



REDIS

Servidores Web de Altas Prestaciones
Grado en Ingeniería Informática
Universidad de Granada

David Fernando Guerrero Cardona

Contenido

Introducción	2
¿Cómo funciona Redis?	2
Beneficios de Redis	2
Almacén de datos en memoria	2
Estructuras de datos versátiles	2
Simplicidad y usabilidad	3
Replicación y persistencia	3
Disponibilidad y escalabilidad	3
Extensibilidad	3
Casos de uso	4
Almacenamiento en caché	4
Tablas de clasificación de videojuegos.....	4
Análisis en tiempo real	4
Machine Learning.....	4
Uso de Redis	4
Bibliografía	6

Introducción

En la actualidad la escalabilidad es parte de todo sistema web, Salvatore Sanfilippo, tuvo la idea de crear Redis tratando de mejorarla su proyecto, Redis ofrece tiempos de respuesta rápidos, estamos hablando de milisegundos, lo que permite que millones de solicitudes se puedan atender en la aplicación en tiempo real, ya sea gaming, publicidad, actividades financieras, IoT, entre otros.

Redis se define como (**R**emote **D**ictionary **S**erver), es un almacén de datos clave-valor en memoria. Se destaca por ser rápido y de código abierto y lo más importante se puede usar como base de datos, cache, agente de mensaje y cola. Se usa generalmente en servicios como análisis en tiempo real, chat/mensajería, streaming, tablas de clasificación, administración de sesiones.

¿Cómo funciona Redis?

Los Datos se encuentran en memoria, a diferencia de las bases de datos que estamos acostumbrados a usar, donde los datos están almacenados en discos o SSD, esto ofrece que no haya necesidad de acceder al disco, por lo tanto obtenemos velocidad evitando retrasos cuando accedemos a los datos en cuestión de milisegundos. Redis incluye estructuras de datos versátiles, alta disponibilidad, datos geoespaciales, scripts Lua, transacciones, persistencia en disco y soporte de clúster, lo que simplifica la creación de aplicaciones a escala de Internet en tiempo real.

Beneficios de Redis

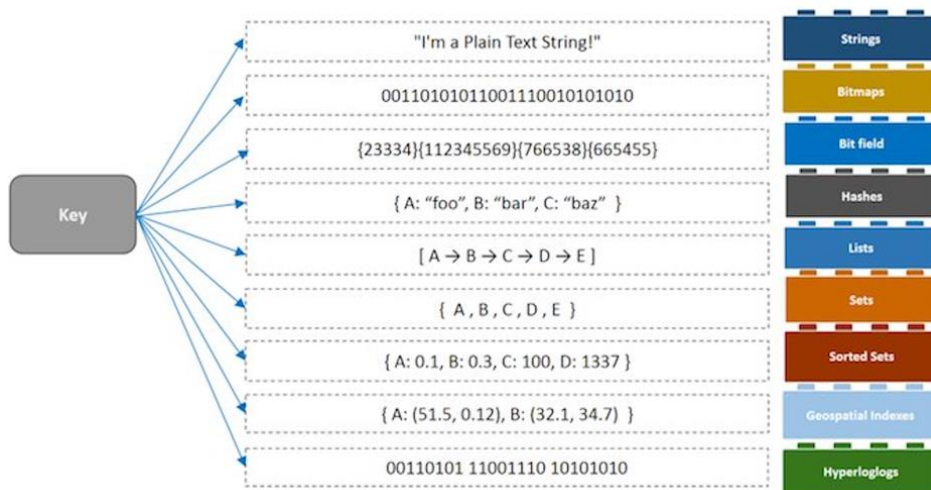
Almacén de datos en memoria

Como se ha dicho anteriormente los datos residen en la memoria principal del servidor, a diferencia de bases de datos como PostgreSQL, MongoDB o Cassandra. Normalmente las bases de datos se ven afectadas por el hecho de tener que acceder a disco, perdiendo tiempo, cosas que no afecta a Redis, esto implica que el servidor pueda admitir más operaciones y tiempos de respuesta más rápidos. El resultado es un incremento medio de rendimiento tanto de escritura como lectura.

Estructuras de datos versátiles

Redis tiene una gran variedad de datos que probablemente cumpla con los requisitos de la aplicación necesite.

- **Strings:** Datos de texto o binarios de hasta 512 MB.
- **List:** Colecciones de cadenas con orden de llegada.
- **Sets:** Conjuntos sin orden, donde tenemos intersección, unión y diferencia con otros del mismo tipo.
- **Sorted sets:** Conjuntos ordenados por valor.
- **Hashes:** Estructuras de datos para guardad lista de campos y valores.
- **Bitmaps:** Datos que permiten operaciones a nivel de bit.
- **HyperLogLogs:** Estructura de datos que permite estimar datos únicos en un conjuntos de datos (probabilidad).



1. Tipos de estructuras de Redis

Simplicidad y usabilidad

Redis permite en pocas líneas código acceso, guardado y uso de datos para las aplicaciones. Por ejemplo, si tu aplicación usa un hashmap y se quieren guardar datos, en la base de datos, basta con usar el hashmap que ofrece Redis, para no tener que utilizar cambios de formato. Básicamente sacar provecho de las estructuras, permite código simple para la manipulación e interacción con nuestros datos.

Lenguajes soportados: Java, Python, PHP, C, C++, C#, Javascript, Node.js, Ruby, R y Go. Entre otros.

Replicación y persistencia

Redis utiliza una arquitectura que permite la replicación de datos en múltiples servidores replica. Esto permite aumento de rendimiento en la lectura ya que dividimos nuestras peticiones se dividen entre los servidores, también en caso de perder conexión con nuestro servidor, redirigimos esas peticiones a otros.

Para la persistencia Redis soporta **point-in-time** backups (Copiar los datos de Redis a un disco).

Disponibilidad y escalabilidad

Redis ofrece una arquitectura con un servidor principal y replica con varios clusters, o con un único nodo. Esto permite que se puedan crear aplicaciones con alto nivel de disponibilidad lo que lleva a tener fiabilidad y un rendimiento estable. Así que en caso de tener que ajustar el tamaño del cluster basta con configurar las opciones de escalado basándonos en las necesidades de nuestra aplicación.

Extensibilidad

Al ser Redis un proyecto de código abierto cuenta con el apoyo de una comunidad muy activa. No hay limitaciones de proveedores ni tecnología por usar estándares abiertos.

Casos de uso

Redis permite que su funcionalidad sea aplicado a múltiples problemas, aquí hablaremos sobre los principales.

Almacenamiento en caché

En este caso Redis permite que crear una cache en memoria con el fin de obtener una alta disponibilidad reduciendo la latencia de acceso a los datos, además de incrementar la capacidad de procesamiento y aliviar la carga de la aplicación y de la base de datos que se esté usando. Que obtenemos con esto, tanto como tiempos inferiores y reducir costos en el backend. Ejemplos tan claros como consultas, en bases de datos, sesiones, páginas web como imágenes, archivos y metadatos.

Tablas de clasificación de videojuegos

Para aquellos desarrolladores de juegos donde se necesite una tabla de clasificación con datos generados en tiempo real. Redis ofrece estructuras de datos como son los conjuntos que permite mantener ordenado por valor, en este caso el puntaje de cada jugador.

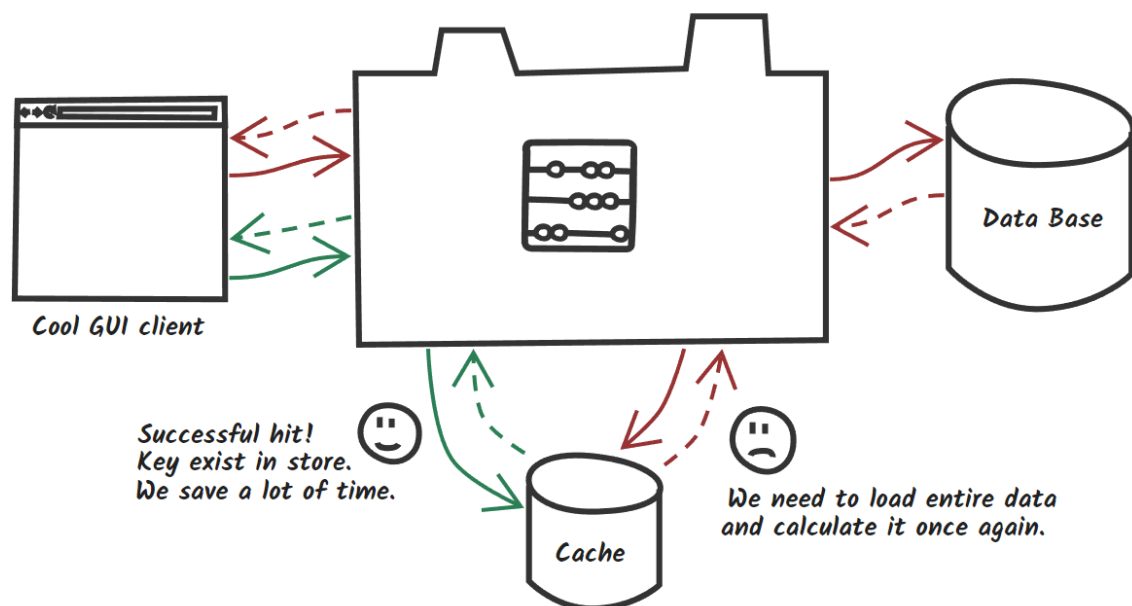
Análisis en tiempo real

Redis junto con otros tipos de aplicaciones como Kafka se puede usar como un almacén de datos para incorporar, procesar y analizar datos en tiempo real. Por ejemplo redes sociales, anuncios, o el IoT.

Machine Learning

Teniendo en cuenta que hoy en día los datos que se usan en el Machine learning exigen que se procesen grandes volúmenes de datos, para automatizar la toma de decisiones, Redis es perfecto para situaciones así. Ejemplos claros son como detección de fraudes en juegos o servicios financieros, subastas en tiempo real, viajes compartidos. Redis permite crear, entrenar e implementar diferentes modelos de Machine learning.

Uso de Redis



[3] Ejemplo de uso de Redis cache

Este ejemplo muestra dos escenarios: si seguimos el camino rojo nos damos cuenta de una situación particular llamada ***"cache miss"*** es en el cual donde ya no esta la información que necesitamos en la cache y tenemos que recalcularla desde el principio, por lo tanto perdiendo tiempo de respuesta. En cambio si seguimos el camino verde es un ejemplo claro que cuando realizamos una petición esta se encuentra en cache ***"cache hit"*** ya estamos ahorrando tiempo recursos, si necesidad de realizar operaciones extra.

Bibliografía

[1] Redis: in-memory data store. How it works and why you should use it. (2019). Retrieved from https://aws.amazon.com/redis/?nc1=h_ls

[2] Engel, J. (2019). Top 5 Redis use cases | ObjectRocket. Retrieved from <https://www.objectrocket.com/blog/how-to/top-5-redis-use-cases/>

[3] Spring Boot cache with Redis. (2019, 9 abril). Recuperado 5 mayo, 2019, de <https://medium.com/@MatthewFTech/spring-boot-cache-with-redis-56026f7da83a>