Tecnologie e applicazioni web

REDIS

Filippo Bergamasco (filippo.bergamasco@unive.it)
http://www.dais.unive.it/~bergamasco/
DAIS - Università Ca'Foscari di Venezia
Anno accademico: 2017/2018

Cos'è REDIS?

- Un database (structure store) **in-memory**:
  - Supporta stringhe / hash-maps / liste / insiemi / sorted sets / indici geospaziali
- Cache:
  - LRU eviction
- Message broker
  - Pub/Sub messaging paradigm

Caratteristiche interessanti

- Tutti dati restano in memoria (RAM). Questo garantisce accessi estremamente veloci rispetto ad un database standard
- I dati possono essere resi persistenti (non è garantita la persistenza immediata)
- Pensato per gestire carichi di lavoro enormi
- Supporto per operazioni atomiche (ex. Incremento di un valore)

Caratteristiche interessanti

- Supporto alle transazioni
- Single-threaded / architettura asincrona
- Scriptabile internamente in linguaggio LUA

Quando conviene usarlo?

Per tutte le operazioni di accesso concorrente ai dati o come layer di caching. In ogni caso, se i dati sono inferiori alla quantità di memoria della macchina...

Persistenza

I dati possono essere memorizzati sul disco, in modo asincrono o immediatamente (tradeoff rischio vs. performance). Due modalità:

- Forking e dump di tutte le strutture dati. Veloce ma introduce un ritardo variabile nella scrittura
- AOF (append-only file): viene loggata ogni operazione di scrittura in un file di log. La scrittura del file può essere resa sincrona con le operazioni (diventa però molto lento)

Utilizzo come cache

Uno degli utilizzi più semplici di REDIS è quello di semplice cache.

Due comandi:

- SET <key> <value>
  - Memorizza una certa stringa <value> data una chiave
- GET <key>
  - Ritorna la stringa precedentemente memorizzata con una data chiave

Utilizzo come cache

Ciascuna lettura al database “principale” aggiunge il valore ottenuto anche nella cache REDIS utilizzando come chiave l’id o un campo univoco dell’oggetto

Le letture successive possono verificare se il dato è già presente nella cache e in caso evitare la lettura dal database principale

REDIS Pub/Sub

Tra le funzionalità fornite vi è quella che segue il paradigma Publish/Subscribe e permette l'invio e ricezioni di dati tra più client senza conoscere il numero di client "in ascolto" in un determinato canale (simile ad UDP)

- I subscribers esprimono il loro interesse verso un determinato "canale"
- I publishers inviano messaggi (dati generici) in un canale

REDIS Pub/Sub

Il vantaggio del paradigma deriva dal fatto che si disaccoppiano le entità che generano i dati da quelli che li ricevono:

- I publishers non conoscono quanti e chi sono i subscribers di un determinato canale
- I subscribers non conoscono chi genera i dati, ma vengono notificati automaticamente ad ogni invio

REDIS Pub/Sub

REDIS permette ai subscribers di iscriversi a canali che corrispondono ad una certa regular expression:

Ex:

- Publishers pubblicano sui canali `it.news.topic1`, `it.news.topic2`, etc.
- Un subscribers si iscrive al canale `it.news.*` e riceve notifiche di entrambi i topics

Redis e Node.js

https://www.npmjs.com/package/redis

Libreria che fornisce una semplice interfaccia a tutti i “comandi” REDIS disponibili.

Pensata per essere basilare ma estremamente veloce

Lista comandi:

https://redis.io/commands