UNDEF

CENTRO REGIONAL
UNIVERSITARIO
CÓRDOBA IUA

REDIS

Base de Datos I - INGENIERÍA INFORMÁTICA

Profesor:
Ing. Mariano, García Mattío.

Integrantes:
Garibaldi, Bruno (bgaribaldi709@alumnos.iua.edu.ar).
Macedo, Bautista (bmacedo197@alumnos.iua.edu.ar).
Bertorello, Tiziano (mbertorello972@alumnos.iua.edu.ar).
Cernik, Joaquín (jcernik294@alumnos.iua.edu.ar)).

2023

ÍNDICE:

Introducción

Antes que nada nos parece primordial definir que es una base de datos: Una base de datos es una colección de información o datos relacionados entre sí, los cuales se almacenan y organizan en un sistema de gestión de base de datos o en un programa que permite acceder a ellos y manipularlos. Las funciones que se pueden realizar con los datos se basan en la adición, eliminación, edición y consulta o lectura.

Podemos diferenciar básicamente en dos grandes grupos a las bases de datos. Por un lado, tenemos las bases de datos relacionales, las cuales organizan los datos en tablas que se relacionan entre sí a través de claves únicas o primarias para cada tabla. Es un modelo altamente estructurado y utiliza SQL (lenguaje de consulta estructurado) para manipular o consultar los datos. Por otro lado, tenemos las bases de datos no relacionales o NoSQL, que se basan en un modelo de datos no estructurado que utiliza objetos y la estructura de datos varía según el tipo de datos que estemos utilizando. Utiliza diferentes lenguajes para la consulta y manipulación de datos. Estas bases de datos son perfectas para manipular información de redes sociales, contenido multimedia, datos de sensores, etc.

En este contexto Redis es una base de datos NoSQL, que almacena estructuras de datos en memoria, es decir que se ejecuta completamente en la memoria ram del servidor y no utiliza memoria en disco para almacenar datos de forma permanente. Redis se caracteriza por utilizar el método de clave-valor o key-value para almacenar datos, donde se guardan como pares únicos. Dentro de las aplicaciones más usuales, al guardar los datos en memoria hace que tenga una gran velocidad y performance, lo que permite su uso en aplicaciones de tiempo real que requieran de baja latencia y alto rendimiento. También es muy utilizado como sistema de caché de datos para optimizar al máximo el acceso a datos y la performance general de las aplicaciones.

Entonces podemos definir como objetivos principales de Redis, reducir los tiempos de latencia de acceso a datos, implementar una memoria caché de alta disponibilidad y aunque no es su punto principal, mantener los datos persistentes.

Historia

A continuación, brevemente introduciremos por puntos lo que fue la historia de Redis.

→ Fue fundado en 2009 por Salvatore Sanfilippo (Italiano nacido en 1975, conocido por su desarrollo en bases de datos en memoria y en tecnologías de bases de datos en general), se fundó enfocado en la velocidad de acceso y simplicidad.

→ En 2010 se publicó la versión v1.0.0, centrada en la capacidad de almacenar y recuperar datos de manera eficiente en estructuras de datos simples, luego se centró en el soporte para estructuras más complejas. La empresa VMWare apoya el proyecto contratando a Salvatore y otro desarrollador para que se dediquen a tiempo completo a Redis.

→ En 2012 se publicó la v2.6 que introdujo la replicación maestro-esclavo.

→ En 2015 se publicó la v3.0 que mejoró el rendimiento y permite nuevos tipos de estructuras de datos.

→ En 2016 llegó la v4.0 con mejoras de escalabilidad y rendimiento.

→ Al año siguiente llegó la v5.0 con soporte para modelos (permite que los desarrolladores agreguen funcionalidades personalizadas a Redis).

→ En 2020 con la v6.0, aparecen nuevas medidas de seguridad, soporte para protocolo RESP3.

→ Twitter, Instagram y GitHub, fueron los primeros en utilizar este motor, dándole mayor reconocimiento mundialmente y mostrando la amplia capacidad de escalabilidad que genera en las empresas.

→ Actualmente, Redis es una de las bases de datos más utilizadas, más específicamente por aquellas aquellas empresas/proyectos que buscan una alta velocidad de escritura y lectura de datos. Además, a lo largo del tiempo ha tenido notables mejoras respecto a su soporte, integración con otros sistemas y herramientas tales como AWS y Kubernetes. Para concluir esta sección, Redis es

una base de datos de alto rendimiento la cual sigue mejorando a medida que pasa el tiempo, y si bien no es algo muy complejo, como veremos más adelante, es utilizado en muchos proyectos hoy en día.

Características

Velocidad

Debido al almacenamiento de datos en la memoria ram, permite una mayor velocidad de lectura y escritura de los datos en comparación de que si estuvieran en el disco. Gracias a ello, la base de datos puede generar más transacciones por segundo.

Simplicidad y facilidad de uso

Este sistema simplifica el código para que sea más fácil y ligero, permitiendo escribir código tradicionalmente complejo en menos líneas. Además esto permite una mayor compatibilidad con una gran cantidad de lenguajes de programación tales como Java, Python, PHP, C, C++, C#, JavaScript, Node.js, Ruby, R, Go, entre otros.

Persistencia

La persistencia se refiere a la escritura de datos en un almacenamiento duradero. Redis no fue diseñado para ser una base de datos duradera. Al almacenar los datos en memoria, estos son volátiles con lo cual ante cualquier falla o apagón, los datos se pierden. Sin embargo, Redis posee varios sistemas de persistencia:

→ RDB: Cada cierto intervalo de tiempo, Redis captura y almacena los datos.

→ AOF: Redis registra cada operación que se haya realizado en el servidor. Cuando se vuelve a reproducir el servidor, estas ejecutan nuevamente reconstruyendo la base de datos nuevamente.

→ Sin persistencia: En Redis es posible desactivar la persistencia, se utiliza cuando se almacena caché.

→ RDB + AOF: Pueden combinarse.

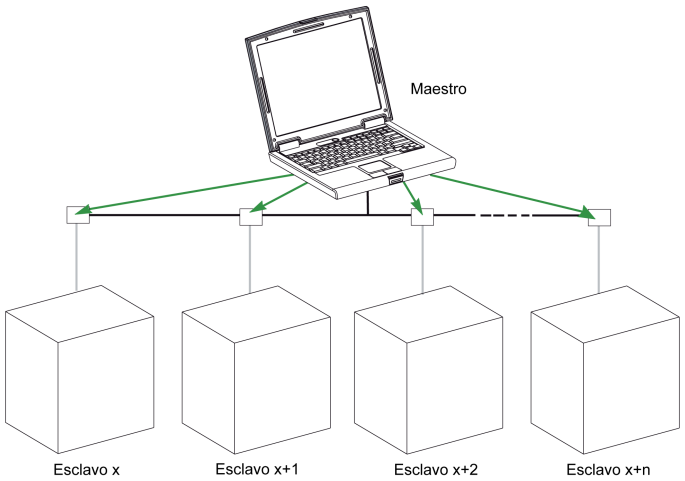
Replicación

El mecanismo de solicitud-respuesta denominado maestro/esclavo, se da entre un cliente (maestro) y un servidor (esclavo). En el cual un dispositivo (maestro) controla a uno o más dispositivos (esclavos), el cual le permite tener una copias exactas de los datos almacenados, y a su vez los esclavos pueden ser maestros de otros esclavos.

El principio maestro-esclavo presenta las siguientes características:

→ Un solo maestro puede estar conectado a la red en cada momento. Por lo tanto, solo él puede iniciar la comunicación y enviar solicitudes a los esclavos.

→ La arquitectura maestro-esclavo permite la replicación asíncrona mediante la que los datos se replican en numerosos servidores esclavos. De este modo, se logra mejorar el desempeño de lectura (ya que las lecturas se pueden repartir entre servidores) y de recuperación cuando el servidor principal sufre un fallo.



Los esclavos solamente pueden enviar respuestas al maestro, no pueden iniciar comunicación, ni con el maestro ni con otros esclavos.

SENTINEL: Son los procesos encargados de monitorizar las instancias de Redis (maestras y esclavos) y de iniciar el proceso de Failover (la capacidad de un sistema de seguir funcionando, aún en caso de producirse algún fallo en el sistema) en caso de que sea necesario promover una instancia esclavo a maestro.

•Expiración de Claves

Para una mejor optimización y para evitar un cierto “desperdicio” de memoria RAM, REDIS asigna un cierto tiempo de vida a las claves, permitiendo que en caso de que no estén siendo utilizadas o no estén siendo requeridas, estas claves o datos pueden ser eliminados. Esto puede realizarse tanto de manera manual como automática.

Código Abierto y Comunidad

Actualmente Redis es un sistema OpenSource. Es desarrollado y mantenido por Redis Labs, pero cuenta con el apoyo de una gran comunidad activa que contribuye al uso y desarrollo del software.[1]

Principales usos e implementaciones comerciales

Twitter lo usa para las estadísticas en tiempo real, Instagram para almacenar información de sesión de usuario, gestionar su cola de comentarios en tiempo real, GitHub para su sistema de cacheo de datos, Pinterest para gestionar su caché, Uber para gestionar su sistema de geolocalización y seguimiento de conductores, Twitch para gestionar su sistema de chat en tiempo real y mantener un alto rendimiento.

Los principales usos que le podemos dar son: Almacenamiento en caché (latencia reducida), chat, mensajerías, tablas de clasificación de videojuegos, almacen de sesiones, streaming de contenido multimedia, análisis geoespacial y colas (necesaria la rapidez para los servicios en tiempo real), actualmente se utiliza en Machine Learning, ya que requiere de procesamiento de datos rápidos en gran volumen, por lo que también se utiliza.

Tipos y estructuras de datos

En Redis hay diferentes estructuras y tipos de datos los cuales tienen sus propias características y formas de almacenamiento. A continuación se nombraran y explicarán los más importantes.

→ **Strings:**

[1] El código fuente, así como la documentación, discusiones y demás de Redis está alojado en su repositorio de Github. Link: https://github.com/redis

Almacenan cualquier tipo de datos tales como; cadenas (binarias o texto), números, etc. A su vez, estas aceptan operaciones de incremento y decremento (para números).

→ **Listas:**

Las listas en Redis son “series” de strings ordenadas las cuales se pueden agregar o sacar de distintas posiciones (nodos). A su vez, aceptan operaciones como búsquedas por índice.

→ **Sets:**

Colecciones de strings (no ordenadas) que admiten operaciones tales como operaciones de unión, diferencia e intersección (permiten hasta cuatro billones de elementos).

→ **Sorted Sets:**

Similar a los sets, pero con la gran diferencia de ser una estructura de datos ordenada. Estos tienen valores numéricos asociados y permiten operaciones de unión, intersección y diferencia tal como los sets. Permiten una búsqueda por rango de puntuación y se ordenan lexicográficamente.

→ **Hashes:**

Útiles para el guardado de objetos o estructuras de datos complejos, almacenados como colecciones de pares clave-valor. Son utilizados para las operaciones en los distintos campos del hash (HSET, HGET, HMSET, etc).

→ **Bitmaps:**

Útiles para la representación de datos binarios pero en forma de “imágenes”. Son matrices que posibilitan la eliminación y establecimiento de prueba de bits individuales.

Ventajas y desventajas de utilizar Redis:

Ventajas:

→ Buena velocidad debido a que los datos se almacenan principalmente en memoria.

→ Escalabilidad (capacidad para ampliar el sistema según las necesidad de aplicación).
→ Buen rendimiento para usos que requieran actualizaciones en tiempo real.
→ Compatible con una gran variedad de lenguajes.
→ Al ser opensource, permite una alta disponibilidad.
→ De la mano del punto anterior, no hay limitaciones de proveedores ni tecnología.

Desventajas:
→ Normalmente se almacena sólo en RAM, además de ser más costoso, si el sistema se apaga, se pierden los datos. Además no brinda seguridad para transacciones complejas. También limita la cantidad de operaciones que se realizan ya que la memoria RAM normalmente tiene menos capacidad que el disco de almacenamiento.

REDIS en funcionamiento y algunas sentencias importantes

Todas las sentencias y comandos de REDIS las podemos encontrar en su página web en el siguiente link: https://redis.io/commands/.

A continuación daremos algunos ejemplos de sentencias importantes REDIS:

Una vez instalado redis podemos inicializar nuestro servidor de redis ejecutando **redis-server**, si todo sale bien no veremos ningún error.

Luego para empezar a trabajar con REDIS se ejecutará **redis-cli,** esto nos permitirá ejecutar los siguientes comandos:

Ping- Si todo sale bien, recibiremos un **PONG**,con esto nos aseguramos que REDIS está funcionando correctamente.

- SET key value - Establece una clave con un valor específico.
- GET key - Obtiene el valor asociado a una clave.
- DEL key - Elimina una clave y su valor asociado.
- INCR key - Incrementa el valor asociado a una clave en uno.
- EXPIRE key time - Establece un tiempo de expiración (en segundos) para una clave.

- KEYS pattern - Encuentra todas las claves que coinciden con un patrón específico.
- EXISTS key - Verifica si una clave existe o no.
- HSET key field value - Establece un valor para un campo específico dentro de una clave hash.
- HGET key field - Obtiene el valor de un campo específico dentro de una clave hash

Conclusión:

Para concluir, Redis es una base de datos NoSQL conocida principalmente por su performance y velocidad que tiene a la hora de operar con aplicaciones en tiempo real y que requieren de una base de datos que provea baja latencia. Su método key-value (clave-valor) para el almacenamiento de datos junto con la simplicidad de uso y su método para cargar y recuperan datos de forma rápida y eficiente, hacen a Redis una gran opción para aquellos proyectos que requieran software con mayor rapidez entre el usuario y la base. A la vez, permite crear estructuras complejas de datos que antes no eran posibles; tales como mapas, gráficos y listas. A lo largo del tiempo, redis ha ampliado en todo aspecto hasta tal punto de que hoy en día es una de las bases de datos más utilizadas a nivel mundial.

Bibliografía:

→ https://proximahost.es/blog/redis-almacen-datos-memoria/

→ https://redis.io/docs/data-types/tutorial/

→ https://aws.amazon.com/es/redis/

→ https://redis.io/docs/management/persistence/