

1. Suponiendo que un sistema de bases de datos relacional que presenta un read-heavy workload y los queries son muy diferentes, explique detalladamente ¿porque el uso de caches puede afectar el rendimiento del sistema de forma negativa?

Los caches funcionan al almacenar datos consultados con frecuencia en una memoria temporal. Facilitan el acceso a los datos y reducen las cargas de trabajo de las bases de datos. Por lo tanto, reduce en gran medida el tiempo para leer información cuando se necesita nuevamente. La memoria caché se puede configurar en diferentes niveles o por sí sola, según el caso de uso. [1] Al tener un workload read-heavy y queries diferentes entre sí, el uso de los caches es innecesario ya que los datos consultados en un query, muy probablemente van a ser diferentes a los que sea desea acceder en otra consulta. Por esto, afecta el rendimiento del sistema de forma negativa, ya que la primera consulta se le haría al cache. Como la consulta va a ser muy diferente a las consultas previas, no se van a encontrar los datos buscados. Al no encontrarlos, la consulta se redirige a la base de datos. El cache, en este caso, agrega un paso innecesario que solo aumenta el tiempo que toma la consulta.

2. El particionamiento de tablas en bases de datos relacionales es un concepto muy parecido al de shards en bases de datos NoSQL, explique detalladamente ¿Cómo afecta el particionamiento y el sharding en el rendimiento de bases de datos SQL y NoSQL?

El **particionamiento de tablas en bases de datos relacionales** es el proceso de base de datos donde las tablas muy grandes se dividen en varias partes más pequeñas. Esto permite que las consultas que acceden solo a una fracción de los datos pueden ejecutarse más rápido porque hay menos datos que analizar. El objetivo principal de la partición es ayudar en el mantenimiento de tablas grandes y reducir el tiempo de respuesta general para leer y cargar datos para operaciones SQL particulares.

Por otro lado, el **Sharding** es replicar el esquema y dividir los datos en función de una clave de partición en una instancia de servidor de base de datos independiente, para distribuir la carga. Esto conlleva a varios beneficios:

- Tamaño de índice reducido: dado que las tablas se dividen y distribuyen en varios servidores, se reduce el número total de filas de cada tabla de cada base de datos. Esto reduce el tamaño del índice, lo que generalmente mejora el rendimiento de la búsqueda.
- Distribuir base de datos en varias máquinas: una partición de base de datos se puede colocar en hardware independiente y varias particiones se pueden colocar en varias máquinas. Esto permite una distribución de la base de datos en un gran número de máquinas, mejorando en gran medida el rendimiento - Segmentar datos por geografía: si la partición de la base de datos se basa en alguna segmentación del mundo real de los datos entonces es posible inferir la pertenencia de partición adecuada de forma fácil y automática, y consultar solo la partición relevante. [2]

3. En un sistema de bases de datos con Strong Consistency cuyo workload es de read-heavy y write-heavy, ¿Cómo afectan los exclusive locks el rendimiento de las bases de datos NoSQL?

Una base de datos con Strong Consistency implica que los datos deben ser constantemente consistentes en todo momento. Un workload de read-heavy y write-heavy lleva a cambios constantes en la base de datos, por lo que esta debe estar constantemente replicando la información a sus nodos para que todos contengan la misma información. Los exclusive locks afectan el rendimiento de manera negativa, ya que cada vez que se hace una operación de write por ejemplo, se debe bloquear el recurso compartido de manera que no sea necesario hacer un rollback en cascada en todas las réplicas. Al ser el sistema write-heavy, este está constantemente realizando bloqueos exclusivos, que ya por si solos elevan el overhead y aumenta el tiempo de espera. Este efecto se ve elevado por la cantidad de operaciones que se realicen en el sistema.

#### 4. Explique detalladamente, ¿Cómo afecta la selección de discos físicos el rendimiento de una base de datos SQL y NoSQL?

Cada vez que se necesita leer un fragmento de datos de SQL Server se debe recuperar información del disco. Esta recuperación provoca una E/S de disco. Es por esto, que la selección de discos físicos afecta el rendimientos de una base de datos, ya sea SQL o NoSQL. Los discos físicos pueden ser un disco duro de cabeza lectora (HDD por sus siglas en inglés) o un disco de estado sólido (SSD). Un HDD es un dispositivo de almacenamiento de datos que vive dentro de la computadora. Tiene discos giratorios en el interior donde los datos se almacenan magnéticamente y un cabeza que lee y escribe datos en el disco. Por otro lado, en un SSD, todos los datos se almacenan en circuitos integrados. Su capacidad, o la cantidad de datos que pueden contener, varía, lo que los hace flexibles para dispositivos más pequeños, como computadoras portátiles delgadas, convertibles o 2 en 1. Reducen drásticamente el tiempo de acceso, ya que los usuarios no tienen que esperar a que se inicie la rotación de disco. [3]

En sí, seleccionar uno de estos discos influye fuertemente en el rendimiento, particularmente en los siguientes aspectos:

- Velocidad: Los SSD superan a los HDD porque utilizan circuitos eléctricos y no tienen partes móviles físicas. Esto conduce a tiempos de espera más cortos cuando se está iniciando y menos retrasos al abrir aplicaciones o realizar tareas informáticas pesadas.
- Fiabilidad: Los SSD están hechos para manejar cargas de trabajo diarias con lectura y escritura frecuentes, en lugar de estar llenos de archivos que permanecen intactos durante años.
- Durabilidad: Con el uso regular, los SSD suelen durar unos diez años, una vida útil que rivaliza con su unidad de disco duro típica. Este no fue siempre el caso, porque las celdas en una unidad de estado sólido se pueden escribir y leer solo un número finito de veces. Por el contrario, los HDD tienen una capacidad de lectura/escritura teóricamente infinita, suponiendo que la mecánica interna siga funcionando correctamente. [4]

#### Referencias

1. Bluzelle. (2019, March 15). Things You Should Know About Database Caching. Medium.  
<https://blog.bluzelle.com/things-you-should-know-about-database-caching-2e8451656c2d>
2. Database Sharding vs. Partitioning: What's the Difference? (2021, October 29).  
<https://www.singlestore.com/blog/database-sharding-vs-partitioning-whats-the-difference/>
3. Intel. (2021, April 6). SSD vs. HDD: Which Is Best for You?  
<https://www.intel.com/content/www/us/en/products/docs/memory-storage/solid-state-drives/ssd-vs-hdd.html>
4. Freda, A. (2022, November 2). SSD or HDD: Which Is Right for You? AVG.  
<https://www.avg.com/en/signal/ssd-hdd-which-is-best>