

- 1) En un sistema de bases de datos relacional con un workload orientado a la lectura y queries muy diferentes, el uso de caches puede afectar negativamente el rendimiento. Esto se debe a que el tamaño limitado de la cache puede provocar una baja tasa de aciertos, las frecuentes invalidaciones de cache generan sobrecarga adicional, la gestión de la cache se vuelve más compleja y se experimenta una degradación del rendimiento debido a la falta de localidad de los datos. Es importante considerar cuidadosamente el diseño y la configuración de la cache en estos casos para minimizar los impactos negativos en el rendimiento del sistema.
- 2) En el caso del particionamiento en SQL, se divide una tabla en fragmentos más pequeños, esto permite un procesamiento paralelo de consultas y una mejor utilización de los recursos. Además, el particionamiento reduce la congestión al distribuir las consultas entre diferentes nodos, mejorando la tolerancia a fallos y facilitando el acceso eficiente a datos específicos mediante la segmentación de datos basada en criterios como rangos de valores.
En el contexto del sharding en bases de datos NoSQL, se distribuyen los datos en shards o fragmentos de datos que se almacenan en diferentes servidores o nodos. Esto proporciona paralelismo y escalabilidad al procesar diferentes shards en paralelo y agregar más nodos según sea necesario. El sharding también distribuye la carga de escritura y lectura entre los shards, evitando la congestión en un solo nodo y mejorando el rendimiento global del sistema. Además, el sharding permite una reducción de la latencia al acercar los datos a los usuarios o aplicaciones a través de la ubicación estratégica de los shards, y brinda flexibilidad en el esquema al permitir que los shards contengan diferentes tipos de datos, adaptándose a las necesidades y consultas específicas.
- 3) En un sistema de bases de datos NoSQL con Strong Consistency, los exclusive locks pueden afectar el rendimiento de diferentes maneras. Los locks exclusivos en operaciones de escritura pueden causar cuellos de botella y retrasos en un workload write-heavy, ya que las operaciones deben esperar su turno para acceder a los datos. Del mismo modo, los locks exclusivos en operaciones de lectura pueden generar bloqueos y disminuir el rendimiento en un workload read-heavy. Además, el uso de exclusive locks puede generar contención y limitar la escalabilidad del sistema, especialmente en situaciones de alta concurrencia.
- 4) La elección de los discos físicos tiene un impacto significativo en el rendimiento de bases de datos SQL y NoSQL. En el caso de las bases de datos SQL, la elección del tipo de disco, como los discos de estado sólido (SSD) en lugar de los discos magnéticos tradicionales (HDD), puede mejorar la velocidad de acceso a los datos y reducir la latencia. Además, la configuración RAID y la distribución física de los archivos también influyen en el rendimiento, ya que una

configuración RAID adecuada y la colocación de datos en discos separados pueden evitar cuellos de botella y mejorar la eficiencia.

En las bases de datos NoSQL, la selección de discos físicos se relaciona con la arquitectura de almacenamiento, la distribución de datos y la configuración de particionamiento. Dependiendo de la base de datos NoSQL utilizada, es posible que se prefiera el almacenamiento en memoria (RAM) para un rendimiento ultrarrápido o SSDs para un acceso rápido a los datos. Además, la distribución adecuada de los discos físicos en relación con la distribución y replicación de datos puede reducir la latencia y mejorar el rendimiento. El particionamiento también puede optimizarse al seleccionar los discos físicos correctos para evitar cuellos de botella en discos específicos.