NDBI040 - Redis

Miriam Supeková

Key-value data structure storage

- Open source
- In memory key-value database
- Master-slave architecture
- Implemented in C
- MapReduce through RedisGears engine
- Data consistency
    - Eventual Consistency
    - Causal Consistency in Active-Active databases
    - Strong Consistency with Redis Raft
    - Strong Eventual Consistency with Active-Active
- No foreign keys

Installation

- On Ubuntu systems

```
sudo apt install redis-server

sudo systemctl enable redis-server
```

- Edit the config file, change the supervised directive to `systemd`

```
sudo nano /etc/redis/redis.conf

sudo systemctl restart redis.service

sudo systemctl status redis

redis-cli
```

Data Persistence

- Data is stored in memory
- Keys can have various TTL
- Various persistence options
    - RDB - Redis Database
    - AOF - Append Only File
    - No persistence
    - RDB + AOF

Replication

- Master-slave replication
- Any number of replicas
- Replicas can accept connections from other replicas
  - Replicas can connect with other replicas in a cascading structure
- Replication is non-blocking on the master side
- Useful for read scalability and data redundancy
- Asynchronous replication
- Configuration through replica configuration file using REPLICAOF command with specified IP address and port

Data models, Data types

- Redis database - collection of objects
- Databases are identified by integer IDs

Keys

- Strings
- binary safe - any binary sequence can be used as key (“”, “foo”, content of JPEG file)

Values

- Strings, Lists, Sets, Hashes, Sorted Sets, Bitmaps, HyperLogLogs, Streams, Geospatial indexes

Redis Extensions

- Redis Stack (data models extension)
    - supports following as values for string keys
        - JSON documents (RedisJSON)
        - full-text search (RediSearch)
        - time series (RedisTimeSeries)
        - graph data models (RedisGraph)
        - probabilistic data structures (RedisBloom)
    - RedisInsight
- Redis Enterprise Edition - Redis Enterprise Cloud
- ScaleGrid for Redis

Query language

- No query language such as SQL
- Redis allows to insert and retrieve key(s)/value(s), modify value(s) - add, remove, intersection, increase/decrease value, …
- Redis commands - different for each data type
  - https://redis.io/commands/
- Redis does not allow joins we know from SQL but allows intersection of key/value pairs saved in sets and sorted sets
- Redis allows the operation sort for values saved in lists, sets or sorted sets
  - values can be sorted according to alphabet or numerical values
- Supports transactions

Transactions

- Commands MULTI, EXEC, DISCARD, WATCH
- All commands in a transaction are serialized and executed sequentially = commands are executed as a single isolated operation
- EXEC - triggers the execution of all the commands in the transaction
- MULTI - queues all following commands, until the EXEC is called
- DISCARD - aborts the transaction
- WATCH - allows to monitor keys and their changes

Publish/Subscribe System

- Subscribers subscribe to some channels where the publishers publish messages
- Key commands - PUBLISH, SUBSCRIBE, UNSUBSCRIBE

Operations with key/values pairs

- basic commands - see the tutorial slides, p.11 and basic operations for the main data types on p.12-23
- STRINGS -> APPEND key value; GETDEL key; GETEX key [EX]; GETRANGE key start end; LCS key1 key2;
- SETS -> SDIFF key; SINTER key; SMOVE source destination member, SPOP key [count]; SREM key member;
- SORTED SETS -> ZDIFF numkeys key; ZLEXCOUNT key min max; ZPOP numkeys key; ZRANK key member; ZUNION numkeys key; ZSCORE key member;

Operations with key/values pairs

- HASHES -> HINCRYBY key field increment; HSCAN key cursor; HSETNX key field value; HSTRLEN key field;
- BITMAPS -> BITCOUNT key; GETBIT key offset; SETBIT key offset value; BITPOS key bit;
- HYPERLOGLOGS -> PFADD key; PFDEBUG; PFDEBUG; PFSELFTEST
- STREAMS -> XADD key, XDEL key id, XLEN key, XRANGE key start end; XREAD [COUNT count]
- GEOSPATIAL INDICES -> GEOADD key; GEOPOS key member; GEORADIUS key longitude latitude
- ... and many others

Advantages & Disadvantages

- Key/value pairs of size 512 MB
- Redis Hashing
- Data replication
- Redis cache
- APIs in various prog. languages
- Publish/subscribe messaging system
- Huge amounts of data stored in cache
- RESP
- Transactions

- Big RAM due to in-memory storage
- No query language, no support for RA

Sources

- https://redis.com/
- https://redis.io
- https://en.wikipedia.org/wiki/Redis
- https://dzone.com/articles/10-traits-of-redis
- https://medium.com/weekly-webtips/redis-what-and-why-pros-cons-ae2f5bc750fd
- https://stackoverflow.com/questions/10906246/what-is-the-disadvantage-of-just-using-redis-instead-of-an-rdbms
- https://db-engines.com/en/system/Redis
- Lecture and tutorial slides