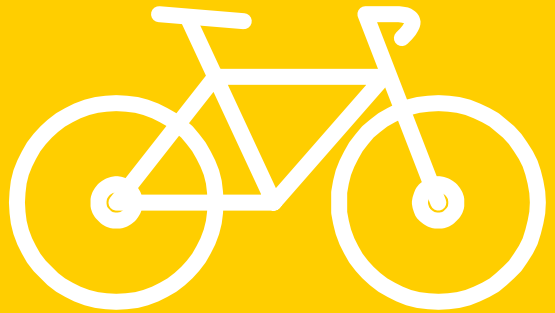
Hello!

I am Esmaeil Kazemi

I'm interested in learning how are you?

You can find me at @eskazemi



NOSQL

vs

SQL

eskazemi

Redis

Redis stands for Remote Dictionary Server

eskazemi

Map

Redis

1- Features

2- use

3- data type

4- Message Queue

5- Transactions

6- Pipelining

7- Lua Scripts

8- Persistence

9- Benchmarks

10- configuration

11- ACLs

12- Redis Cluster

13- Redis vs Memcached

14- Redis vs Hazelcast

eskazemi

Transaction-Pipelining-Lue script.pdf

سوالی که در انتها می خواهیم به آن پاسخ بدهیم این است که کدام دیتابیس in-memory مناسب برای ما

وجود دارد تعداد زیادی دیتابیس های in-memory

- Redis
- Memcached
- CouchDB
- Hazelcast

redis VS MEMCACHED

از آنجایی که هر دوی این ها برای موارد مشترکی مانند cache می توانند استفاده شود همواره مقایسه ای بین آنها انجام می شود و در پارامتر های زیادی از جنبه های مختلف می توان این دو رو بررسی کرد اما با توجه به تغییرات Redis در ورژن های بالاتر تقریبا انتخاب Redis امری بدیهی هست.

eskazemi

✓ data type: تفاوت بزرگ آنها در نوع داده هایی هست که نگه می دارند. memcached فقط توانایی ذخیره نوع ساده رشته را به عنوان مقدار دارد در صورتی که Redis دارای ۵ نوع Data Type اصلی هست از آنجایی که Redis دارای Data Type های بیشتری هست استفاده از آن می تواند باعث سهولت در نگه داری ساختار های داده ای پیچیده تر و کاربرد های متفاوت تری بشود.

✓ Memory usage : Redis عملکرد بهتری داد به خصوص در بحث آزاد سازی سریع فضای RAM استفاده شده بعد از flush کردن داده ها.

- نگهداری داده: توی Redis شما می تونین دیتاتون رو persistence کنین، یعنی اجازه بدین دیتاتون ریخته بشه توی دیسک و بعدش هر زمانی که بهش نیازش داشتین از دیسک دوباره انتقال داده میشه به موری. در حالی که memcached این فیچر رو ساپورت نمی کنه و اگه سرورتون پایین بیاد یا ری استارت بشه دیتای ذخیره شدتون نابود میشه. البته در صورتی که Memcached نیاز به ابزار های third-party source برای dumpگرفتن داده ها دارد. البته ناگفته نماند که در بعضی موارد هم مانند سرعت خواندن و نوشتن و یا Scaling این دو بسیار به هم نزدیک هستند و برتری خاصی حس نمی شود.

- کلاسترینگ: Redis به طور داخلی دارای فیچر رپلیکیشن و پارتیشنینگ هستش و همین فیچر برای hash کردن مناسبش می کنه. توی memcached ما این فیچر ها رو نداریم در memcashed ولی چون memcached دیزاین ساده ای داره، برای اسکیل کردن به صورت horizontally خیلی مناسبه، یعنی می تونیم کلید هامون رو توی چندین memcached ذخیره کنیم.

- همزمانی: memcached از ساختار multithreade استفاده می کنه که بهش اجازه میده چندین کانکشن رو به طور همزمان با استفاده از ترد های جداگانه هندل کنه. در حالی که Redis برای هندل کردن درخواست ها از یک ترد استفاده می کنه که این موضوع در memcached می تونه باعث بالا رفتن پرفورمنس با استفاده از همزمانی بشه.

جمع بندی

اگه تنها هدفتون کش کردن دیتاس، اگه به یه سیستم کش ساده علاقه دارین، اگه نمی خواین از دیتا تایپ های پیشرفته ای استفاده کنین، اگه دیتای زیادی دارین و می خواین با استفاده از چندین core از پرفورمنس Multithread استفاده کنین، اگه می خواین سیستم کش کردنتون رو همون جوری که رشد می کنین به صورت horizontally گسترش بدین، در همه این موارد از memcached استفاده کنین.

eskazemi

redis vs  MEMCACHED

redis VS  MEMCACHED

Architecture

Redis is an in-memory data store that supports data replication and persistence for durability.

Memcashed is a high-performance, distributed memory object caching system

Replication

Data replication is supported.

Data replication is not supported.

concurrency

It is a single-threaded architecture.

It is a multi-threaded architecture.

redis VS  MEMCACHED

Data Model

Advanced Data Structures (e.g., lists, sets, hashes)

uses a simple key-value store, which is great for caching results

eskazemi

Transactions

Transactions is supported.

Transactions is not supported.

Data eviction

LRU (Least Recently Used).

LRU, LFU (Least Frequently Used), Random, TTL (Time-to-Live)

Persistence

Optional persistence (e.g., snapshots, append-only files)

No built-in persistence

redis VS  MEMCACHED

Scalability

offers primary-replica replication, allowing it to scale reads. With Redis Cluster, it can also scale writes.

can scale horizontally, but it does not support replication or partitioning out of the box

Performance

Faster than Redis for small value sizes, but slower for larger values

Generally slower than Memcached for small value sizes, but faster for larger values

Popular Use Cases

Caching, session management, message broker, real-time data processing, job queue

is used by platforms like YouTube, Reddit, and Facebook to speed up dynamic web applications by reducing database load.

redis vs

HAZELCAST

redis VS

HAZELCAST

Architecture

Redis is an in-memory data store that supports data replication and persistence for durability.

Hazelcast is an in-memory data grid that provides distributed data structures and supports execution of distributed computations

Data Model

Redis supports multiple data types such as strings, hashes, lists, and sets

Hazelcast provides distributed data structures including map, queue, multimap, topic, and lock

redis VS

HAZELCAST

Persistence

Redis supports disk persistence, which means it can store data permanently

Hazelcast supports persistence through the MapStore and QueueStore interfaces, allowing data to be stored and reloaded

eskazemi

Scalability

Redis offers primary-replica replication, allowing it to scale reads. With Redis Cluster, it can also scale writes.

Hazelcast is highly scalable. It allows for dynamic clustering and data partitioning, scaling well for both reads and writes.

Popular Use Cases

Redis is used by Twitter for storing user sessions and timelines, and for real-time analytics, caching, and message brokering

Hazelcast is used by financial institutions for low-latency data processing, caching, web session clustering, and distributed computing.

redis vs RDBMS

Redis Vs RDBMS

معیار ها	Redis	RDBMS
Data storage	Stores data in memory, with optional disk persistence	Stores data on disk
Data model	Supports a key-value data model and various data structures such as strings, hashes, sets, lists, and sorted sets	Supports a relational data model with tables, columns, and rows
ACID compliance	Does not guarantee ACID compliance, but offers some ACID-like guarantees	Generally supports full ACID compliance
Concurrency control	Uses a single-threaded architecture, which can limit its concurrency capabilities	Supports multi-threading, allowing for greater concurrency

eskazemi

Redis Vs RDBMS

eskazemi

معیار ها	Redis	RDBMS
Performance	Offers high performance due to its in-memory storage and efficient data structures, capable of processing hundreds of thousands of requests per second	Performance may be impacted by disk I/O and relational data modeling
Scalability	Can scale horizontally through sharding or replication	Can scale vertically through hardware upgrades or partitioning
Use cases	Ideal for real-time applications and scenarios where high throughput is required	Suitable for applications with complex data structures and relationships, as well as batch processing and reporting
Query language	Offers a limited set of commands and does not support SQL	Typically supports SQL or a similar query language
Cost	Open-source version is available for free, with paid options for enterprise-level features	Often requires paid licenses and ongoing maintenance costs

eskazemi

eskazemi

ماژول های Redis

1. RedisJSON is a module for Redis that allows developers to store, manipulate, and query JSON data.
2. RedisBloom is a module for Redis that provides probabilistic data structures such as Bloom filters and HyperLogLog
3. RedisTimeSeries is a module for Redis that provides time-series data functionality. It allows developers to store, manipulate, and query time-series data.
4. RedisGraphQL is a module for Redis that provides GraphQL functionality. It allows developers to query Redis data using the GraphQL query language.
5. RedisGraphAI is a module for Redis that provides graph algorithms and analytics functionality.
6. RedisTimeFlow is a module for RedisTimeSeries that provides anomaly detection and forecasting functionality.
7. RedisRaft is a Redis module that provides distributed consensus and replication functionality.
8. RedisJSONAI is a module for Redis that provides machine learning functionality for JSON data.
9. RedisCell is a Redis module that provides cell-based rate-limiting functionality for APIs and web applications.

Thanks!

Any questions?

You can find me at:

- @eskazemi
- m.esmaeilkazemi@gmail.com