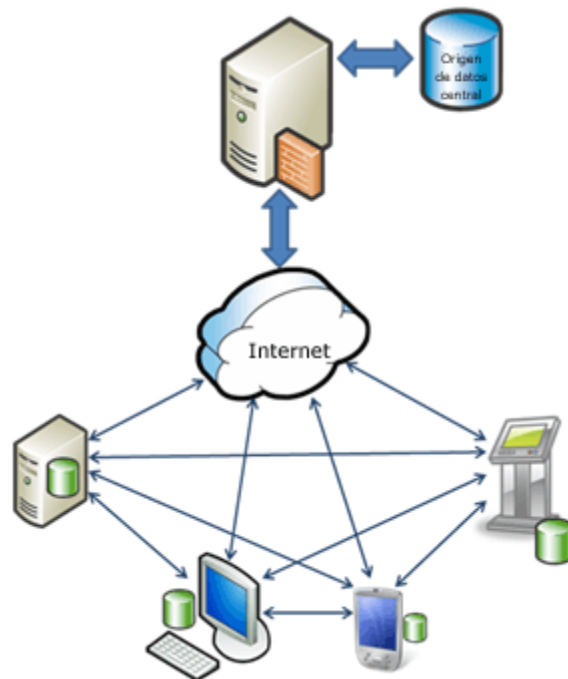


Cómo mejorar el rendimiento de las bases de datos



A la hora de crear una aplicación web dinámica, es fundamental utilizar bases de datos donde guardar la información que vamos a ir mostrando en nuestra aplicación web.

Según la programación que nosotros utilicemos para el desarrollo de la aplicación, podremos utilizar un tipo u otro de base de datos. Tipos hay muchos, pero los más habituales y los que nosotros utilizamos en nuestros servidores son dos: MySQL y Sql Server. Esto no quiere decir, que si el cliente solicita otro motor de bases de datos, no se pueda instalar.



MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario, que pertenecía a la empresa Sun Microsystems y que ahora es controlado por la empresa Oracle, que además de su sistema de base de datos, sigue manteniendo actualizado MySQL. Este sistema de bases de datos, se ha convertido en la más utilizada para las aplicaciones webs, sobre todo al ser software libre, pero bajo una licencia dual, es decir, por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Este tipo de bases de datos puede funcionar en distintos sistemas operativos, independientemente del que usemos.

Algunas características de MySQL son:

- Uso de multihilos mediante hilos del kernel
- Usa tablas en disco b-tree para búsquedas rápidas con compresión de índice
- Tablas hash en memoria temporales
- Completo soporte para operadores y funciones en cláusulas select y where.
- Completo soporte para cláusulas group by y order by, soporte de funciones de agrupación
- Seguridad: ofrece un sistema de contraseñas y privilegios seguro mediante verificación basada en el host y el tráfico de contraseñas está cifrado al conectarse a un servidor.
- Gran cantidad de registros, pudiendo tener tablas de hasta 50 millones de registros.



Por otro lado tenemos SQL Server, que es un sistema de bases de datos creada por Microsoft, basada en el modelo relacional. Esta es la base de datos que se suele utilizar cuando la aplicación desarrollada es programada mediante ASP o ASP.NET, aunque si el desarrollo no es muy grande, se puede llegar a utilizar Access, aunque SQL Server es bastante más potente. Es un sistema exclusivo para Servidores Windows.

Algunas características de este tipo de bases de datos son:

- Soporte de transacciones.
- Escalabilidad, estabilidad y seguridad.
- Soporte para procedimientos almacenados
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite administrar información de otros servidores de datos.

Como hemos comentado, las bases de datos son parte fundamental de las aplicaciones ya que todas las consultas que hagan los clientes, actuarán con estas bases de datos, por lo que es fundamental optimizar lo máximo posible el uso de la base de datos.

Una buena opción es cachear el contenido de las webs, para que si ya se ha realizado una petición a la base de datos, si alguien la vuelve a repetir, la recupere de la caché, en vez de solicitar la información a la base de datos, consumiendo de esta forma menos recursos del servidor.

Esa opción está bien, pero nunca está de más ver opciones para optimizar el servicio de bases de datos.

Optimizar MySQL

Una buena optimización del servidor de bases de datos nos puede llegar a permitir tener cientos de bases de datos con una gran cantidad de registros, que nos puede permitir sacar más rendimiento a nuestro servidor. Veamos algunas ideas que podemos hacer.

1.- Activar la caché de peticiones

Activando este tipo de caché, lo que estamos haciendo es que si una petición se repite, y está en caché, los resultados los saca de ahí, en vez de ejecutar la consulta. Para poder activar la caché de peticiones, nos debemos de ir al fichero de configuración "my.conf" y descomentar la directiva "query_cache_size", indicándole un tamaño que queremos utilizar. Podemos empezar con 32 megas, por ejemplo.

Si hemos realizado esto, si nos conectamos al servidor y ejecutamos el comando "SHOW STATUS LIKE 'qcache%';", podremos ver los resultados que están en caché.

2.- Comprobar peticiones lentas o sin índices

El servidor MySQL nos permite registrar aquellas peticiones que tardan mucho tiempo en ser ejecutadas. Para ello debemos de abrir el fichero de configuración y buscar las siguientes líneas:

```
[mysqld]
; enable the slow query log, default 10 seconds
log-slow-queries
; log queries taking longer than 5 seconds
long_query_time = 5
; log queries that don't use indexes even if they take less than
long_query_time
; MySQL 4.1 and newer only
log-queries-not-using-indexes
```

- Con la primera instrucción activamos el registro de peticiones lentas.
- Con la segunda, "long_query_time", define que guardará un log con toda petición que dure más de 5 segundos.
- Por último "log-queries-not-using-indexes" sirve para registrar en el log las peticiones SQL que no usen índices.

3.- Endurecer los límites

Cuando hablamos de endurecer los límites, nos referimos a modificar la configuración del servidor para que este no se saturate. Podemos hacer referencia a tres variables distintas.

```
set-variable=max_connections=500
set-variable=wait_timeout=10
max_connect_errors = 100
```

-
- Con la primera línea le indicamos el número máximo de conexiones que se pueden establecer en el servidor.
 - Con la segunda línea, le estamos indicando que termine toda conexión que ha estado inactiva durante más de 10 segundos.
 - La tercera línea es por razones de seguridad. Cuando más alto es el valor, más reintentos hará para conectarse en caso de error.

Además de este tipo de optimizaciones, se puede revisar las consultas que se hacen sobre la base de datos para intentar que estas se ejecuten lo más rápidamente posibles.

Al igual que podemos optimizar MySQL, también se puede hacer lo mismo en SQL Server para que este funcione de una forma más óptima.

Optimizar SQL Server

1.- Memoria

SQL Server divide la memoria en dos espacios distintos, la cache de procedimientos y la cache de datos. El SQL Server es el encargado de distribuir eficazmente la memoria entre ambos procedimientos.

La mejor manera de determinar cómo es usada la memoria por el Servidor de SQL es ejecutar DBCC MEMUSAGE. Esta sentencia indica la cantidad de memoria asignada al Servidor de SQL en el comienzo, los 12 objetos más grandes en el cache de procedimientos, y los 20 objetos más grandes en el cache de datos. Por consiguiente, las recomendaciones son basadas en el uso de estos datos que determinaran el tamaño óptimo para los caches.

2.- Optimizando la Caché de datos

Este tipo de caché, está compuesto de la memoria sobrante del Servidor de SQL una vez que los requisitos del cache de procedimiento han sido satisfechos. Lo mejor es tener bastante espacio de cache para contener la mayoría los índices usados y un porcentaje respetable de las tablas mas accedidas frecuentemente, reduciendo las entradas/salidas físicas.

3.- Optimizando la Caché de Procedimientos

Al poner a punto el cache de procedimiento, lo mejor es determinar el tamaño óptimo para contener los procedimientos almacenados más activos.

SQL atribuye, por defecto, el 30 % de la memoria disponible a este tipo de caché, por lo que debemos de determinar si este porcentaje es nuestro valor adecuado.

Podemos determinar si la cache de procedimientos es bastante grande ejecutando los procedimientos almacenados más frecuentemente usados y ejecutando la sentencia DBCC MEMUSAGE; esto mostrara los 12 procedimientos almacenados más grandes en el cache de procedimientos.

4.- TEMPDB en RAM

Las consultas ejecutadas contra el Servidor de SQL que usan espacio de trabajo temporal como las operaciones, GROUP BY, tablas temporales, uniones de tablas (JOIN), son beneficiadas si la tempdb se encuentra en RAM. Para poner la tempdb en RAM, debemos de tener más memoria disponible que la que el servidor trae por defecto.

Esto que hemos visto, son ideas para mejorar el rendimiento de nuestro servidor de bases de datos, pero no son las únicas cosas que podemos hacer. Lo mejor para lograr un buen rendimiento, es mirar la documentación oficial.