Examen final

Tecnológico de Costa Rica
Escuela de Ingeniería en Computación
Base de datos II (IC 4302)
Segundo Semestre 2022
Estudiante: Pamela González López - 2019390545

PREGUNTA 1 (60 pts)

a) Dar una solución detallada de cómo podría mejorar el rendimiento de la base de datos actual, reduciendo el downtime al mínimo, esto permitirá ganar tiempo para dar una solución mucho más duradera con la mínima afectación a los usuarios. (10 pts)

Desde mi punto de vista, el principal problema en este momento sería obviamente la tabla post. En este caso en concreto conforme la tabla crece y crece y hay más inserción de datos, una consulta que pudo haber sido rápida al inicio ahora resulta más lenta. Para atacar el problema primero tendría que analizar la forma en que se están realizando las consultas y tratar de agregar algún índice como en el id del usuario, si es que aún no lo tiene; analizar si el tipo de consulta está utilizando correctamente los índices o los está desaprovechando. Analizaría la forma en que se están realizando las consultas para mejorar el rendimiento.
Como segunda y última medida, trataría de realizar un particionamiento a la tabla post específicamente teniendo como base el TIMESTAMP, en este caso lo haría por año y mes o hasta se podría reducir más.

b) Dar una recomendación detallada de qué tipo de base de datos se debería utilizar para abordar este problema, además debe recomendar algunas de las bases de datos SQL o NoSQL estudiadas durante el curso tanto en lecturas, así como las utilizadas en proyectos o ejemplos en clase. Tome en cuenta que sería posible utilizar más de una base de datos para optimizar el almacenamiento de los datos de la tabla post, amigos y usuario, tome en cuenta que tan fácil es escalar la base de datos en su recomendación, debe dar prioridad a servicios managed services y SaaS, no olvide la localidad y naturaleza de los datos. (30 pts)

Desde mi punto de vista, se podría seguir utilizando MariaDB para las tablas de usuario y amigos, si bien ambas tablas juegan un papel importante en el rendimiento ambas se pueden manejar bien en esta base de datos por su naturaleza de no tener un crecimiento abismal en el transcurso del tiempo.

Se podría tratar de normalizar la tabla usuario, por ejemplo:

- Separar la parte de provincia, estado y zip en una tabla diferente y referenciar esa localidad con una llave foránea.
- Y otras dos tablas de país y estado. En donde se indique el idPais y el nombre del país, y otra tabla en donde se indique idEstado y nombre del estado. Para así hacer una tabla de localidad que referencia el id del país y el id del estado.
- Separar los número de teléfono. Se podría poner una tabla 'Telefonos x usuario' en donde se indique el id del usuario y el teléfono.
- Separar los dos apellidos.

Tabla usuario

idPersona	PrimerApellido	SegundoApellido	idLocalidad

Teléfono X Usuario

idPersona	NTelefono

Localidad

idLocalidad	idPais	idEstado

El mayor problema de este caso radica, como se indicó anteriormente, en la tabla post por su naturaleza; en donde un mismo usuario puede crear muchísimos post en cuestión de horas o hasta minutos, en este caso acudiría más a base de datos

NoSQL (Elasticsearch o Bigtable) por su escalabilidad, en donde se puede tener un conjunto de servidores que trabajan como un todo y poder de este modo dividir la demanda de trabajo (consultas, inserciones, etc) causado por los post y afrontar las cargas que se vayan a crear entre más crezca o sea utilizada la red social.

c) Comente acerca de qué tan conveniente es mantener la base de datos actual en la casa de uno de los fundadores, comparado con mover ésta algún Cloud Provider como AWS. (10 pts)

No es nada conveniente mantener la base de datos en la casa de uno de los fundadores, los motivos: El crecimiento que tendrán los datos y el actual rendimiento deficiente que está teniendo la base de datos. Si se emigra a algún Cloud Provider como AWS se vería beneficiado con el tema de la flexibilidad, pues depende de la demanda que se esté presentando en la base de datos así se podría manejar los recursos aumentándolos o disminuyendolos, el tema la disponibilidad de la base y la seguridad y recuperación de los datos, pues en este tipo de Cloud la seguridad y recuperación de imprevistos está muy bien desarrollado e implementado.

d) Basándose en el funcionamiento de un índice invertido el cual fue estudiado en clase y es utilizado por motores como Elasticsearch y el concepto de Natural Languaje Processing (NLP) llamado Stemming el cual también fue discutido en clase, comente ¿Cómo se podría reducir el memory footprint de la base de datos actual? (10 pts)

El índice invertido se centra en dos temas: el diccionario de términos y la cantidad de apariciones de dicho término, si utilizáramos esto en la base actual podríamos hacer una índice invertido con las palabras más repetidas que se publican en la red social, teniendo en cuenta que se hagan búsquedas sobre estas palabras, y guardando así su localización, podría crear una nueva tabla que relacione la tabla post con la “nueva tabla de términos”. Para reducir el memory footprint habría que juntar esta idea de crear un índice invertido junto al stemming que utiliza la raíz de las palabras en lugar de las palabras conjugadas. Para hacer índices mucho más reducidos y que el rendimiento sea mayor.

PREGUNTA 2 (10 pts)

- Comente, ¿Cómo afectan los índices en el rendimiento de las bases de datos relacionales?, enfoque su respuesta tanto en cómo benefician el rendimiento así como la forma en la cual lo impactan de forma negativa. Suponiendo que el hardware no es un problema (se puede comprar cuanto se necesite), ¿Podemos crear cuantos índices queramos o estos no tendrán mayor impacto en el rendimiento?

Beneficio en el rendimiento: Lo que buscan los índices es mejorar las lecturas y escrituras, más rapidez y en consecuencia un mejor rendimiento. Por ejemplo, tener un clustered index lo que haría es crear un único índice dentro del archivo y su objetivo es que dure menos la búsqueda de los datos. Lo que buscan es tratar de no entrar tanto a disco para no consumir tiempo de ejecución.

Impacto de forma negativa: Se pensaría que crear índice, uno tras otro se traduce en mayor rendimiento, pero esto no es cierto. El motivo es que para hacer un índice se debe conocer bien el tipo de datos que se van a necesitar, el workload, etc. Por ejemplo (visto en clase), un caso que estaría mal utilizado un índice.
Se realizáramos:

```
SELECT nombre, edad, fechaN FROM Persona
WHERE edad >= 30 AND provincia = ‘x’
ORDER BY provincia
```

Este tipo de consulta afectaría de manera negativa el rendimiento por hacer un order by con columnas fuera de la tabla, pues se debe hacer un ordenamiento para analizar ambas.
Para finalizar, teniendo el supuesto que el dinero no es problema, considero igualmente que crear cuantos índices queramos es contraproducente y habría un impacto negativo en el rendimiento pues el consumo de recursos crecerá demasiado pero crecerá tontamente pues si se crea un índice correctamente no habría necesidad de consumir recursos innecesariamente además.

PREGUNTA 3 (20 pts)

- El rendimiento de todo sistema de base de datos puede verse afectado por muchos factores, uno de ellos es el ambiente en el cual se ejecuta, este se encuentra compuesto por los componentes de hardware y el sistema operativo y otros programas de usuario compitiendo por los recursos del computador. Comente de forma clara y concisa, ¿Cómo afecta el rendimiento de una base de datos los componentes ilustrados en la Figura 1?

El rendimiento de una BD se podría ver afectado si no utilizamos un disco correcto y que supla las necesidades, por ejemplo, si tuviéramos un crecimiento en el volumen de los datos de una BD que se está dando de manera constante y en gran medida. Deberíamos tener presente el tipo de disco que estamos utilizando. También si tenemos en cuenta que cuando el memory footprint de la BD alcanza un nivel muy alto de uso de memoria la máquina se ve obligada a la paginación del disco pues ésta serviría para almacenar los procesos que están en memoria actualmente y que superan el máximo de memoria que tiene el sistema
.
Si la memoria es insuficiente en nuestra máquina, el rendimiento de la base cae considerablemente, pues el tiempo de espera por cada transacción aumenta. Igualmente si el caché es pequeño se recurriría a la memoria principal, lo cual podría ralentizar la BD.

Para el sistema operativo, si la base de datos estuviera en un sistema monolítico se vería afectada de manera negativa pues estos son sistemas que trabajan con procedimientos entrelazados entre sí. Lo mejor sería utilizar aquellos OS que trabajen por capas.

El cpu afecta la BD por el tema del paralelismo, si se tiene pocos cores los trabajos que se pueden hacer en paralelo se reduce. En consecuencia se reduce el rendimiento. Lo ideal es tener un equilibrio entre cpu y memoria.

Tener muchos programas de usuario corriendo al mismo tiempo, reduce los recursos de la base de datos. Lo ideal es tratar de mantener los programas de usuarios controlados para que no afecte el rendimiento de la BD.

La red delimita la cantidad de usuario que se puede manejar en una base de datos, cómo se podrán sincronizar las bases de datos, etc. En este caso, una red deficiente afecta directamente en el rendimiento de la BD.

PREGUNTA 4 (10 pts)

- La escalabilidad automática es una característica muy deseada en los sistemas de bases de datos tanto SQL como NoSQL, la misma permite mediante la obtención de métricas en tiempo real interpretar el comportamiento actual para predecir el comportamiento futuro, con esto se puede ajustar tanto el hardware como la configuración de las bases de datos, para poder atender el workload de un sistema. Comente la importancia de la Observabilidad tanto a nivel de aplicación como de base de datos para lograr una escalabilidad automática adecuada, ¿Considera que las métricas de memoria, CPU y disco son suficientes para lograr esto?

La observabilidad es sumamente importante a nivel de aplicación como de base de datos porque de esta manera, al juntar todas las métricas suministradas mientras los sistemas se están ejecutando nos permite analizar cómo se está comportando los programas, si hay cuellos de botella, qué tan eficientes están siendo y si se podrían mejorar.

Considero que las métricas de memoria, CPU y disco no son suficientes para tener una escalabilidad automática, se debería obtener más información de otras métricas como: métricas para saber la concurrencia de usuarios, la cantidad de grupos o transacciones que se procesan en ciertos periodos de tiempo o en general, métricas para la cantidad de errores que causan un tipo de transacción en específico, tiempos de ejecución, etc. Se podría obtener muchas métricas para una escalabilidad más robusta.