

Diez consejos para la optimización del rendimiento de SQL Server

Escrito por Patrick O'Keefe y Richard Douglas



Resumen

La optimización de rendimiento de SQL Server puede ser todo un desafío. Se ofrece una gran cantidad de información sobre cómo abordar los problemas de rendimiento en general. Sin embargo, no hay mucha información detallada para problemas específicos y aún menos información sobre cómo aplicar esos conocimientos específicos en su propio entorno.

En este documento se ofrecen 10 consejos para la optimización del rendimiento de SQL Server, teniendo como modelo SQL Server 2008 y 2012. No hay listas definitivas de los consejos más importantes, pero no puede equivocarse si comienza por aquí.

Como un DBA, seguramente tiene su propio punto de vista y sus propios consejos, trucos y scripts de optimización favoritos. ¿Por qué no se une al debate en [SQLServerPedia?](#)

Introducción

Tenga en cuenta sus objetivos de optimización.

Todos deseamos obtener el mayor valor de nuestras implementaciones de SQL Server. Al aumentar la eficacia de los servidores de la base de datos se liberan recursos del sistema para otras tareas, como informes empresariales y consultas ad hoc. Para obtener el mejor rendimiento para la inversión en hardware de la organización, debe asegurarse de que el SQL o la carga de trabajo de la aplicación que se ejecuta en los servidores de la base de datos se ejecuten tan rápido y eficientemente como sea posible.

Pero la optimización depende de sus objetivos. Quizás se pregunte "¿Obtengo la mejor eficacia de mi implementación de SQL Server?". Y quizás otra persona se pregunte "¿Mi aplicación podrá escalar?". Aquí presentamos las maneras principales de optimizar un sistema:

- Optimización para cumplir con el acuerdo de nivel de servicio (SLA) o los objetivos del indicador clave de rendimiento (KPI)
- Optimización para mejorar la eficacia, con el fin de liberar recursos para otros propósitos
- Optimización para garantizar escalabilidad, con el fin de ayudar a mantener los SLA o los KPI en el futuro

Trabaje en la maximización de la escalabilidad y la eficacia de todos los servidores de bases de datos, incluso cuando se cumplen los requisitos empresariales.

Recuerde que la optimización es un proceso continuo, no un arreglo de una sola vez.

La optimización del rendimiento es un proceso continuo. Por ejemplo, cuando realiza una optimización para los objetivos de SLA, puede "terminar". Sin embargo, si la optimización es para mejorar la eficacia o garantizar la escalabilidad, nunca termina su trabajo realmente; este tipo de optimización debe continuarse hasta que el rendimiento sea "suficientemente bueno". En el futuro, cuando el rendimiento de la aplicación ya no sea lo suficientemente bueno, debe realizarse nuevamente el ciclo de optimización.

"Suficientemente bueno" se define generalmente según los imperativos empresariales, como los SLA o los requisitos de rendimiento del sistema. Aparte de esos requisitos, debe contar con la motivación para maximizar la escalabilidad y la eficacia de todos los servidores de la base de datos, incluso si se cumplen los requisitos empresariales actualmente.

Acerca de este documento

La optimización del rendimiento en SQL Server es un desafío. Hay muchos recursos de información generalizada sobre diversos puntos de datos (por ejemplo, contadores de rendimiento y objetos de administración dinámica [DMO]), pero hay muy poca información respecto a qué hacer con estos datos y cómo interpretarlos. En este documento, se describen 10 consejos importantes que serán útiles en las trincheras y que le permiten convertir algunos de estos datos en información accionable.

Consejo n.º 10. La metodología "Línea de base y evaluación comparativa" le permite detectar problemas.

Información general sobre el proceso de línea de base y evaluación comparativa La Figura 1 muestra los pasos del proceso de línea de base. En las siguientes secciones se explican los pasos clave del proceso.

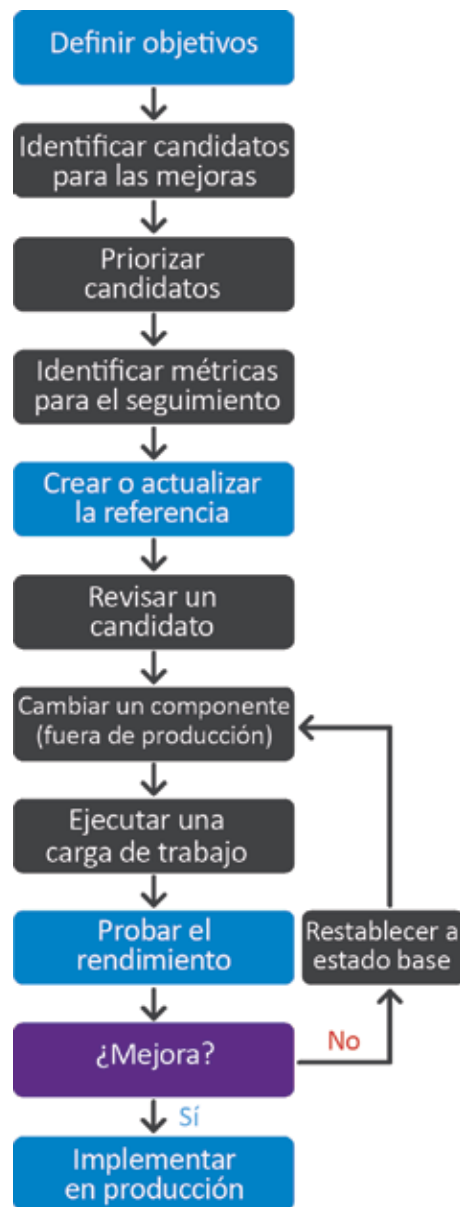


Figura 1. El proceso de línea de base



Antes de comenzar a realizar cambios, obtenga un registro del rendimiento actual del sistema. Cuando se trata de configurar y optimizar, la mayoría nos vemos tentados a comenzar a cambiar todo de inmediato. ¿Nunca buscó su auto por el mecánico y se preguntó si en realidad funciona peor que antes? Le gustaría quejarse, pero no está seguro. También es probable que se pregunte si el nuevo problema que cree observar se debe a algo que hizo el mecánico o es algo que sucedió luego. Lamentablemente, el rendimiento de la base de datos puede sufrir los mismos problemas.

Después de leer este documento, tendrá muchas ideas y deseará implementar la estrategia de línea de base y evaluación comparativa de inmediato. El primer paso que debe dar no es el más atractivo, pero sin dudas es el más importante: es el momento de determinar en qué condiciones se encuentra el entorno en comparación con los criterios que planea modificar.

Determine sus objetivos.

Antes de hacer nada en el sistema, decida qué desea conseguir. ¿Hay aspectos de los SLA relacionados con el rendimiento, el consumo de recursos o la capacidad que desea abordar? ¿Está abordando un problema actual en producción? ¿Ha habido quejas con respecto al tiempo de los recursos? Establezca algunos objetivos claros.

La mayoría cuenta con muchas bases de datos e instancias de las que ocuparse. Para maximizar el valor de sus esfuerzos, debe pensar cuidadosamente qué se necesita para un sistema en particular con el fin de que tenga un buen rendimiento y cumpla con las expectativas de los usuarios. Si realiza un análisis y una optimización demasiado extensos, descubrirá que invierte una cantidad desproporcionada de tiempo en sistemas de baja prioridad, lo que va en detrimento de los sistemas de producción principales. Sea claro precisamente en lo que desea obtener a través de sus esfuerzos de medición y optimización. Priorícelos e, idealmente, obtenga el consenso y la inversión de un patrocinador empresarial.

Establezca la norma.

Una vez que detectó qué desea obtener, entonces debe decidir cómo medirá el resultado. ¿Cuáles son los contadores de sistema operativo, los contadores de SQL Server, las medidas de recursos y los otros puntos de datos que le proporcionarán la visibilidad necesaria?

Una vez que tenga esa lista, debe establecer su línea de base o, en otras palabras, el rendimiento típico del sistema en comparación con los criterios elegidos. Debe recopilar suficientes datos durante un período lo suficientemente largo para obtener una muestra representativa del rendimiento típico del sistema. Una vez que tenga los datos, puede sacar un promedio de los valores durante el período para establecer su primera línea de base. Después de realizar modificaciones en el sistema, debe realizar una nueva evaluación comparativa y compararla con la original para poder medir objetivamente el efecto de las modificaciones.

No realice un seguimiento de los valores promedio únicamente, también debe realizar un seguimiento de las desviaciones de la norma. Aun así, es importante ser cuidadoso con los promedios. Por lo menos, calcule la desviación estándar de cada contador para obtener una indicación de la variación en el tiempo. Piense en el montañista al que se le informa que el diámetro promedio de la soga es de 1 cm. El montañista salta con confianza hasta el otro lado. Pasa por encima de varios metros de rocas filosas y sonríe con satisfacción. Luego se le informa que la sección más gruesa de la soga es de 2 cm y la más delgada es de 0,1 cm. ¡Lo sentimos!

Si no está familiarizado con la desviación estándar, consulte un libro de estadísticas para principiantes. No necesita estudiar demasiado, pero sí es útil conocer lo básico.

El mensaje clave aquí es no estudiar los promedios únicamente, sino también estudiar la desviación de la norma (la media). Decida cuál debe ser la norma (esta información suele encontrarse en los SLA). Su misión no es obtener el mejor rendimiento posible, sino cumplir con sus objetivos de rendimiento de la mejor manera posible y luego limitar la desviación de esos objetivos en la mayor medida posible. Cualquier otra

Las evaluaciones comparativas le permiten detectar el comportamiento anómalo porque usted cuenta con un buen indicador de qué es el comportamiento normal.

Su misión no es obtener el mejor rendimiento posible, sino cumplir con sus objetivos de rendimiento de la mejor manera posible y luego limitar la desviación de esos objetivos en la mayor medida posible.

cosa que haga será tiempo y esfuerzo malgastados y quizás también resulte en la utilización ineficiente de los recursos de la infraestructura.

¿Cuántos datos se necesitan para una línea de base? La cantidad de datos necesaria para establecer una línea de base depende de cuánto varía la carga en el tiempo. Consulte a los administradores de sistemas, los usuarios finales y los administradores de aplicaciones. Generalmente tienen una idea acertada de cuáles son los patrones de uso. Debe reunir la suficiente cantidad de datos para cubrir los períodos promedio, pico y no pico. Es importante medir la cantidad en la que varía la carga y con cuánta frecuencia. Se necesitan menos datos para los sistemas con patrones predecibles. Mientras más variación haya, el intervalo de medición será menor y deberá realizar más mediciones para obtener una línea de base confiable. Para desarrollar un poco más la analogía del montañista, mientras mayor sea la longitud de la soga que examina, más chances tendrá de detectar las variaciones. La criticidad del sistema y el impacto que tendría una falla de este también afectan la cantidad de investigación y la confiabilidad de la muestra.

Almacenamiento de los datos

Mientras más parámetros tenga en cuenta y menor sea la frecuencia, mayor será el conjunto de datos recopilado. Quizás parezca obvio, pero debe tener en cuenta la capacidad necesaria para almacenar los datos de las mediciones. Una vez que cuente con algunos datos, debería ser bastante sencillo extrapolar el crecimiento del repositorio en el tiempo. Si realizará la supervisión durante un período extenso, considere el agregado de datos históricos en intervalos con el fin de evitar un crecimiento desmedido del repositorio.

Por motivos de rendimiento, el repositorio de medidas no debe alojarse

en el mismo sitio que la base de datos que está supervisando.

Limite la cantidad de cambios que realiza en determinado momento.

Intente limitar la cantidad de modificaciones que realiza entre cada evaluación comparativa. Ordene las modificaciones para probar una hipótesis en particular. Esto le permitirá confirmar o descartar cada candidato para las mejoras. Cuando se centre en una solución, comprenderá exactamente por qué está analizando los cambios del comportamiento, y comprobará que, con frecuencia, esto revela una serie adicional de potenciales opciones de optimización.

Análisis de los datos

Una vez que realizó los cambios en el sistema, querrá determinar si tuvieron el efecto deseado. Para ello, repita las mediciones que tomó en la línea de base original en una escala de tiempo representativa y similar. Luego, puede comparar las dos líneas de base para:

- **Determine si los cambios tuvieron el efecto deseado:** Cuando modifica la configuración, optimiza un índice o cambia el código SQL, la línea de base le permite saber si el cambio tuvo el efecto deseado. Si recibe alguna queja porque el rendimiento es más lento, puede saber con seguridad que la instrucción es precisa desde el punto de vista de la base de datos.

El error más frecuente que cometen la mayoría de los DBA sin mucha experiencia es realizar conclusiones precipitadas. A menudo verá que alguien salta de la felicidad cuando nota un aumento instantáneo y notable en el rendimiento después de hacer uno o más cambios. Lo implementa en producción y comienza a mandar correos con información de la resolución del problema. Pero estos festejos pueden ser cortos cuando los mismos problemas vuelven a surgir después de un tiempo o se genera otro problema a partir de efectos secundarios no conocidos. Con frecuencia el resultado es una situación menos deseable que la original. Cuando piense que encontró la respuesta a un problema, póngala a prueba y realice una evaluación comparativa con la línea de base. Esta es la única manera confiable de saber si tuvo algún progreso.

- **Determine si un cambio tuvo algún efecto secundario no esperado:** Una línea de base también le permite ver objetivamente si un cambio afectó un contador o una medición que no esperaba que afectara.
- **Anticipe los problemas antes de que ocurran:** Con una línea de base puede establecer normas de rendimiento precisas contra las condiciones de carga típicas. Esto le permite predecir si habrá problemas en el futuro y cuándo, basado en la tendencia actual de consumo de recursos o las cargas de trabajo proyectadas para los escenarios futuros. Por ejemplo, realiza una planificación de capacidad: Puede predecir cuándo los sistemas se encontrarán con un cuello de botella en la conexión de usuarios si extrapola el consumo de recursos típico por usuario conectado.
- **Solucione problemas de manera más eficaz:** ¿Alguna vez estuvo varios días y noches luchando con un problema de rendimiento en la base de datos y descubrió que no estaba relacionado para nada con la base de datos? Al establecer una línea de base es más sencillo eliminar la instancia de base de datos y detectar la causa. Por ejemplo, suponga que el consumo de memoria aumentó mucho repentinamente. Puede observar que la cantidad de conexiones aumentó mucho y se encuentra muy por encima de la línea de base. Consulta la situación con el administrador de aplicaciones y le confirma que se implementó un nuevo módulo en eStore. No pasa demasiado tiempo hasta que se descubre que el nuevo desarrollador con poca experiencia escribe código que no libera las conexiones de base de datos como debería. Apuesto a que puede pensar en más escenarios similares.

Si se descartan las cosas que NO son responsables de un problema, se puede ahorrar mucho tiempo, ya que se despeja el panorama y se obtiene una vista más precisa de la causa del problema. Hay muchos ejemplos en los que puede comparar los contadores del sistema con los contadores de SQL Server para confirmar o descartar si la base de datos es la causa de un problema. Una vez que se descartan los sospechosos de siempre, puede comenzar a buscar las desviaciones significativas de la línea de base, reunir los indicadores relacionados y comenzar a profundizar en la causa raíz.

Repita el proceso de línea de base con tanta frecuencia como sea necesario.

Una buena optimización consiste en un proceso repetitivo y científico. Los consejos que se ofrecen en este documento son un buen punto de partida, pero son solo eso: un punto de partida. La optimización del rendimiento es un proceso altamente personalizado y está determinado por el diseño, la composición y el uso de cada sistema individual.

La metodología de línea de base y evaluación comparativa es la piedra angular de una optimización del rendimiento buena y confiable. Proporciona el mapa, una referencia y las coordenadas que se necesitan para descubrir qué necesitamos para llegar y cómo llegar, con el fin de garantizar que no nos perdamos en el camino. Un enfoque estructurado nos permite construir un rendimiento confiable y uniforme en la cartera de base de datos.

Consejo n.º 9. Los contadores de rendimiento le proporcionan información rápida y útil sobre las operaciones que actualmente están en ejecución.

Razones para supervisar los contadores de rendimiento

Una pregunta muy común con respecto a la optimización del rendimiento de SQL Server es: "¿Qué contadores debo supervisar?". En lo que respecta a la administración de SQL Server, hay dos motivos importantes para la supervisión de los contadores de rendimiento:

- Cada vez mayor eficiencia operativa
- Prevención de los cuellos de botella

Si bien coinciden en algunos puntos, estos dos motivos le permiten elegir con facilidad puntos de datos para supervisar.

Supervisión de los contadores de rendimiento para mejorar la eficiencia operativa

La supervisión operativa comprueba el uso general de recursos. Ayuda a responder a preguntas como:

- El servidor que se está por ejecutar, ¿se quedará sin recursos como la CPU, espacio en disco o memoria?
- Los archivos de datos, ¿pueden crecer?
- Los archivos de datos de tamaño fijo, ¿tienen suficiente espacio para los datos?

La cantidad de datos necesaria para establecer una línea de base depende de cuánto varía la carga en el tiempo.

Limite la cantidad de cambios que realiza entre cada evaluación comparativa para poder evaluar los efectos de cada cambio de manera meticulosa.

También puede usar esos datos para determinar las tendencias. Un buen ejemplo sería recopilar los tamaños de todos los archivos de datos para determinar la tendencia de las tasas de crecimiento y poder predecir los requisitos para los recursos en el futuro.

Para responder a las tres preguntas anteriores, debe observar los siguientes contadores:

Contador	Le permite
Processor\%Processor Time	Supervisar el consumo de la CPU en el servidor
LogicalDisk\Free MB	Supervisar el espacio disponible en los discos
MSSQL\$Instance:Databases\DataFile(s) Size (KB)	Ver el crecimiento de la tendencia en el tiempo
Memory\Pages/sec	Comprobar la paginación, que es un buen indicador de que los recursos de memoria podrían no ser suficientes
Memory\Available MBytes	Ver la cantidad de memoria física disponible para que utilice el sistema

Supervisión de contadores de rendimiento para evitar cuellos de botella
 La supervisión de cuellos de botella se enfoca más en los problemas relacionados con el rendimiento. Los datos recopilados lo ayudan a responder a preguntas como:

- Los subsistemas más importantes de SQL Server, como la caché de búfer y la caché de plan, ¿se encuentran en buen estado?
- ¿Hay contención en la base de datos?

Para responder a estas preguntas, observe los siguientes contadores:

- ¿Hay un cuello de botella de la CPU?
- ¿Hay un cuello de botella de E/S?

Contador	Le permite
Processor\%Processor Time	Mediante la supervisión del consumo de la CPU puede comprobar si hay un cuello de botella en el servidor (se indica a través de un uso alto y sostenido).
High percentage of Signal Wait	La espera de señal es el tiempo que un trabajador debe esperar la CPU después de que ha finalizado la espera de otra cosa (como un bloqueo, un pestillo u otra espera). El tiempo que debe esperarse la CPU es un indicador de un cuello de botella de una CPU. La espera de señal puede encontrarse al ejecutar DBCC SQLPERF(waitstats) en SQL Server 2000 o al consultar sys.dm_os_wait_stats en SQL Server 2005.
Physical Disk\Avg. Disk Queue Length	Comprobar si hay cuellos de botella en los discos: Si el valor supera a 2 es posible que haya un cuello de botella.
MSSQL\$Instance:Buffer Manager\Page Life Expectancy	La expectativa de vida de la página es la cantidad de segundos que se conserva una página en la caché del búfer. Si el valor es un número bajo, las páginas se borran después de un período corto en la caché, lo que reduce la eficacia de esta.
MSSQL\$Instance:Plan Cache\Cache Hit Ratio	Si el valor de la relación de aciertos de la caché de plan es bajo, los planes no se están reutilizando.
MSSQL\$Instance:General Statistics\Processes Blocked	Los bloques grandes son indicadores de contención de recursos.



Consejo n.º 8. La modificación de la configuración del servidor puede proporcionar un entorno más estable.

El cambio de la configuración dentro de un producto para obtener una mayor estabilidad puede sonar contraintuitivo, pero en este caso realmente funciona. Como un DBA, su trabajo consiste en garantizar un nivel de rendimiento uniforme para los usuarios cuando solicitan datos de las aplicaciones. Si no se modifica la configuración descrita en el resto de este documento, es posible que se experimenten escenarios que provoquen la reducción del rendimiento de los usuarios sin ninguna advertencia. Estas opciones pueden encontrarse fácilmente en `sys.configurations`, que detalla las configuraciones disponibles al nivel de los servidores, junto con información adicional. En el atributo `Is_Dynamic` en `sys.configurations` se muestra si la instancia de SQL Server deberá reiniciarse después de realizar una modificación en la configuración. Para realizar el cambio, debería llamar al procedimiento `sp_configure` almacenado con los parámetros relevantes.

La configuración de memoria mínima y máxima puede garantizar un nivel determinado de rendimiento.

Suponga que tiene un clúster Activo/Activo (o un solo host con varias instancias). Podemos realizar algunas modificaciones en la configuración para garantizar que podamos cumplir los SLA en el caso de una conmutación por error en la que ambas instancias estén alojadas en la misma caja física.

En este caso, deberíamos cambiar la configuración de memoria mínima y máxima para garantizar que el host físico tenga la cantidad de memoria suficiente para encargarse de cada instancia sin realizar intentos constantes de recortar de forma drástica el conjunto en funcionamiento de la otra instancia. Se puede realizar una modificación similar en la configuración para usar determinados procesadores con el fin de garantizar cierto nivel de rendimiento.

Es importante tener en cuenta que la configuración de la memoria máxima no solo es adecuada para las instancias en un clúster sino también para las instancias que comparten recursos con cualquier otra aplicación. Si el uso de la memoria de SQL Server es muy alto, el sistema operativo podría recortar notablemente la cantidad de memoria que puede usar con el fin de permitir que el sistema operativo o las aplicaciones puedan funcionar.

- **SQL Server 2008:** En SQL Server 2008 R2 y las versiones anteriores, la configuración de memoria mínima y máxima restringe solo la cantidad de memoria que utiliza el bloque de búferes (más específicamente solo ubicaciones simples de páginas de 8 KB). Esto significa que, si ejecuta procesos fuera del bloque de búferes (como procedimientos extensos almacenados, CLR u otros componentes como Servicios de integración, Servicios de generación de informes o Servicios de análisis), necesita reducir aún más este valor.
- **SQL Server 2012:** En SQL Server 2012 las cosas varían un poco, ya que hay un administrador de memoria central. En este administrador de memoria ahora se incorporan ubicaciones multipágina, como páginas de datos grandes y planes en caché con un tamaño mayor que 8 KB. Este espacio de memoria ahora incluye alguna funcionalidad de CLR.

Dos opciones de servidor que pueden colaborar con el rendimiento de manera indirecta.

No hay opciones que colaboren directamente con el rendimiento, pero hay dos opciones que pueden colaborar de manera indirecta.

- **Compresión de copias de seguridad de manera predeterminada:** Esta opción establece que las copias de seguridad se compriman de manera predeterminada. Si bien esto puede renovar ciclos de las CPU adicionales durante la compresión, en general se usan menos ciclos de CPU en total cuando se lo compara con una copia de seguridad sin comprimir, ya que se escriben menos datos en el disco. Según su arquitectura de E/S, la configuración de esta opción también podría permitir la reducción de la contención de E/S.
- Es posible que la segunda opción se divulgue en un consejo futuro sobre la caché de plan. Deberá esperar y ver si está incluida en esta lista.

El consejo de Sage sobre la solución de problemas en solo tres palabras: "Eliminar o incriminar".

La supervisión de los contadores de rendimiento lo puede ayudar a aumentar la eficacia operativa y prevenir cuellos de botella.

Consejo n.º 7. Encuentre instrucciones no autorizadas en la caché de plan

Una vez que identifica un cuello de botella, debe detectar la carga de trabajo que causa ese cuello de botella. Esta es una tarea mucho más sencilla desde la presentación de los Objetos de administración dinámica (DMO) en SQL Server 2005. Los usuarios de SQL Server 2000 y las versiones anteriores deberán conformarse con Profiler o Trace (más información en el consejo n.º 6).

Diagnóstico de un cuello de botella de la CPU En SQL Server, si detectó un cuello de botella de la CPU, lo primero que querrá es obtener los mayores consumidores de la CPU del servidor. Esta es una consulta muy sencilla en sys.dm_exec_query_stats:

```
SELECT TOP 50
  qs.total_worker_time / execution_count AS avg_worker_time,
  substring(st.text, (qs.statement_start_offset / 2) + 1,
    ( ( CASE qs.statement_end_offset WHEN -1
      THEN datalength(st.text)
      ELSE qs.statement_end_offset END
    - qs.statement_start_offset) / 2) + 1)
  AS statement_text,
  *
FROM sys.dm_exec_query_stats AS qs
  CROSS APPLY sys.dm_exec_sql_text (qs.sql_handle) AS st
ORDER BY
  avg_worker_time DESC
```

La parte realmente útil de esta consulta es la capacidad de usar cross apply y sys.dm_exec_sql_text para obtener la instrucción SQL con el fin de analizarla.

Diagnóstico de un cuello de botella de E/S Sucede algo similar con un cuello de botella de E/S:

```
SELECT TOP 50
  (total_logical_reads + total_logical_writes) AS total_logical_io,
  (total_logical_reads / execution_count) AS avg_logical_reads,
  (total_logical_writes / execution_count) AS avg_logical_writes,
  (total_physical_reads / execution_count) AS avg_phys_reads,
  substring(st.text,
    (qs.statement_start_offset / 2) + 1,
    ((CASE qs.statement_end_offset WHEN -1
      THEN datalength(st.text)
      ELSE qs.statement_end_offset END
    - qs.statement_start_offset) / 2) + 1)
  AS statement_text,
  *
FROM sys.dm_exec_query_stats AS qs
  CROSS APPLY sys.dm_exec_sql_text (qs.sql_handle) AS st
ORDER BY total_logical_io DESC
```


Consejo n.º 6. SQL Profiler es su amigo

Comprender SQL Server Profiler e Extended Events

SQL Server Profiler es una herramienta nativa incluida en SQL Server. Le permite crear un archivo de seguimiento que captura los eventos que ocurren en SQL Server. Estos seguimientos pueden ser invaluable a la hora de proporcionar información sobre la carga de trabajo y las consultas por bajo rendimiento. En este documento técnico no ahondaremos en detalles sobre cómo utilizar la herramienta Profiler. Para obtener más información sobre el uso de SQL Server Profiler, vea el [tutorial en video en SQLServerPedia](#).

Si bien es cierto que SQL Server Profiler se marcó como obsoleto en SQL Server 2012 en favor de Extended Events, debe tenerse en cuenta que esto es solo para el motor de la base de datos y no para los Servicios de análisis de SQL Server. Profiler aún puede proporcionar una gran visibilidad de cómo funcionan las aplicaciones en tiempo real en muchos entornos de SQL Server.

El uso de Extended Events está fuera del alcance de este documento técnico. Para obtener una descripción detallada de Extended Events consulte el documento técnico ["Cómo utilizar Extended Events y Notificaciones de SQL Server para resolver problemas de rendimiento de manera proactiva"](#). Basta con mencionar que los Extended Events se presentaron en SQL Server 2008 y se actualizaron en SQL Server 2012 para incluir más eventos y una muy esperada interfaz de usuario.

Tenga en cuenta que para la ejecución de Profiler se requiere el permiso ALTER TRACE.

Cómo utilizar SQL Profiler

Aquí se explica cómo construir el proceso de la recopilación de datos en Performance Monitor (Perfmon) y relacionar la información del uso de recursos con los datos en los eventos de SQL Server:

1. Abra Perfmon.
2. Si no tiene un conjunto Recopilador de datos ya configurado cree uno con la opción avanzada y los contadores del consejo n.º 9 como guía. No ejecute el conjunto del recopilador de datos todavía.
3. Abra Profiler.
4. Cree un nuevo seguimiento mediante la especificación de los detalles de la instancia, los eventos y las columnas que desea supervisar, además del destino.
5. Inicie el seguimiento.
6. Vuelva a Perfmon para ejecutar el conjunto del recopilador de datos.
7. Deje ambas sesiones en ejecución hasta que se obtengan los datos necesarios.
8. Detenga el seguimiento de Profiler. Guarde el seguimiento y luego ciérralo.
9. Vuelva a Perfmon y detenga el conjunto de recopilación de datos.
10. Abra el archivo de seguimiento guardado recientemente en Profiler.
11. Haga clic en Archivo y luego Importar datos de rendimiento.
12. Navegue hasta el archivo de datos del Recopilador de datos y seleccione los contadores de rendimiento relevantes.

Ahora podrá ver los contadores de rendimiento en conjunto con el archivo de seguimiento de Profiler (consulte la Figura 2), que permitirá una resolución mucho más rápida de los cuellos de botella.

Consejo adicional: En los pasos anteriores se usa la interfaz del cliente; para ahorrar recursos, un seguimiento desde el servidor es más eficaz. Consulte Libros en línea para obtener información sobre cómo comenzar y detener seguimientos desde el servidor.

La supervisión operativa comprueba el uso general de recursos. La supervisión de cuellos de botella se enfoca más en los problemas relacionados con el rendimiento.

El atributo `Is_Dynamic` en `Sys. Configurations` muestra si la instancia de SQL Server necesitará reiniciarse después de hacer una modificación en la configuración.

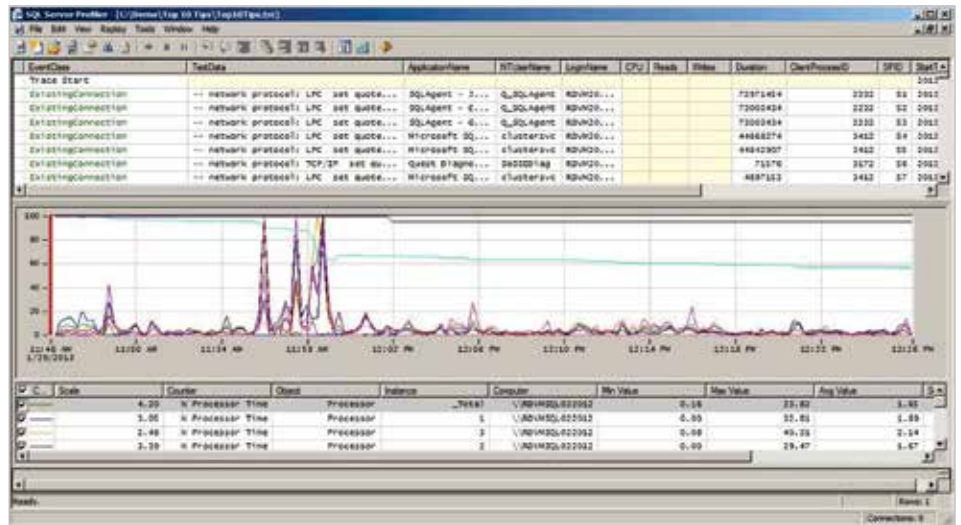


Figura 2. Una vista correlativa de los contadores de rendimiento junto con el archivo de seguimiento de Profiler.

Consejo n.º 5. Configure SAN para el rendimiento de SQL Server.

Las redes de área de almacenamiento (SAN) son geniales. Ofrecen la capacidad de aprovisionar y administrar almacenamiento de manera simple. Las SAN se pueden configurar para obtener un rendimiento veloz desde la perspectiva de SQL Server, pero a menudo no se configuran de este modo. Las organizaciones suelen implementar las SAN por motivos como la consolidación del almacenamiento y la facilidad de la administración, no para el rendimiento. Para empeorar las cosas, en general usted no tiene control directo sobre la manera en que se realiza el aprovisionamiento en una SAN. Entonces, verá que usualmente la SAN se configuró para un volumen lógico en el que tiene que poner todos los archivos de datos.

Las mejores prácticas para la configuración de SAN para el rendimiento de E/S

Agrupar todos los archivos en un solo volumen suele no ser una buena idea si busca el mejor rendimiento de E/S. Como mejores prácticas, querrá:

- Colocar los archivos de registro en su propio volumen, separados de los archivos de datos. Casi con exclusividad, los archivos de registro son de escritura secuencial y no de lectura (con las excepciones de los grupos Creación de reflejo de la base de datos y Siempre disponible). Siempre debe configurarlos para obtener un rendimiento de escritura veloz.
- Coloque `tempdb` en su propio volumen. SQL Server usa `tempdb` internamente para una

infinidad de propósitos, por lo que tenerlo en su propio subsistema de E/S será de utilidad. Para optimizar aún más el rendimiento, primero necesita algunas estadísticas.

- Considere la creación de varios archivos de datos y grupos de archivos en VLDB con el fin de beneficiarse de las operaciones de E/S.
- Ubique las copias de seguridad en sus propias unidades por motivos de redundancia y para reducir la contención de E/S con otros volúmenes durante los periodos de mantenimiento.

Recopilación de datos

Por supuesto que hay contadores de disco de Windows que le proporcionan una idea de lo que Windows cree que sucede. (No olvide configurar los valores sin procesar en la configuración RAID). Los proveedores de SAN suelen proporcionar sus propios datos de rendimiento.

SQL Server también proporciona información de E/S a nivel del archivo:

- Versiones anteriores a SQL 2005: Utilice la función `fn_virtualfilestats`.
- Versiones posteriores: Utilice la función de administración dinámica `sys.dm_io_virtual_file_stats`.

Al usar esta función en el siguiente código puede:

- Derivar tasas de E/S para lecturas y escrituras
- Obtener el rendimiento de E/S
- Obtener el tiempo promedio por E/S
- Observar los tiempos de espera de E/S



```

SELECT db_name (a.database_id) AS [DatabaseName],
       b.name AS [FileName], a.File_ID AS [FileID],
       CASE WHEN a.file_id = 2 THEN 'Log' ELSE 'Data' END AS [FileType],
       a.Num_of_Reads AS [NumReads],
       a.num_of_bytes_read AS [NumBytesRead],
       a.io_stall_read_ms AS [IOStallReadsMS],
       a.num_of_writes AS [NumWrites],
       a.num_of_bytes_written AS [NumBytesWritten],
       a.io_stall_write_ms AS [IOStallWritesMS],
       a.io_stall [TotalIOStallMS],
       DATEADD (ms, -a.sample_ms, GETDATE ()) [LastReset],
       ((a.size_on_disk_bytes / 1024) / 1024.0) AS [SizeOnDiskMB],
       UPPER (LEFT (b.physical_name, 2)) AS [DiskLocation]
FROM sys.dm_io_virtual_file_stats (NULL, NULL) a
     JOIN sys.master_files b
       ON a.file_id = b.file_id AND a.database_id = b.database_id
ORDER BY a.io_stall DESC;

```

Análisis de los datos

Preste especial atención al valor "LastReset" en la consulta; allí se muestra la última vez que se inició el servicio SQL Server. Los datos de los Objetos de administración dinámica no son persistentes, por lo que los datos utilizados para la optimización deben validarse según la cantidad de tiempo de ejecución del servicio. Si no, pueden realizarse conjeturas falsas.

Con estos números puede acotar rápidamente el espectro y ver los archivos responsables del consumo del ancho de banda de E/S y hacerse preguntas como:

- ¿Esta E/S es necesaria? ¿Hay un índice que no estoy viendo?
- El responsable, ¿es una tabla o un índice del archivo? ¿Puedo poner este índice o tabla en otro archivo en otro volumen?

Consejos para la optimización del hardware Si los archivos de base de datos se colocan correctamente y todas las zonas activas de objeto se han identificado y separado en diferentes volúmenes, entonces es hora de ver el hardware en mayor detalle.

La optimización del hardware es un tema para especialistas que no está dentro del alcance de este documento técnico. Sin embargo, hay algunos consejos y mejores prácticas que puedo compartir para facilitar esto:

- No use la unidad de asignación predeterminada cuando cree volúmenes para usar con SQL Server. SQL Server usa tamaños de 64 KB, por lo que este valor debe ser el mínimo.
- Compruebe que las particiones estén correctamente alineadas. Jimmy May escribió un excelente documento técnico sobre este tema. Las particiones mal alineadas pueden reducir el rendimiento hasta en un 30 %.
- Realice una evaluación comparativa de la E/S de su sistema con una herramienta como SQLIO. Puede ver un tutorial sobre cómo utilizar esta herramienta.

Si se configura la compresión de las copias de seguridad para que se realice de manera predeterminada, se reduce la contención de E/S.

DatabaseName	FileName	FileID	FileType	NumReads	NumBytesRead	IOStallReadMS	NumWrites	NumBytesWritten	IOStallWriteMS	TotalIOStall	LastReset	SizeOnDiskMB	DiskLocation
AWTarget2012	Adventureworks2012_Data	1	Data	91	570924	1826	0	73728	2	30626	31/01/2012...	205.000000	C:
NewTest	NewTest	1	Data	36	2113936	9963	1	8192	19	9982	31/01/2012...	5.000000	C:
AWSource2012	Adventureworks2012_Data	1	Data	84	5191072	9731	2	16384	0	9731	31/01/2012...	205.000000	C:
AW3012	Adventureworks2012_Data	1	Data	74	4677632	9151	1	8192	0	9151	31/01/2012...	205.000000	C:

Figura 3. Información de E/S a nivel del archivo en SQL Server



SQL Profiler aún puede proporcionar una gran visibilidad sobre cómo funcionan las aplicaciones en tiempo real en muchos entornos de SQL Server.

Consejo n.º 4. Los cursores y otros T-SQL malos suelen volver para atormentar a las aplicaciones.

Un ejemplo de código malo

En un trabajo anterior descubrí lo que debe ser el peor código que vi en toda mi carrera. El sistema se reemplazó hace mucho tiempo, pero este es un resumen del proceso de la función:

1. Aceptar el valor del parámetro que se eliminará.
2. Aceptar la expresión del parámetro que se eliminará.
3. Encontrar la longitud de la expresión.
4. Cargar la expresión en una variable.
5. Recorrer cada carácter de la variable y comprobar si el carácter coincide con uno de los valores que se eliminará. Si es así, actualizar la variable para eliminarla.
6. Continuar con el próximo carácter hasta que la expresión se haya comprobado en su totalidad.

Si se espantó, entonces está en buena compañía. Claro que estoy intentando explicar el intento que alguien hizo para escribir una instrucción de "REEMPLAZO" de T-SQL.

La peor parte es que esta función se utilizaba para actualizar direcciones como parte de una rutina de correos y se utilizaba miles de veces por día.

La ejecución de un seguimiento de Profiler desde el servidor le permitirá ver la carga de trabajo del servidor y elegir las partes del código que se ejecutan con frecuencia (que es la manera en la que se descubrió esta "gema").

Optimización de consultas mediante planes de consultas El T-SQL malo también puede presentarse como consultas ineficientes que no utilizan índices, mayormente porque el índice es incorrecto o no hay índice. Es de vital importancia tener una buena comprensión de cómo optimizar las consultas mediante planes de consultas.

Un debate detallado sobre la optimización de consultas mediante planes de consulta no está dentro del alcance de este documento técnico. Sin embargo, la manera más sencilla de empezar este proceso es convertir las operaciones SCAN en operaciones SEEK. Las operaciones SCAN leen cada fila de una tabla o del índice, por lo que en el caso de tablas grandes son muy costosas en términos de E/S. En cambio, una operación SEEK usa el índice para ir directamente a la fila deseada, para lo que, obviamente, se necesita un índice. Si encuentra operaciones SCAN en su carga de trabajo, quizás no tenga índices.

Hay muchos buenos libros sobre este tema, incluidos:

- "Professional SQL Server Execution Plan Tuning" de Grant Fritchey.
- "Professional SQL Server 2012 Internals & Troubleshooting" de Christian Bolton, Rob Farley, Glenn Berry, Justin Langford, Gavin Payne, Amit Banerjee, Michael Anderson, James Boother y Steven Wort.
- "T-SQL Fundamentals for Microsoft SQL Server 2012 and SQL Azure" de Itzik Ben-Gan.

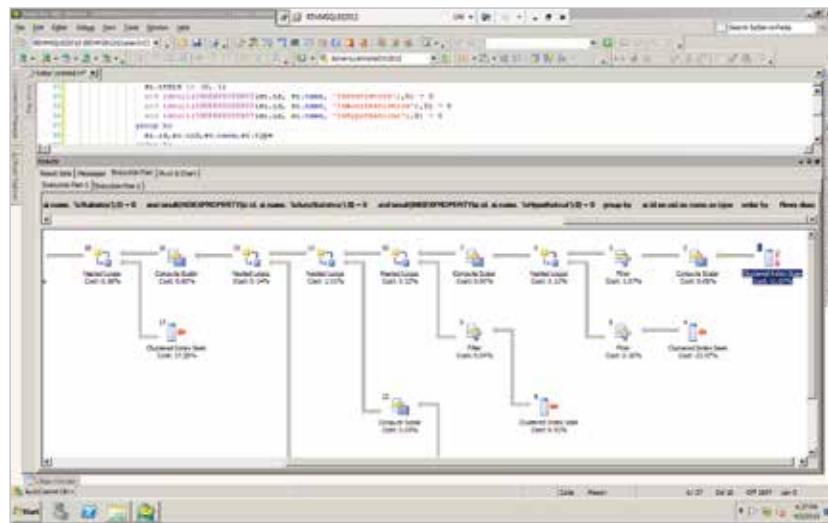


Figura 4. Ejemplo de un plan de consulta grande



Consejo n.º 3. Maximice la reutilización del plan para una mejor caché de SQL Server.

Por qué es importante reutilizar planes de consultas

Antes de ejecutar una instrucción SQL, SQL Server debe crear un plan de consulta primero. Esto define el método que empleará SQL Server para tomar la instrucción lógica de la consulta e implementarla como una acción física en los datos.

Crear un plan de consulta puede requerir un uso significativo de la CPU. Entonces, SQL Server se ejecutará de manera más eficaz si puede reutilizar los planes de consultas en vez de crear uno nuevo cada vez que se ejecuta una instrucción SQL.

$(\text{Batch Requests/sec} - \text{SQL Compilations/sec}) / \text{Batch Requests/sec}$

Abordar una mala reutilización de planes

No es sencillo detectar exactamente qué carga de trabajo es la responsable de la mala reutilización de planes porque el problema suele residir en el código de la aplicación cliente que envía las consultas. Entonces, debe ver el código de la aplicación cliente que envía las consultas.

Para encontrar el código integrado en una aplicación de cliente, debe usar Extended Events o Profiler. Al agregar el evento SQL:StmtRecompile en un seguimiento podrá ver cuando haya un evento de recompilación. (También hay un evento llamado SP:Recompile; este evento se incluye para la compatibilidad hacia atrás desde que se modificó una compilación desde el nivel de procedimiento al nivel de instrucción en SQL Server 2005).

Evaluar si tiene una buena reutilización de planes

En el objeto de rendimiento SQL Statistics hay algunos buenos contadores de rendimiento que le indicarán si cuenta con una buena reutilización de los planes. Esta fórmula le indica la relación de lotes enviados a compilaciones:

Este número debe ser lo más pequeño posible. Una relación de 1:1 significa que cada lote enviado se está compilando y no se reutilizan los planes.

Las SAN se pueden configurar para obtener un rendimiento veloz desde la perspectiva de SQL Server, pero a menudo no se configuran de este modo.

Un problema común es que el código no utiliza instrucciones con parámetros preparadas. Si se utilizan consultas con parámetros, no solo se mejora la reutilización de planes y la carga de la compilación, sino que también se reduce el riesgo de ataque por inyección de código SQL relacionado con el paso de parámetros a través de una concatenación de cadenas. En la Figura 5 se muestran dos ejemplos de código. Si bien son artificiales, muestran la diferencia entre la generación de una instrucción a través de una concatenación de cadenas y la utilización de instrucciones con parámetros preparadas.

La información de E/S a nivel del archivo en SQL Server puede ayudarlo a detectar qué archivos están consumiendo ancho de banda de E/S.

```
public void ExecuteSomeSQL(int aParam) {
    //cmd is a SqlCommand created somewhere else

    cmd.CommandType = CommandType.Text;
    cmd.CommandText = "select foo1, foo2, foo3 from bar where foo1=" + aParam.ToString();

    SqlDataReader dr = cmd.ExecuteReader();
    try {
        while (dr.Read()) {
        }
    } finally {
        dr.Close();
    }
}
```

Malo

```
public void ExecuteSomeSQL(int aParam) {
    //cmd is a SqlCommand created somewhere else

    cmd.CommandType = CommandType.Text;
    cmd.CommandText = "select foo1, foo2, foo3 from bar where foo1 = @foo1";

    cmd.Parameters["@foo1"].Value = aParam;

    SqlDataReader dr = cmd.ExecuteReader();
    try {
        while (dr.Read()) {
        }
    } finally {
        dr.Close();
    }
}
```

Bueno

Figura 5. Comparación de código que genera una instrucción a través de una concatenación de cadenas y código que utiliza instrucciones con parámetros preparadas

SQL Server no puede reutilizar el plan del ejemplo "malo" de la Figura 5. Si el parámetro hubiera sido un tipo de cadena, esta función podría utilizarse para montar un ataque por inyección de código SQL. El ejemplo "bueno" no es vulnerable a un ataque por inyección de código SQL porque se utiliza un parámetro y SQL Server puede reutilizar el plan.

La configuración que faltaba del consejo n.º 8

Las personas con buena memoria recordarán que en el consejo n.º 8 (donde hablamos sobre las modificaciones en la configuración) había un consejo más para ofrecer. SQL Server 2008 presentó una configuración llamada "Optimización para cargas de trabajo ad hoc". Al configurarla, SQL Server almacenará un plan parcial en lugar de un plan completo en la caché de plan. Esto es especialmente útil para los entornos que utilizan código T-SQL construido dinámicamente o Linq que puede ocasionar una no reutilización del código.

La memoria asignada a la caché de plan se encuentra en el bloque de búferes. Por lo tanto, una caché de plan muy llena reduce la cantidad de páginas de datos que se pueden almacenar en la caché de búfer, entonces habrá más viajes de ida y vuelta para buscar datos en el subsistema de E/S, lo que puede ser muy costoso.

Consejo n.º 2. Aprenda cómo leer la caché de búfer de SQL Server y minimizar la hiperpaginación de caché.

Por qué es importante la caché de búfer
Tal como se mencionó anteriormente, la caché de búfer es una gran área de memoria que utiliza SQL Server para reducir la necesidad de realizar E/S física. Ninguna ejecución de consulta de SQL Server lee datos directamente del disco; las páginas de base de datos se leen desde la caché de búfer. Si la página que se busca no está en la caché de búfer, se pone en cola de espera una solicitud de E/S física. Entonces la consulta espera y se busca la página en el disco.

Las modificaciones realizadas a los datos de una página desde una operación ELIMINAR o ACTUALIZAR también se realizan en las páginas de la caché de búfer. Luego, estas modificaciones se eliminan del disco. Todo este mecanismo permite que SQL Server optimice la E/S física de muchas maneras:

- Se pueden leer y escribir varias páginas en una sola operación de E/S.
- Se puede implementar la lectura previa. SQL Server puede notar que para algunos tipos de operaciones puede ser útil leer páginas secuenciales. Se asume que después de leer la página solicitada se desea leer la página siguiente.

Nota: La fragmentación del índice dificulta la capacidad de SQL Server de realizar la optimización de la lectura previa.

Evaluación del estado de la caché de búfer

Hay dos indicadores del estado de la caché de búfer:

- **MSSQL\$Instance:Buffer Manager\Buffer cache hit ratio:** Es la relación entre las páginas encontradas en la caché y las páginas no encontradas en la caché (las páginas que deben leerse desde el disco). Idealmente, este número debe ser lo más pequeño posible. Es posible que tenga una relación de aciertos alta y aun así experimente hiperpaginación de caché.
- **MSSQL\$Instance:Buffer Manager\Page Life Expectancy:** Es la cantidad de tiempo que SQL Server mantiene las páginas en la caché de búfer antes de que se borren. Microsoft dice que una expectativa de vida de página mayor de cinco minutos está bien. Si la expectativa de vida es menor, puede ser un indicador de presión de memoria (no hay suficiente memoria) o hiperpaginación de caché.

Quiero concluir esta sección con una analogía. Muchas personas sostienen que, debido a la innovación de frenos antibloqueo y otras tecnologías de asistencia, las distancias de frenado deben reducirse y que también deben aumentarse los límites de velocidad de acuerdo con esta nueva tecnología. El valor de advertencia de 300 segundos (cinco minutos) para la expectativa de vida de página trae un debate similar en la comunidad de SQL Server. Algunas personas creen que es una regla rápida y severa y otras personas creen que, debido a este aumento en la capacidad de memoria de la mayoría de los servidores de la actualidad, la cifra debería estar en los miles en lugar de las centenas. Para mí, esta diferencia de opiniones resalta la importancia de las líneas de base y el por qué es tan importante tener una comprensión detallada de cuáles deben ser los niveles de advertencia de cada uno de los contadores de rendimiento de su entorno.

Hiperpaginación de caché

Durante un análisis de índices o de tablas grandes, cada página del análisis debe pasar por la caché de búfer, lo que significa que posiblemente las páginas útiles se borren para dar lugar a las páginas que quizás no se lean más de una vez. Esto produce una E/S alta, ya que las páginas borradas deben leerse nuevamente desde el disco. Esta hiperpaginación de caché suele ser un indicador de que se están analizando tablas o índices grandes.

Para determinar qué tablas e índices ocupan más espacio en la caché de búfer, puede examinar la vista de administración dinámica (DMV) `sys.dm_os_buffer_descriptors` (disponible a partir de SQL Server 2005). La consulta de ejemplo que se encuentra a continuación muestra cómo acceder a la lista de tablas o índices

Si los archivos de base de datos se colocan correctamente y todas las zonas activas de objeto se han identificado y separado en diferentes volúmenes, entonces es hora de ver el hardware en mayor detalle.

```
SELECT o.name, i.name, bd.*
FROM sys.dm_os_buffer_descriptors bd
INNER JOIN sys.allocation_units a
    ON bd.allocation_unit_id = a.allocation_unit_id
INNER JOIN
sys.partitions p
    ON (a.container_id = p.hobt_id AND a.type IN (1, 3))
    OR (a.container_id = p.partition_id AND a.type = 2)
INNER JOIN sys.objects o ON p.object_id = o.object_id
INNER JOIN sys.indexes i
    ON p.object_id = i.object_id AND p.index_id = i.index_id
```

que consumen espacio en la caché de búfer en SQL Server:

También puede usar las DMV de índice para detectar qué tablas o índices tienen una gran cantidad de E/S física.

Consejo n.º 1. Comprender cómo se utilizan los índices y encontrar índices malos.

SQL Server 2012 proporciona algunos datos muy útiles sobre los índices, que puede buscar con los DMO implementados en SQL Server 2005.

Uso del DMO

sys.dm_db_index_operational_stats

El objeto sys.dm_db_index_operational_stats contiene información sobre la actividad de E/S, bloqueo, pestillo y método de acceso de bajo nivel actual para cada índice. Use este DMO para responder a las siguientes preguntas:

- ¿Tengo un índice "activo"? ¿Tengo un índice en el que hay contención? Las columnas row lock wait in ms/page lock wait in ms pueden indicarnos si hubo esperas en este índice.
- ¿Tengo un índice que se está usando de manera no eficiente? ¿Qué índices son cuellos de botella de E/S actualmente? La columna page_io_latch_wait_ms puede indicarnos si hubo esperas de E/S al tiempo que lleva páginas de índice a la caché de búfer, lo que es un buen indicador de que hay un patrón de acceso de análisis.

- ¿Qué tipo de patrones de acceso se usan? Las columnas range_scan_count y singleton_lookup_count pueden indicarnos qué clase de patrones de acceso se utilizan en determinado índice.

En la Figura 6, se ilustra el resultado de una consulta que ordena los índices según la espera PAGE_IO_LATCH total. Esto es muy útil cuando intenta determinar qué índices forman parte de los cuellos de botella de E/S.

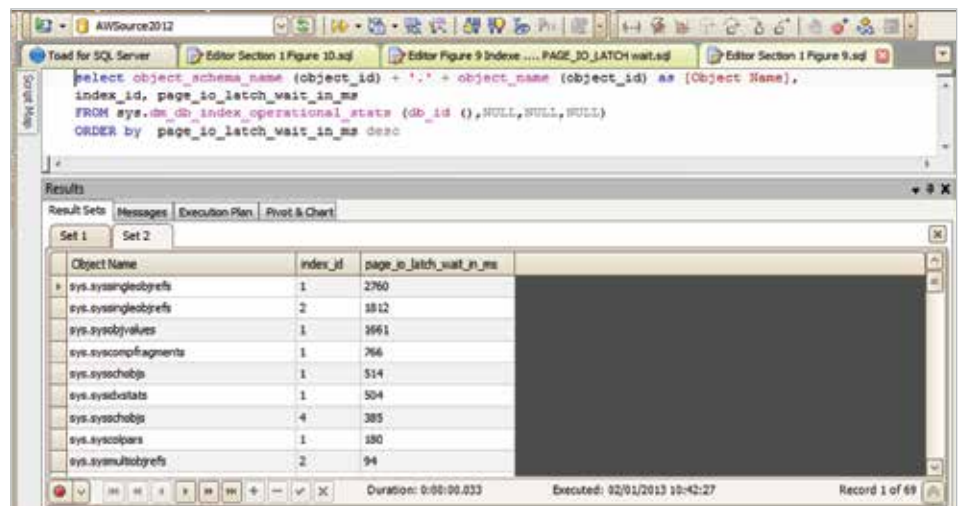
Uso del DMO

sys.dm_db_index_usage_stats

El objeto sys.dm_db_index_usage_stats contiene contadores de diferentes tipos de operaciones de índice y el momento en que se realizó cada operación por última vez. Use esta DMV para responder a las siguientes preguntas:

- ¿Cómo utilizan los índices los usuarios? Las columnas user_seeks, user_scans, user_lookups pueden indicarle los tipos y el significado de las operaciones de usuario en relación con los índices.
- ¿Cuál es el costo de un índice? La columna user_updates puede indicarle el nivel de mantenimiento de un índice.
- ¿Cuándo se utilizó un índice por última vez? Las columnas last_* le indican la última vez que se realizó una operación en un índice.

Si se ejecuta un seguimiento de Profiler del servidor podrá ver la carga de trabajo del servidor y elegir las partes del código que se ejecutan con frecuencia.



The screenshot shows a SQL Server query window with the following query:

```
select object_schema_name (object_id) + '.' + object_name (object_id) as [Object Name],  
index_id, page_io_latch_wait_in_ms  
FROM sys.dm_db_index_operational_stats (db_id (), NULL, NULL, NULL)  
ORDER by page_io_latch_wait_in_ms desc
```

The results are displayed in a table with the following columns: Object Name, index_id, and page_io_latch_wait_in_ms. The data is sorted in descending order of page_io_latch_wait_in_ms.

Object Name	index_id	page_io_latch_wait_in_ms
sys.syssingleobjrefs	1	2760
sys.syssingleobjrefs	2	1812
sys.sysobjvalues	1	1661
sys.syscompfragments	1	766
sys.syschobjs	1	514
sys.sysdvalats	1	504
sys.syschobjs	4	385
sys.syscolpars	1	180
sys.sysmultobjrefs	2	94

Figura 6. Índices ordenados por la espera PAGE_IO_LATCH total

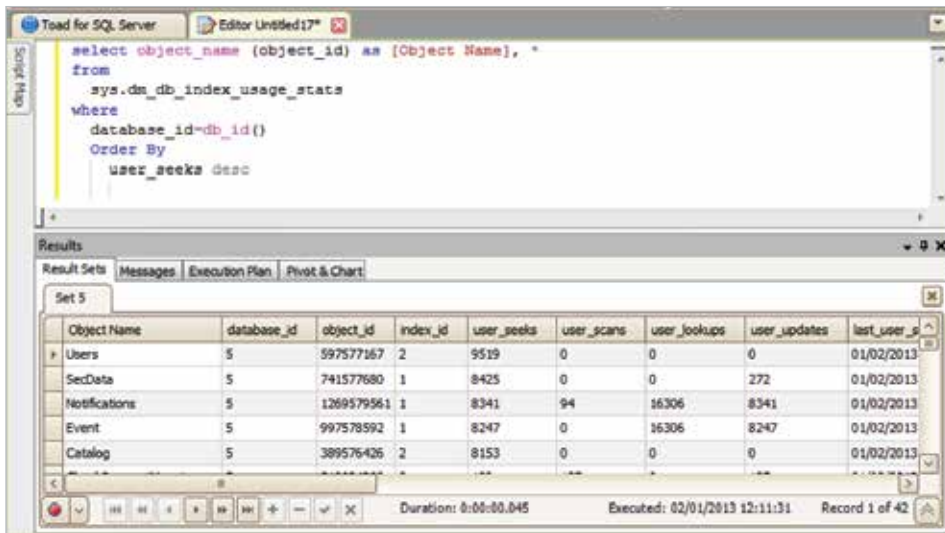


Figura 7. Índices ordenados por la cantidad total de user_seeks

En la Figura 7, se ilustra el resultado de una consulta que ordena los índices según la cantidad total de user_seeks. Si, en cambio, desea identificar los índices que tuvieron una gran proporción de análisis, puede ordenarlos por la columna user_scans. Ahora que tiene un nombre de índice, ¿no sería bueno si pudiera descubrir qué instrucciones SQL utilizan ese índice? En SQL Server 2005 y en las versiones posteriores puede hacerlo.

Por supuesto que hay otras áreas en los índices, como estrategia de diseño, consolidación y mantenimiento. Si desea obtener más información sobre esta área fundamental de la optimización del rendimiento de SQL Server, consulte SQLServerPedia o las transmisiones por Internet y los documentos técnicos de Dell sobre este tema.

Conclusión

Claro que hay muchas más que 10 cosas que debe saber sobre el rendimiento de SQL Server. Sin embargo, en este documento técnico se ofrece un buen punto de partida y algunos consejos prácticos sobre la optimización del rendimiento que puede aplicar en su entorno de SQL Server.

A modo de recapitulación, recuerde estas 10 cosas para la optimización del rendimiento de SQL Server:

10. Las evaluaciones comparativas facilitan las comparaciones de los comportamientos de las cargas de trabajo y le permiten detectar comportamiento anómalo porque usted cuenta con un buen indicador de qué es el comportamiento normal.
9. Los contadores de rendimiento le proporcionan información rápida y útil sobre las operaciones que actualmente están en ejecución.
8. La modificación de la configuración del servidor puede proporcionar un entorno más estable.
7. Los DMO lo ayudan a identificar cuellos de botella de rendimiento rápidamente.
6. Aprenda cómo utilizar SQL Profiler, seguimientos e Extended Events.
5. Las SAN son algo más que cajas negras que realizan operaciones de E/S.
4. Los cursores y otros T-SQL malos suelen volver para atormentar a las aplicaciones.
3. Maximice la reutilización del plan para una mejor caché de SQL Server.
2. Aprenda cómo leer la caché de búfer de SQL Server y minimizar la hiperpaginación de caché.

Y el consejo más importante para la optimización del rendimiento de SQL Server:

1. Investigue cómo se utilizan los índices y cómo encontrar índices malos para convertirse en un experto.

Una expectativa de vida de las páginas menor de cinco minutos puede indicar presión de memoria (no hay suficiente memoria) o hiperpaginación de caché.

Llamado a la acción

Estoy seguro de que no puede esperar para implementar lo que aprendió en este documento técnico. La tabla que se presenta a continuación incluye las acciones con las que debe comenzar el camino hacia un entorno de SQL Server más optimizado:

Si se utilizan consultas con parámetros, no solo se mejora la reutilización de planes y la carga de la compilación, sino que también se reduce el riesgo de ataque por inyección de código SQL relacionado con el paso de parámetros a través de una concatenación de cadenas.

Acción	Subtareas	Fecha fijada
Conseguir la aprobación para comenzar el proyecto	Hable con su administrador de línea y explíquelo que con este proyecto puede ser proactivo en lugar de reactivo.	
Identificar los objetivos de rendimiento	Hable con las partes interesadas de la compañía para determinar los niveles aceptables de rendimiento.	
Establecer una línea de base para el rendimiento del sistema	Recopile los datos relevantes y almacénelos en un repositorio de terceros o construido a medida.	
Identificar los contadores de rendimiento más importantes y configurar los seguimientos o los eventos extendidos	Descargue el contador Perfmon de Dell.	
Revisar la configuración del servidor	Preste especial atención a la memoria y a la configuración "Optimización para cargas de trabajo ad hoc".	
Revisar un subsistema de E/S	Si es necesario, hable con los administradores de SAN y considere la realización de pruebas de carga de E/S con herramientas como SQLIO, o simplemente determine la relación con la que puede realizar tareas intensivas de lectura y escritura, como cuando realiza copias de seguridad.	
Identificar consultas con bajo rendimiento	Analice los datos de los seguimientos, las sesiones de eventos extendidos y la caché de plan.	
Refactorizar código con bajo rendimiento	Consulte las últimas prácticas óptimas en el servicio de blog sindicado de SQLServerPedia.	
Mantenimiento del índice	Asegúrese de que sus índices estén tan optimizados como sea posible.	

Para obtener más información

© 2013 Dell Inc. TODOS LOS DERECHOS RESERVADOS. Este documento contiene información con derechos de propiedad protegida por copyright. Ninguna parte de este documento se puede reproducir ni transmitir de ninguna manera ni por ningún medio, electrónico o mecánico, incluso la grabación o la fotocopia, por ningún propósito, sin el consentimiento por escrito de Dell, Inc. ("Dell").

Dell, Dell Software, los productos y el logotipo de Dell Software —como se los identifica en este documento— son marcas comerciales registradas de Dell, Inc. en los Estados Unidos y en otros países. Todas las otras marcas comerciales y marcas comerciales registradas son propiedad de sus respectivos propietarios.

La información presentada en este documento se proporciona en relación con los productos Dell. En este documento, no se otorga ninguna licencia, expresa o implícita, por impedimento o de lo contrario, a los derechos a la propiedad intelectual o en relación con la venta de los productos Dell. EXCEPTO LO QUE SE ESTABLECE EN LOS TÉRMINOS Y CONDICIONES DE DELL, SEGÚN SE ESPECIFICA EN EL ACUERDO DE LICENCIA PARA ESTE PRODUCTO,

DELL NO ASUME NINGUNA RESPONSABILIDAD Y RENUNCIA A CUALQUIER GARANTÍA EXPRESA, IMPLÍCITA O LEGAL RELACIONADA CON SUS PRODUCTOS, ENTRE ELLOS, LA GARANTÍA IMPLÍCITA DE COMERCIALIZACIÓN, ADECUACIÓN PARA UN PROPÓSITO PARTICULAR Y DE NO INFRACCIÓN. EN NINGÚN CASO DELL SERÁ RESPONSABLE DE DAÑOS DIRECTOS, INDIRECTOS, RESULTANTES, PUNITIVOS, ESPECIALES O INCIDENTALES (QUE INCLUYEN, ENTRE OTROS, DAÑOS POR PÉRDIDA DE BENEFICIOS, INTERRUPCIÓN EMPRESARIAL O PÉRDIDA DE INFORMACIÓN) QUE SURJAN DEL USO O DE LA IMPOSIBILIDAD DE USAR ESTE DOCUMENTO, INCLUSO SI DELL HA SIDO ADVERTIDO DE LA POSIBILIDAD DE TALES DAÑOS. Dell no presenta declaraciones o garantías con respecto a la precisión o integridad del contenido de este documento y se reserva el derecho de realizar cambios a las especificaciones y a las descripciones de los productos en cualquier momento, sin previo aviso. Dell no se hace responsable de actualizar la información contenida en este documento.

Acerca de Dell

Dell Inc. (NASDAQ: DELL) escucha a los clientes y ofrece tecnología innovadora, servicios y soluciones empresariales en todo el mundo que ellos valoran y en los que confían. Para obtener más información, visite www.dell.com.

Si tiene alguna pregunta acerca del uso potencial de este material, comuníquese con:

Software de Dell

5 Polaris Way
Aliso Viejo, CA 92656
www.dell.com.

Consulte nuestro sitio web para obtener información acerca de la oficina regional e internacional.

