Recomendaciones para la afinación de bases de datos

Mtra. Susana Laura Corona Correa

Dirección General de Cómputo y de Tecnologías de Información y Comunicación, UNAM

susanacc@unam.mx

Introducción

Las entidades y dependencias universitarias para el cumplimiento de sus objetivos y funciones designadas, se apoyan de sistemas informáticos que utilizan varios recursos para poder operar, entre los cuales se encuentran las bases de datos usadas para almacenar la información generada continuamente en la universidad, los servidores de aplicaciones, por mencionar algunos.

La cantidad de datos generados en la universidad apoyados de las tecnologías de información han contribuido al aumento de información contenida en las bases de datos. Junto con el aumento de información, la cantidad de recursos necesarios para poder explotarla también crecen.

Debido a la importancia de hacer un mejor uso de los recursos disponibles, así como de incrementar el desempeño de las bases de datos en ambientes de producción, tratando de que los usuarios obtengan la información deseada de forma oportuna, surge la necesidad de realizar actividades de optimización aprovechando las características que los manejadores de bases de datos tienen al respecto, entre las cuales se encuentra la afinación de bases de datos en ambientes de producción.

Consideraciones generales en la afinación de bases de datos

Una afinación de bases de datos consiste en un conjunto de actividades que hacen que una base de datos se desempeñe de manera más eficiente en términos de almacenamiento y rapidez de acceso, busca que reducir el tiempo de respuesta en las consultas hechas a la base de datos, haciendo un uso óptimo de los recursos existentes (Coronel, 2011:653).

Considerando lo anterior, algunos de los motivos para realizar una afinación de bases de datos son:

- aumentar la velocidad de respuesta de las consultas devolviendo la respuesta correcta en el menor tiempo posible.
- mejorar el uso de los recursos disponibles tales como: memoria RAM, CPU, disco duro, red, por mencionar algunos.

Para poder realizar un proceso de afinación, es necesario llevar a cabo un monitoreo y diagnóstico de la base de datos para determinar los aspectos que deban ser ajustados, tanto del lado del cliente en la optimización de las consultas, como del lado del servidor para mejorar el uso de recursos.

Antes de realizar cualquier cambio en la configuración o la estructura de la base de datos, es necesario hacer un respaldo tanto de los archivos de configuración como de las bases de datos con el fin de disminuir los riesgos y proteger los activos de información en caso de que un cambio ocasione un error o daño en la base de datos.

Recomendaciones generales

Diseño de la base de datos

El diseño adecuado de la base de datos puede ser un punto de partida para mejorar su desempeño, considerando los aspectos siguientes:

a) **llaves primarias.** Todas las tablas deberían tener una llave primaria asignada. Cuando se crea una llave primaria, los manejadores de bases de datos relacionales crean un índice de manera implícita, esto puede ayudar a disminuir el tiempo de respuesta de las consultas, además la llave primaria permite identificar de manera única un registro.
b) **Llaves foráneas.** Cuando se crean llaves foráneas, también se crean de manera implícita los índices, que pueden disminuir el tiempo de respuesta de las consultas, además de ayudar a mantener la integridad de la información.
c) **Índices.** Los campos que se utilizan en la cláusula WHERE o el operador JOIN que no son llaves primarias o foráneas, pueden necesitar que se genere un índice sobre ellos con la finalidad de disminuir el tiempo de respuesta de las consultas.
d) **Tipos de datos.** Verificar que los tipos de datos usados sean adecuados a los datos que contienen.

Revisión del plan de ejecución de consulta

Los manejadores de bases de datos usan los planes de consulta que contienen los pasos para acceder a los datos y devolver un resultado, estos planes de consulta se pueden revisar para identificar puntos de mejora en la consulta o la estructura de los objetos. Los comandos para obtener los planes de ejecución de las consultas pueden variar de acuerdo con el Sistema Manejador de Bases de Datos y su versión, por ejemplo:

Sistema manejador de bases de datos	Comando para revisar el plan de una consulta
PostgreSQL	EXPLAIN
MySQL	EXPLAIN
Oracle	EXPLAIN PLAN
Microsoft SQL Server	SET SHOWPLAN_ALL { ON | OFF }

Figura 1. Comandos para revisar los planes de consultas

Monitoreo

Revisar los archivos de bitácora o logs de la base de datos para detectar incidentes que deban ser atendidos incluso si no han afectado la operación de la base de datos.

Modificación de parámetros

La modificación de parámetros de configuración del sistema manejador de bases de datos se realiza mediante la edición de los archivos correspondientes en algunos manejadores como postgreSQL o MySQL o mediante las herramientas que proporcione el manejador para ese fin como en el caso de Oracle.

Sistema manejador de bases de datos	Archivos de configuración
PostgreSQL	postgresql.conf pg_hba.conf pg_ident.conf

Sistema manejador de bases de datos	Archivos de configuración
MySQL	my.cnf my.ini
Oracle	Server Parameter Files. Archivos de parámetros del servidor, son archivos binarios. Initialization parameter File. Archivos de texto para la inicialización de parámetros: init.ora, initdw.ora

Figura 2. Archivos de configuración de algunos DBMS

Para mantener un control de la modificación de parámetros se puede realizar usando un documento de bitácora en el que se haga el registro de la modificación o empleando un control de versiones de los archivos de configuración previos al cambio, en caso de que sea necesario volver a mantener los valores como se encontraban previamente, por ejemplo:

Parámetro	Valor anterior	Valor nuevo	Fecha del cambio	Motivo

Figura 3. Bitácora de cambios en los parámetros en la base de datos.

Configuración de parámetros de PostgreSQL

La configuración de parámetros de PostgreSQL se realiza en el archivo postgresql.conf, algunos de los parámetros de postgreSQL.

El archivo de configuración agrupa los parámetros en diferentes secciones:

- Ubicación de los archivos de la instancia instalada de postgreSQL
- Conexiones y autenticación
- Uso de recursos
- Logs
- Replicación
- Afinación de consultas
- Reportes y bitácoras
- entre otros.

Después de modificar los valores de uno o varios parámetros en el archivo es necesario ejecutar el comando pg_ctl. El comando pg_ctl cuenta con varios modos start, restart, stop y reload, los cuales permiten iniciar, reiniciar o detener el servicio de postgres o recargar los parámetros.

Para que el servicio de postgres aplique los cambios de los parámetros que se encuentran en el archivo de configuración con la leyenda # (change requires restart), es necesario reiniciar el servicio, en este caso se ejecutaría el comando pg_ctl usando el modo restart o con la ejecución del comando pc_ctl usando el modo stop seguido de la ejecución del mismo comando usando el modo start, para detener e iniciar el servicio.

Para confirmar los cambios de los parámetros que no requieren que se reinicie el servicio después de modificar el archivo es necesario ejecutar el comando pg_ctl reload el cual envía un SIGHUP signal al proceso de postgres ocasionando que se lean nuevamente los archivos de configuración.

Algunos de los parámetros que se pueden considerar en un proceso de afinación de la base de datos son:

shared_buffers

Establece la cantidad de memoria que el servidor de bases de datos utiliza para los buffers de memoria compartida, esta memoria es usada para almacenar datos en caché, esto permite que el acceso sea más rápido y en consecuencia el tiempo para ejecución de las consultas puede disminuir. Si el servidor cuenta con más de 1GB y es un servidor dedicado se podría configurar al 25% de memoria disponible del sistema. El incremento de este valor cuando no es requerido por las operaciones de la base de datos no va a significar una mejora en el desempeño de la base de datos, por lo que se podría establecer un valor inicial y monitorear los resultados obtenidos hasta encontrar el valor óptimo a las necesidades del servidor de bases de datos.

work_mem

Este parámetro especifica la cantidad de memoria que es utilizada para las operaciones de ordenamiento, tales como ORDER BY, DISTINCT, JOIN y hash de las tablas antes de escribir los datos en los archivos de disco temporales.

Para consultas que involucran ordenamientos grandes o para realizar la operación JOIN puede ser útil incrementar la cantidad de memoria de trabajo que puede ser usada por una consulta (Riggs, 2017:377).

maintenace_work_mem

Este parámetro nos permite establecer la cantidad máxima de memoria usada para las actividades de mantenimiento como las creaciones de índices con CREATE INDEX, agregar restricciones de llave foránea a una tabla con ALTER TABLE ADD FOREIGN KEY y el proceso de VACUUM. Incrementar el valor de este parámetro puede beneficiar en el desempeño de estas tareas, así como en la creación de respaldos de la base de datos.

effective_cache_size

Este parámetro sirve para establecer el tamaño efectivo del caché de disco que el planeador de consultas asume que está disponible para una sola consulta. Este valor es considerado en las estimaciones del costo de usar un índice, aumentar el valor del parámetro hace más probable que se utilice un INDEX SCAN en lugar de escaneo secuencial, lo que puede contribuir a que el tiempo de respuesta de las consultas sea menor.

autovacuum = on

Habilitar el subproceso de autovacuum, el cual reclama el almacenamiento ocupado por tuplas muertas de forma automática, es decir, aquellas que fueron eliminadas o que se volvieron obsoletas por un update, las cuales son eliminadas físicamente hasta que se ejecuta un vacuum en la tabla, estas actividades de limpieza ayudan a mejorar el rendimiento de la base de datos. El funcionamiento de este parámetro necesita que también se habilite el parámetro:

track_counts = on

autovacuum_max_workers

Establece el número máximo de subprocesos que utiliza el subproceso de autovacuum, si habilitamos el subproceso de autovacuum puede ser útil ajustar los valores de este parámetro principalmente cuando se ejecutan comando de INSERT, UPDATE y DELETE frecuentemente en las tablas y se observa que la ejecución del proceso de autovacuum tarda más del tiempo deseado.

Referencias

Coronel, Carlos. 2011. Bases de datos. Diseño, implementación y administración. México, Cengage Learning Editores, 690 pp.

Oracle. 2021. Oracle Database Release 21. Consultado el 07/12/2021 en https://docs.oracle.com/en/database/oracle/oracle-database/21/refrn/parameter-files.html#GUID-4590634E-85B1-4BA8-8293-FE9960D4E2C2

PostgreSQL. 2021. 20.4. Resource Consumption. Consultado el 07/12/2021. https://www.postgresql.org/docs/14/runtime-config-resource.html

PostgreSQL. 2021. 20.7. Query Planning. Consultado el 07/12/2021. https://www.postgresql.org/docs/14/runtime-config-query.html#RUNTIME-CONFIG-QUERY-GEQO

Riggs, Simon. 2017. PostgreSQL Administration Cookbook, Reino Unido, Pack Publishing, 517 pp.

CRÉDITOS

DGTIC-UNAM

Elaborado por: Mtra. Susana Laura Corona Correa

Revisión técnica: Mtro. Hugo Alonso Reyes Herrera

Revisión de estilo: Pamela Valdés Reséndiz

Autorización de publicación: Dra. Marcela J. Peñaloza Báez

Dirección de Colaboración y Vinculación, 2021