados.

El proceso es similar al de SQL Server. Generalmente un backup en SQL Server se hace a nivel de base de datos. Las bases de datos más grandes pueden emplear procesos de backup a nivel de grupo de archivos o de archivo individual para hacer copias de seguridad parciales de la base de datos. Estas prácticas generalmente se dan cuando se trata de reducir el tiempo necesario para completar el backup o el volumen de información a proteger. Los backups del log de transacciones se encargan de la copia de seguridad de estos registros. En SQL Server también puede hacerse backup de ciertos componentes de seguridad importantes como son la Clave Maestra del Servicio, la Clave Maestra de la Base de Datos, la Clave de Cifrado y Certificados.

Backups en frío / consistentes

En Oracle, el backup “consistente” de una base de datos, completa o parcial, es un backup en el que todos los archivos de datos de lectura/escritura y los archivos de control están marcados con el mismo SCN (System Change Number), que es un valor único asignado a cada transacción confirmada de Oracle. Los backups en frío no son posibles en la mayoría de entornos de misión crítica con requisitos de alta disponibilidad, ya que exigen la parada del sistema.

En Oracle, para crear un backup consistente de toda la base de datos hay que detenerla previamente con las opciones NORMAL, IMMEDIATE o TRANSACTIONAL. El paso siguiente sería hacer el backup mientras la base de datos está cerrada.

Con las bases de datos de SQL Server se puede hacer un backup mientras están online. El backup online completo es una funcionalidad parecida al backup en frío (o “consistente”) de Oracle. Este método hace una copia de toda la base de datos y parte del log de transacciones, pero con la diferencia de que no requiere la parada de la base de datos: puede tener conexiones activas mientras se realiza la copia. Al restaurar el contenido desde este backup, la base de datos se recupera a un estado consistente y puede utilizarse y accederse a los datos siempre que no sea necesario aplicar copias de los log de transacción posteriores.

En SQL Server los backups completos reflejan el estado de la base de datos en el momento en que había terminado el backup. Un backup completo online por si solo es suficiente en todos los aspectos para restaurar una base de datos a un estado consistente.

Backup online / inconsistente

Las demandas de disponibilidad 24x7 de muchos entornos en explotación dificultan la posibilidad de dejar fuera de servicio toda la base de datos. En tal caso, Oracle permite realizar un backup online, en el cual se captura la información de la base de datos en un estado inconsistente, donde no todos los cambios están marcados con el SCN. Los administradores pueden cambiar el modo de la base de datos a ARCHIVELOG para realizar el backup de tablespaces y archivos de control en momentos distintos. Con esta técnica se distribuye el backup en el tiempo y su ventaja es que solo afecta a tablespaces concretos –no a todos ellos-

y cada uno de los backups parciales se ejecuta en menos tiempo. Si se utiliza RMAN para realizar un backup online no es necesario dejar los tablespaces en modo de backup.

En SQL Server el backup online completo de la base de datos contiene toda la base de datos y parte del log de transacciones. Se pueden hacer backups completos, diferenciales y parciales de los archivos y grupos de archivos utilizando la sentencia BACKUP DATABASE o con la herramienta SQL Server Management Studio. Los log de transacciones pueden copiarse aparte también. Un backup completo no limpia el log de transacciones: hay que iniciar un proceso de backup para limpiar el log cada cierto tiempo e impedir que se llene el espacio reservado a este fin. Normalmente una buena estrategia de backup debería incluir backups periódicos del log de transacciones, para evitar este efecto. Cualquier transacción en curso en el momento de realizarse el backup se incluye dentro de él.

La copia de seguridad del log de transacciones permitirá además recuperar la base de datos al estado que tenía en cualquier momento anterior.

Backup incremental

Los backups incrementales físicos se hacen para capturar únicamente los bloques modificados y con ello reducir el tiempo y espacio necesario para la copia de seguridad. Estos backups se realizan siempre después de un backup completo. Aunque su restauración lleva más tiempo, los backups incrementales son muy utilizados por el ahorro de tiempo y espacio de almacenamiento que suponen.

En Oracle los backups incrementales se pueden hacer desde el RMAN o desde herramientas de otros fabricantes. En el caso de RMAN, se analizan los archivos de datos y solamente se vuelcan al backup los bloques que han sido modificados. RMAN ofrece dos tipos de backups incrementales: el incremental propiamente y el diferencial. Los backups incrementales (nivel 1) afectan solamente a los datos modificados desde el último backup completo (nivel 0) o incremental (nivel 1). Un backup diferencial (también nivel 1) recopila todos los datos modificados desde el último backup completo (nivel 0).

En SQL Server las modalidades de backup que explicamos a continuación son de alguna manera análogas a los backups incrementales de Oracle:

- **Backups del log de transacciones**—Los backups de log de transacciones de SQL Server obedecen a los mismos fines que los backups incrementales de Oracle. Este tipo de backup almacena todas las transacciones registradas desde el último backup de log de transacciones.
- **Backups diferenciales**— se capturan todos los datos modificados desde el último backup completo. Los backups diferenciales son acumulativos desde el último backup completo solamente, no son incrementales. Los backups diferenciales no permiten recuperar la base de datos a un momento concreto o a un punto en el log de transacciones.
- **Backups parciales**—SQL Server dispone también de backups parciales, que contienen solo los datos de los grupos de archivos Primario y lectura/escritura de la base de datos. Los backups parciales no incluyen los grupos de archivos de solo lectura. Si se

indica la directiva READ_WRITE_FILEGROUPS en la sentencia BACKUP, solamente se copian los grupos de archivos de lectura/escritura y se omiten los de solo-lectura, ya que se supone que ya debe existir copia de seguridad de éstos.

Cada modalidad de backup tiene sus ventajas y sus inconvenientes, dependiendo del entorno donde se apliquen. A la hora de implementar una solución concreta hay que tener en cuenta distintos factores, como el tiempo disponible, el volumen de información a copiar y la capacidad de recuperación. Por ejemplo, un backup parcial puede hacerse en una ventana de tiempo mucho más corta que un backup completo y puede evitar tener que hacer backups frecuentes de los log de transacciones. Pero por el contrario, no se puede utilizar para restauraciones completas o a un momento concreto del pasado.

De igual manera, los backups diferenciales parciales pueden ser más rápidos que los backups parciales, pero durante el proceso de restauración puede que se necesite utilizar tanto el backup parcial como el diferencial, aumentando el tiempo necesario y la complejidad de la operación de restauración.

Seguridad de la base de datos

La seguridad en la base de datos es una estrategia que se despliega a varios niveles y con distintos mecanismos para controlar el acceso a la información, el diccionario de datos, los objetos y la propia base de datos. En esta sección se analizan los mecanismos disponibles en Oracle y se comparan con los de SQL Server.

Oracle incorpora medidas de seguridad muy importantes para responder a las necesidades de los entornos corporativos, como son la integridad de los datos y la privacidad, autenticación, logon unificado (SSO) y autorización de accesos. El conocido protocolo SSL (Secure Sockets Layer) también está disponible como mecanismo de cifrado en Oracle Net para la encriptación de la información por la red. En estos entornos es indispensable la implementación de un mecanismo de autenticación potente, lo que se consigue mediante estándares como Kerberos, tarjetas inteligentes y certificados digitales.

En SQL Server los mecanismos de autenticación fuerte y cifrado de datos están integrados dentro del sistema. SQL Server ofrece cifrado de datos, protección de accesos a la base de datos mediante passwords, certificados para la ejecución de procedimientos almacenados, SSO y, por supuesto, Kerberos y SSL como protocolos de encriptación.

SQL Server protege los datos mediante una infraestructura jerárquica de cifrado y gestión de claves. Cada capa protege a las inferiores, utilizando una combinación de certificados, claves asimétricas y simétricas. En este diagrama se resume este diseño.

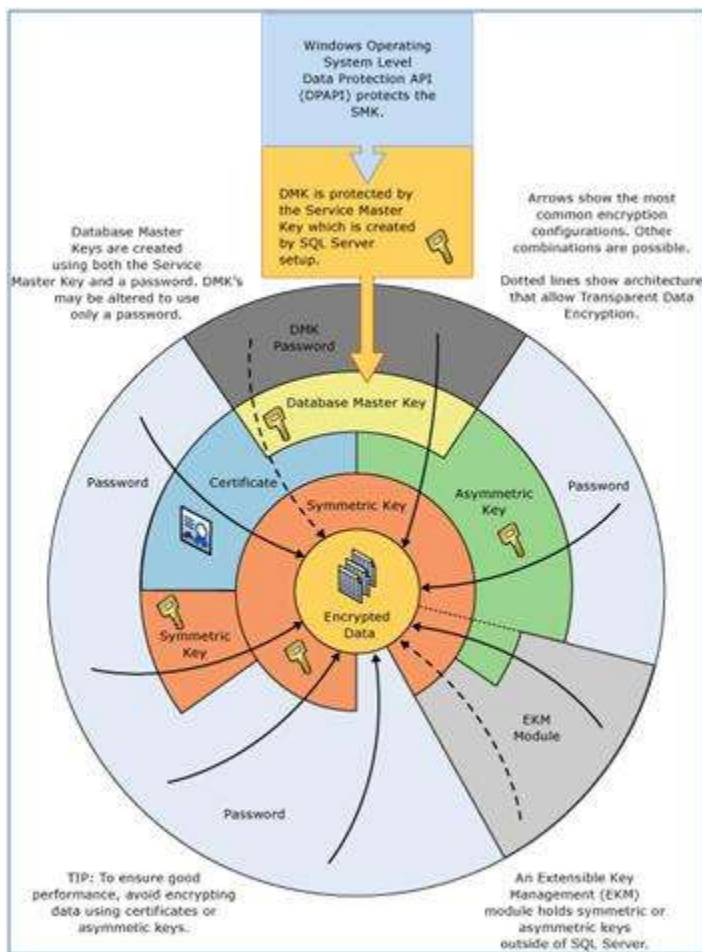


Figura 4 Cifrado de datos en SQL Server

SQL Server 2008 amplía el alcance de este modelo de gestión de claves permitiendo la recuperación de las claves desde un sistema de gestión de claves externo (EKM, External Key Management), que puede ser por hardware o por software. Así se puede integrar perfectamente SQL Server dentro de una infraestructura de gestión de claves corporativa.

Cifrado de datos transparente

Tanto Oracle como SQL Server utilizan cifrado de datos para proteger la información. SQL Server 2008 incorpora la tecnología TDE (Transparent Data Encryption) y Oracle ofrece una modalidad de TDE mediante la definición de una clave maestra y un almacén de claves personales (el "wallet"). En Oracle el uso de TDE se declara en las definiciones de tabla y se establece a nivel de columna.

En SQL Server 2008, TDE se encarga de cifrar los datos, logs y archivos de backup de toda la base de datos. Con ello se impide que un intruso pueda acceder a los archivos de la base de datos y simplemente los abra con otra instancia de SQL Server. Los archivos se encriptan por medio de un certificado y el acceso a la base de datos solo se puede hacer utilizando dicho certificado. El proceso de encriptación/desencriptación tiene lugar en el momento en que las páginas de los datos o logs se recuperan desde el sistema de E/S para su lectura o escritura.

Los datos dentro de la cache de SQL Server generalmente están en texto plano, pero en el disco se almacenan encriptados.

Una vez activada la TDE, toda la base de datos y el log se encriptan, una operación que llevan a cabo ciertos procesos en background, y las páginas se encriptan en el momento en que van o vienen desde el disco. EL impacto que este método tiene sobre el rendimiento depende de los patrones de acceso a que está sometida la base de datos. La base de datos **tempdb** estará encriptada si cualquiera de las bases de datos de la instancia lo está, por lo que el rendimiento general del sistema puede verse afectado en la medida en que **tempdb** es un recurso común a todas ellas.

Ni qué decir tiene que es esencial tener copia de seguridad del certificado de encriptación de la base de datos. Sin él es imposible acceder a ella o restaurarla. El backup del certificado debe hacerse de manera segura y, como con cualquier otro sistema basado en claves, los datos estarán tan protegidos como lo esté la propia clave.

Cuentas de usuario

Tanto Oracle como SQL Server utilizan el concepto de “cuenta de usuario” para que los usuarios autorizados puedan conectarse a la base de datos. En Oracle recibe el nombre de usuario o “username”. En SQL Server se le llama “login ID” o simplemente “login”. Cualquier operación que realice un usuario está regida por los privilegios asignados a su cuenta.

En las bases de datos Oracle encontramos tres tipos de usuarios:

- Propietarios del Esquema—usuarios que crean y mantienen los objetos relacionados con una aplicación
- Usuarios de la aplicación—usuarios (o sistemas) que manipulan los datos de las tablas creadas por los propietarios.
- Usuarios administrativos—usuarios con roles especiales como el de administrador de la base de datos o administrador de la seguridad.

En SQL Server la cuenta permite al usuario conectarse a una instancia. No obstante, el acceso a otras bases de datos dentro de la instancia no es automático y está controlado por otras cuentas (llamadas usuarios) que existen en cada una de las bases de datos a las cuales el login puede acceder. Los privilegios a nivel de instancia se asignan al login y los privilegios dentro de una base de datos se asignan al usuario correspondiente dentro de la base de datos. El usuario de base de datos se mapea con un login de instancia aunque el nombre de usuario en cualquier base de datos puede consistir en un pseudónimo. El login de SQL Server es el objeto sobre el que se verifica el proceso de autenticación, mientras que el usuario de base de datos es el que recibe la autorización para acceder a los objetos de dicha base de datos

Los usuarios de SQL Server pueden también clasificarse en tres categorías, como en Oracle:

- Propietarios del Esquema – usuarios que crean y mantienen objetos relacionados con una aplicación.

- Usuarios de la aplicación —usuarios (o sistemas) que manipulan los datos contenidos en las tablas.
- Usuarios administrativos—usuarios con roles especiales como el de administrador de la base de datos o administrador de la seguridad.

Los requisitos de cuenta y privilegios de cada uno de estos tres tipos son distintos. EN el caso de los administradores de una instancia, el login es suficiente, pero los administradores de las bases de datos necesitan disponer de cuentas en dichas bases de datos.

Los propietarios de objetos de base de datos en SQL Server necesitan un login para el servidor y una cuenta de usuario en cada una de las bases de datos donde tengan que crear objetos. Igualmente, los usuarios de aplicaciones necesitan un login para el servidor y una cuenta de usuario en las bases de datos a las que acceden.

Hay un usuario especial, llamado “dbo” con perfil de propietario de base de datos presente en todas las bases de datos de SQL Server y dispone de los privilegios necesarios para realizar cualquier actividad dentro de ellas. Un login administrativo con el rol **sysadmin** se mapea directamente en todas las bases de datos con el usuario dbo. Es más o menos lo mismo que cuando un usuario se conecta a una base de datos identificándose como SYSDBA o SYSOPER, que se conecta automáticamente como SYS. Los logins de SQL Server pueden mapearse también contra el usuario dbo en una base de datos, con lo que adquieren privilegios de administrador en dicha base de datos.

Autenticación

El proceso de autenticación es aquel por el cual se comprueba que el login ID o el username introducido por un usuario al conectarse a una base de datos pertenece a un usuario autorizado. Tanto Oracle como SQL Server admiten la autenticación desde el sistema operativo y desde la propia base de datos (servidor). En SQL Server el modo de autenticación desde el sistema operativo se llama “Autenticación de Windows” y el modo de base de datos se llama “Autenticación de SQL Server”.

Oracle Advanced Security soporta los siguientes métodos de autenticación:

- Kerberos
- RADIUS
- SSL
- Entrust/PKI

SQL Server soporta la Autenticación Kerberos dentro de los mecanismos proporcionados por el dominio de Directorio Activo de Windows Server®. Los puntos de conexión (“endpoints”) de SQL Server también se pueden configurar para que utilicen autenticación Kerberos.

SQL Server puede utilizar el protocolo SSL para encriptar todos los datos transmitidos entre el sistema cliente donde se ejecuta la aplicación y una instancia de SQL Server dentro de un sistema servidor. El cifrado mediante SSL se efectúa dentro de la librería de red del cliente

nativo de SQL Server y se aplica a todos los protocolos de comunicación entre sistemas soportados por SQL Server.

Aparte esto, el propio sistema operativo Windows soporta el protocolo IPsec (Internet Protocol Security) y se configura a nivel del sistema operativo, no en SQL Server. Los tres métodos de autenticación basados en estándares IETF utilizados para establecer la seguridad en las comunicaciones entre sistemas son los siguientes. Todos los tipos de cifrado están soportados en Windows:

- Autenticación Kerberos v5.0 soportado por el dominio de Windows
- Sistemas PKI basados en certificados y firmas digitales
- Passwords, claves de autenticación compartidas, que solamente se utilizan para el establecimiento de la relación de confianza, pero no para la protección de datos en las transmisiones por la red.

Passwords

Las passwords de Oracle presentan estas características y funcionalidades:

- Restricciones en la composición y complejidad de las passwords
- Directivas de complejidad basadas en función
- Configuración de políticas de renovación (antigüedad)
- Políticas de ciclo de vida de la password (reutilización y otros atributos)
- Bloqueo de cuenta basado en intentos fallidos, caducidad de la password, etc.
- Cifrado de la password

Puesto que la arquitectura de seguridad de SQL Server está diseñada para aprovechar las posibilidades de seguridad de Windows, se integra de forma muy estrecha con los mecanismos de autenticación y autorización del sistema operativo. Por tanto, cuando se utiliza la autenticación Windows en SQL Server se aplican las directivas de protección de password, complejidad, caducidad, perfil, reintentos y cifrado vigentes en el sistema operativo. SQL Server 2008 ha reforzado los mecanismos de autenticación de SQL Server y además utiliza el API NetValidatePasswordPolicy, disponible en Windows Server 2003 y versiones posteriores, para potenciar aún más las políticas de password de Windows del sistema en el cual se ejecuta SQL Server.

Las opciones disponibles para reforzar los logins de SQL Server son:

- CHECK_POLICY
- CHECK_EXPIRATION
- MUST_CHANGE

Estas opciones pueden indicarse en la sentencia CREATE LOGIN DDL o aplicarse desde el SQL Server Management Studio.

Escalabilidad y Alta Disponibilidad

La escalabilidad es la capacidad de gestionar bases de datos muy grandes (VLDB) y/o grandes cantidades de transacciones online (OLTP). El tamaño de las VLDBs puede venir de la existencia de unas cuantas tablas de gran tamaño o un número elevado de tablas de tamaño menor, o una combinación de ambas situaciones. La escalabilidad OLTP se mide en términos como número de conexiones de usuario simultáneas (conurrencia), tiempos de respuesta y número de transacciones procesadas por unidad de tiempo.

En un sentido más amplio, la escalabilidad se mide también por la capacidad de escalar horizontal y verticalmente de la arquitectura subyacente. La escalabilidad vertical se refiere a la capacidad de un servidor o sistema para añadir más recursos, como memoria o procesadores.

En las siguientes secciones analizaremos algunas de las características de escalabilidad que ofrecen Oracle y SQL Server, empezando con las posibilidades y estructuras que ofrecen internamente las bases de datos, para ir avanzando después a otras características de mayor envergadura.

Gestión dinámica de la memoria

Tanto Oracle como SQL Server pueden adaptar dinámicamente el tamaño de los componentes dentro de sus respectivas áreas de memoria para adecuarse a las condiciones de ejecución de las distintas tareas. Con ello se reduce la probabilidad de paginación. SQL Server libera toda la memoria no utilizada cuando lo solicita el sistema operativo. Si el sistema operativo no utiliza toda la memoria para sus propios procesos o para los de otras aplicaciones, SQL Server asignará la memoria a sus buffers de cache cuando lo necesite. En SQL Server se puede especificar la cantidad de memoria mínima reservada para la ejecución de consultas. Este valor puede ir desde 512 Kb a 2 Gb para una sola consulta.

Ejecución de sentencias SQL en paralelo

La ejecución de sentencias SQL en paralelo se subdivide dinámicamente entre diferentes tareas que se distribuyen entre los distintos procesadores. Estas operaciones en paralelo:

- Están soportadas en sistemas de multiprocesamiento simétrico (SMP), procesamiento paralelo masivo (MPP) y clusters.
- Ofrecen mejores rendimientos en operaciones de consulta, creación de índices, operaciones masivas (DML y cargas de datos), agregaciones, copias y creación de tablas utilizando la sentencia SELECT INTO.
- Soportan paralelismo intra-operación e inter-operación.

Oracle y SQL Server disponen de arquitecturas muy similares para las operaciones de SQL paralelas. Los optimizadores de ambos productos controlan la ejecución en paralelo y utilizan las técnicas como los procesos de consulta esclavos (“parallel query slaves”) en el caso de Oracle y los procesos de trabajo (“worker threads”) en SQL Server para llevar a cabo el plan de

ejecución. SQL Server determina de forma automática el grado de paralelismo (DOP, “degree of parallelism”) de cada consulta, de forma muy similar a como hace Oracle.

Inserciones Direct-Path

Durante las inserciones Direct-Path en Oracle, los datos de la tabla se añaden a un archivo de datos, ignorando la cache de base de datos. Esta característica se puede utilizar en sentencias INSERT y con la utilidad SQL*Loader. Las inserciones con Direct-path se pueden ejecutar tanto en serie como en paralelo.

Todas las instrucciones INSERT en SQL Server tienen que pasar por la cache de la base de datos. Las páginas se vuelcan a disco en los intervalos normales CHECKPOINT o cuando existe necesidad de liberar memoria.

Procesos de servidor compartidos (antes denominado MTS)

Oracle ofrece una arquitectura de servidor compartido que utiliza los threads del gestor de procesos (“dispatcher”) para dirigir las peticiones de los usuarios a un conjunto de procesos compartidos de servidor, con lo que se evita la necesidad de disponer de procesos de servidor dedicados para cada conexión. Gracias a esta técnica las bases de datos Oracle pueden escalar para manejar cientos de conexiones de usuario concurrentes.

El modo de servidor compartido es el modo de operación por defecto en SQL Server. Puesto que SQL Server solo funciona sobre sistema operativo Windows, está diseñado para planificar las conexiones de manera eficiente en el nivel de thread o en el de fibra. Con ello SQL Server es capaz de soportar cientos o miles de conexiones de usuario concurrentes.

Asignación de espacio recuperable

La funcionalidad “*resumable space allocation*” (asignación de espacio recuperable) de Oracle permite suspender aquellas operaciones que den lugar a errores de asignación de espacio de disco (espacio insuficiente, agotamiento del número máximo de extents, superación de la cuota) en vez de retrocederles mediante un rollback. La operación en cuestión se recupera una vez que el administrador de la base de datos resuelve la incidencia. Esta característica ahorra muchísimo esfuerzo durante la ejecución de tareas masivas y en operaciones rutinarias en VLDBs, siempre que el disco donde esté alojada la base de datos Oracle no se llene por completo.

Aunque la opción de recuperar las operaciones detenidas no está disponible en SQL Server, esta situación rara vez causa incidencias ya que se puede habilitar el crecimiento automático de los archivos de base de datos. Salvo que la propia partición de disco se quede sin espacio libre, los archivos de la base de datos no se deberían llenar nunca. Se pueden preparar alertas para avisar al DBA en caso de que se supere un umbral preestablecido, lo que da un margen de acción para preparar espacio de disco adicional. No es preciso fijar cuotas salvo que se desee expresamente limitar el crecimiento de la base de datos, y tampoco hay límite para el número máximo de extents.

Métodos de acceso

Para acelerar el acceso a los datos, Oracle dispone de índices en árbol B*-tree (ascendente/descendente), índices de bitmap, índices basados en funciones, tablas

organizadas por índices, índices particionados, Join Indexes, Join Indexes con mapas de bits e índices de clave inversa.

SQL Server acelera el acceso a los datos mediante índices clusterizados y no-clusterizados, índices particionados, índices XML primarios y secundarios, índices de texto completo e índices alineados. Los *índices cubiertos* son consultas donde cada una de las columnas utilizadas forma parte del índice. Los índices cubiertos sirven para mejorar la eficiencia de la consulta y en algunos casos, permiten distribuir los datos físicamente para evitar el agotamiento del espacio de disco.

Gestión de datos agregados

La funcionalidad Data Warehouse de Oracle incluye mecanismos para crear, utilizar y gestionar dinámicamente tablas de datos agregados y reescribir consultas de forma transparente para poder utilizar estos datos resumidos mediante un sistema inteligente de reescritura de consultas. Esta funcionalidad redirige automáticamente las queries ejecutadas contra tablas de datos de detalle para utilizar en su lugar las de resumen.

SQL Server dispone de vistas indexadas, que permiten generar información agregada. El optimizador determina de manera automática si las vistas indexadas pueden servir para resolver todo o parte de una query que solicita esta información.

Tamaños de bloque múltiples

Oracle soporta varios tamaños de bloque (2, 4, 8, 16 y 32 Kbytes) dentro de la base de datos de forma que los objetos pueden colocarse en el grupo de archivos más adecuado para mejorar la eficiencia en operaciones de E/S. Esta capacidad para optimizar la E/S supone un plus de trabajo administrativo puesto que el desplazamiento de las tablas se tiene que hacer mediante operaciones de importación y exportación completas, aun tratándose de tablespaces transportables, si el tamaño de los bloques no coincide.

SQL Server utiliza un tamaño de bloque fijo, la página de 8 Kb. Aunque se restringe la capacidad del DBA para optimizar el rendimiento de las operaciones de E/S, facilita mucho la portabilidad y automatización de tareas. Con un tamaño de bloque fijo, SQL Server puede optimizar de forma automática las operaciones de E/S y asignación de memoria, ya que conoce exactamente cuánto espacio requiere cada operación de E/S.

Tablas externas

Las tablas externas son tablas de solo-lectura cuyos datos se guardan en archivos de texto plano fuera de la base de datos. Suponen una forma de acceder a información no almacenada dentro de la base de datos de la misma forma que si estuviesen dentro de ella, pero sin tener que cargar estos datos dentro de la propia base de datos. Es una herramienta muy útil en operaciones ETL dentro de entornos de Data Warehouse.

SQL puede referenciar orígenes de datos externos mediante servidores vinculados o mediante el comando OPENDATASOURCE

Los archivos externos como son los archivos de texto plano, pueden leerse utilizando el comando OPENROWSET (BULK...) desde el SQL Server Management Studio o con comandos desde la shell de SQL Server Management Studio en modo SQLCMD. Los

servidores vinculados permiten conectar a distintos orígenes de datos, como pueden ser hojas de cálculo o archivos de texto. Los usuarios pueden después realizar operaciones de inserción, actualización y borrado con sentencias SQL contra estos orígenes de datos.

RAC

La arquitectura RAC (Real Application Cluster) de Oracle, en el contexto de las funciones de escalabilidad del producto, ofrece un mayor rendimiento y mejora de prestaciones al permitir la escalabilidad horizontal de la base de datos en un entorno de servidores en cluster. Los nodos de servidor acceden a una misma base de datos compartida, y comparten los recursos de almacenamiento y base de datos, aunque cada uno utiliza su propia memoria y CPU. RAC distribuye la carga entre los distintos servidores. La nueva tecnología **Cache Fusion** minimiza los problemas de rendimiento originados por las actividades de sincronización.

SQL Server utiliza una arquitectura de servidores de bases de datos federados y vistas particionadas distribuidas (DPVs) para lograr una solución de escalabilidad horizontal sin compartir elementos. Una federación de servidores de bases de datos permite distribuir la carga entre un grupo de máquinas particionando horizontalmente los datos en una base de datos SQL Server. Estos servidores se administran de manera independiente, pero son capaces de cooperar para responder a las peticiones de procesamiento de la base de datos.

Particionado de tablas

El particionado de tablas e índices resuelve muchos problemas de disponibilidad, gestión y rendimiento en entornos de tablas extremadamente grandes, ya que:

- Las unidades de gestión de datos son más pequeñas
- Se facilita el emplazamiento de los datos en el disco, mejorando el rendimiento en E/S
- Ofrecen disponibilidad en caso de fallo o durante operaciones de mantenimiento
- Aumenta el paralelismo en operaciones de carga, actualización y batch
- Aumenta el rendimiento mediante el uso de un optimizador que reconoce las particiones y lleva a cabo operaciones especialmente adaptadas al particionamiento, incluyendo consultas, joins y DML en paralelo

Todas las funcionalidades de particionamiento en los niveles de tabla e índice están disponibles tanto en SQL Server 2008 como en Oracle.

SQL Server soporta además una vista particionada, que consiste en una vista UNION ALL de datos particionados almacenados en tablas, dentro de las mismas o distintas bases de datos, dentro de los mismos o distintos servidores). Esta vista se genera con chequeo de restricciones, algo parecido a las vistas particionadas que ofrecía Oracle antes de disponer de tecnología de particionamiento.

Replicación de datos

La replicación es una excelente alternativa para escalar aplicaciones, sobre todo a través de una WAN. No obstante, esta técnica es escalable siempre que las vistas generadas solamente

se utilicen en consultas. La replicación multi-master (Oracle) y la replicación “peer-to-peer” de SQL Server son técnicas que permiten actualizar datos en cualquier sitio *maestro* y funcionarán bien si se prepara un particionamiento horizontal de los datos modificados en cada sitio, para luego propagarlos de forma asíncrona en lugar de síncrona. La replicación transaccional y la replicación por fusión (“merge”) de SQL Server facilitan también el flujo de los cambios desde el nodo Suscriptor al Publicador. La replicación en SQL Server también permite la propagación de cambios sobre el esquema

Base de datos en standby

En Oracle y SQL Server hay una función relacionada con la escalabilidad consistente en dejar las bases de datos en modo standby, únicamente accesibles para lectura con el fin de segregarse de la base de datos en producción aquellas tareas que consumen recursos de forma intensiva, como son la generación de informes o el soporte a la decisión. La base de datos standby lógica en Oracle permite generar nuevos objetos en el esquema de la base de datos, lo que aumenta su flexibilidad de cara a ciertos escenarios de uso. La base de datos standby de SQL Server, una instantánea de base de datos, no puede modificarse de ningún modo.

Soporte para AWE

SQL Server y Oracle sobre Windows pueden configurarse para compatibilidad con AWE, lo que les permite direccionar memoria física más allá del límite de 4 Gb por proceso que imponen los sistemas de 32 bits.

Soporte para arquitecturas de 64 Bits

La arquitectura de 64 bits ofrece ventajas sustanciales desde los niveles del hardware y sistema operativo que tienen un impacto muy positivo en la escalabilidad y rendimiento de los sistemas de bases de datos. A cada escalón ascendido desde los sistemas de 8 bits, 16 bits y 32 bits hemos podido comprobar estos beneficios. La escalabilidad procede de la capacidad para utilizar direcciones de memoria de 64 bits, es decir, espacios de direccionamiento de 16 millones de Gigabytes.

El tamaño de los archivos ha pasado de 2 Gb máximo en los sistemas de archivos de 16 bits a 4 Gb en los de 32 bits y a 16 exabytes (EB) en los actuales de 64 bits. Puesto que hay más memoria disponible y se puede mantener en ella más información, se reducen las operaciones de E/S a disco, paginación y swap, con lo que mejora sensiblemente el rendimiento. También se obtienen mejoras en el rendimiento de los procesadores a nivel de cálculo y operaciones de coma flotante, y todo ello mejora sustancialmente la velocidad de los sistemas. No obstante, las cifras indicadas son solamente teóricas, ya que en realidad el tamaño del espacio de memoria direccionable y accesible viene delimitado por el sistema operativo y la arquitectura del hardware.

La única diferencia entre Oracle y SQL Server 2008, utilizando el mismo código base para las versiones de 32 bits y 64 bits del software de base de datos, está en la compilación. Las funcionalidades de ambas versiones son parecidas. SQL Server 2008 soporta los procesadores de 64 bits Opteron y Athlon 64 de AMD, así como los Intel Xeon con tecnología EMT64. En sistemas de 64 bits, SQL Server puede utilizar hasta 32 terabytes (TB) de memoria y soporta también procesadores multi-core y hyper-threading.

NUMA

SQL Server es compatible con arquitecturas NUMA (Non-Uniform Memory Access). La principal ventaja de esta arquitectura es su escalabilidad: NUMA supera los límites de la arquitectura SMP (Multiprocesamiento Simétrico). Con SMP todos los accesos a memoria se transportan por el mismo bus de memoria compartida. Esto funciona bien para un número reducido de CPUs, pero el problema con el bus compartido aparece cuando tenemos decenas, o incluso cientos de CPUs compitiendo por el acceso a este canal. NUMA resuelve estos cuellos de botella limitando el número de CPUs en cada uno de los buses de memoria y conectando los distintos nodos (memoria local y remota) por medio de conexiones de alta velocidad.

SOA

La Arquitectura Orientada a Servicios (SOA, Service Oriented Architecture) supone un gran avance para las aplicaciones al vincular estrechamente las aplicaciones asíncronas para mejorar su escalabilidad y su rendimiento. Algunos de los pilares básicos del diseño de SOA son el uso de colas, mensajes y servicios.

Broker de Servicio de SQL Server

SQL Server 2005 introdujo el concepto de “intermediario” (broker), el Service Broker de SQL Server que soporta diversas funcionalidades necesarias para la construcción de aplicaciones basadas en SOA:

- Un potente modelo de programación asíncrona basado en servicios de Service Broker, colas y mensajes.
- Un sistema de mensajería y encolamiento fiable.

Las aplicaciones y procedimientos almacenados (en Transact-SQL o CLR) pueden aprovechar las funciones del Service Broker. Normalmente una aplicación o un procedimiento almacenado envían un mensaje al servicio Service Broker (con el cual se inicia una conversación). El servicio después coloca el mensaje en la cola correspondiente y finalmente la cola activa un procedimiento almacenado (que extrae el mensaje de la cola y lo procesa). El procedimiento almacenado puede también utilizarse para validar el mensaje y enviar una respuesta a la aplicación y finalizar la conversación. La aplicación recibe la respuesta y finaliza el diálogo. El Service Broker proporciona la infraestructura necesaria para conseguir un entorno de enrutamiento de mensajes, conversaciones, colas, servicios y activación fiable y seguro.

Las ventajas más importantes que ofrece el Service Broker de SQL Server son:

- Los mensajes se pueden procesar como transacciones de SQL Server para garantizar la integridad de los datos
- Se garantiza que los mensajes solo se procesan una vez, y en el mismo orden
- El broker garantiza la fiabilidad de la entrega de mensajes.
- Las conversaciones, junto con los datos, se mantienen intactos en caso de reinicio del sistema, fallos a nivel de servidor/instancia y fallos de la red.

- Facilita el desarrollo rápido y flexible de aplicaciones, al estar integrado el Service Broker dentro de SQL Server Management Studio.
- Su monitorización es sencilla utilizando contadores de monitorización del sistema y utilidades de diagnóstico.

Oracle Streams Advanced Queuing

Oracle ofrece unas funcionalidades similares al Service Broker con Oracle Streams Advanced Queuing (AQ). Dentro de un AQ de Oracle Streams, las aplicaciones productoras encolan mensajes y las consumidoras los extraen. Estos mensajes pueden verse sometidos a ciertas transformaciones durante los procesos de encolamiento y desencolamiento. Oracle Streams AQ proporciona los elementos de soporte necesarios para el encolamiento, extracción, transformación, seguridad, planificación, control de flujo y notificaciones asíncronas. Para el desarrollo de aplicaciones basadas en Oracle Streams AQ se utiliza el entorno Integrated Development Environment de Oracle, y dentro de las herramientas de gestión existen distintas utilidades de monitorización.

Conclusión

Microsoft SQL Server 2008 es una solución de gestión de datos completa para todos los usuarios corporativos, y ofrece una plataforma segura, consistente y productiva para la gestión integral de los datos y aplicaciones de Business Intelligence (BI). SQL Server 2008 mantiene la tradición de Microsoft de ofrecer herramientas potentes y familiares a los profesionales de IT y usuarios finales cuyo objetivo es aumentar el valor de negocio mediante una reducción del Coste Total de Propiedad (TCO), una mayor facilidad de uso y capacidades de integración amplias. SQL Server 2008 mejora además sus servicios e infraestructura extendiendo las posibilidades de los sistemas de business intelligence con un esfuerzo de desarrollo e integración mucho menor que con cualquier otro producto.

Más información disponible en:

<http://www.microsoft.com/sqlserver/>: Sitio Web de SQL Server

<http://technet.microsoft.com/en-us/sqlserver/>: TechCenter de SQL Server

<http://msdn.microsoft.com/en-us/sqlserver/>: DevCenter de SQL Server

¿Le ha sido útil este documento? Envíenos sus comentarios. Valore su interés en una escala de 1 (pobre) a 5 (excelente), e indíquenos por qué le asigna esta puntuación. Por ejemplo:

- ¿Lo valora con nota alta porque incluye buenos ejemplos, capturas de pantalla, explicaciones claras o por algún otro motivo?
- ¿Lo puntúa bajo porque no incluye ejemplos, es confuso o los gráficos no se entienden bien?
-

Sus comentarios nos ayudarán a mejorar la calidad de la documentación técnica que publicamos.

[Envíenos sus comentarios.](#)