

# Quiz #9

---

Fecha: 8/11/2022

Autor: Deyan Sanabria Fallas #2021046131

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?

La cache o los caches suelen hacer referencia a un espacio de memoria relativamente rapido que sirve para almacenar datos que suelen necesitarse con frecuencia, por lo que, mediante un algoritmo, se intenta ver que datos son utiles tener en la cache para no tener que ir a un storage mas lento a recuperar los datos. Suena algo beneficioso, pero en el contexto de queries muy diferentes y de muchas lecturas ocurre lo siguiente, si estas queries son muy diferentes y suelen hacerse llamadas con datos muy aleatorios, ¿de que sirve tener muchos caches? los caches no suelen almacenar demasiados datos, entonces, si se realizan llamadas a datos de forma desordenada y aleatoria los caches no podran cumplir con su objetivo y el andar guardando datos que se piensan que seran utiles en las caches solo hara que al final el rendimiento se vea perjudicado mas que beneficiado.

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 o sharding es en general la idea de separar los datos y, por asi decirlo, generar grupos, esto sirve bien con replicas ya que cada replica va a administrar un "grupo" al que en bases NoSQL se le llaman Shards, entonces tendriamos multiples sistemas cada uno con grupos de datos. En temas de rendimiento pues logramos hasta cierto punto separar el workload en diversas maquinas, donde cada una va a tener una parte de los datos y cuando se necesiten datos de dicha parte, se le solicita a la replica correspondiente.

Ahora necesitar de todos los datos puede ser un tanto distinto. Por ejemplo, una Consulta federada, donde todos los datos estan distribuidas en grupos/shards entre varios servidores, pues aqui el rendimiento es afectado porque el tiempo que se tarde va ser igual al servidor mas lento, puesto a que se necesita del resultado de su query para poder unir las todas.

En general las ventajas y desventajas de rendimiento van a depender de como esten distribuidos los datos y los tipos de consulta que se realicen.

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?

Los exclusive locks ocurre cuando una transaccion requiere de un recurso, entonces se bloquea el acceso al recurso hasta que termine la transaccion, esto suele aplicarse a escrituras puesto que varias transacciones leyendo una misma cosa no afecta su estado, en cambio una escritura no puede permitir que otros vayan a leer si el estado del recurso va a cambiar, ni escribir porque estaria sobrescribiendo lo de la otra transaccion. En un workload donde hay muchas escrituras y muchas lecturas en una base de datos NoSQL es algo bastante perjudicial, puesto a que apenas se coloquen locks en los multiples recursos, el monton de escrituras y lecturas que vengan detras van a tener que esperar, ahora en el caso que sean multiples escrituras a un mismo recurso y despues multiples lecturas, las lecturas tendrian un tiempo de respuesta muy grande. Para una base de datos NoSQL no es muy habitual la consistencia fuerte.

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

En general, la seleccion de los discos de una base de datos puede afectar el rendimiento en varios aspectos, por ejemplo, elegir de disco un disco duro convencional que tiene discos girando y una cabeza lectora va ser mucho más lento que con un disco de estado solido, el cual no tiene piezas mecanicas y son chips muy rapidos. En el caso de los discos duros, los datos se deben almacenar de forma secuencial debido a que, sino, la cabeza lectora tiene que saltar mucho y es muy lento, el proceso de desfragmentacion que se suele hacer en estos discos, puede llegar a ser un proceso tardado pero necesario para mantener la velocidad del sistema. En cambio un disco de estado solido no tiene estas desventajas y puede ser accedido de forma aleatoria sin tanta penalizacion como en un disco duro. Discos duros suelen tener mas espacio de almacenamiento por lo que seria ideal usarlos en data tiers "frios" mientras que los SSDs en data tiers "calientes" por que necesita de su velocidad mantener alto el rendimiento sobre los datos mas frecuentados.

Los discos fisicos a comparacion de discos de red suelen ser mucho mas rapidos, al estos estar conectados directamente a la maquina no depende la velocidad de la red. Al elegir un disco para una base de datos, es importante tener en cuenta los IOPS (operaciones de entrada y salia) que nos van a definir la velocidad del disco, y tambien el tamaño del disco que nos va a decir cuanto podemos almacenar.