CASTLE ROCK

Memory Is The New Disk, Say Hello To Redis

by Bratish Goswami

whoami

Agenda

- The Background
- Concept of NoSQL and some projects
- What is Redis?
- Redis Data Types
- Clients
- Supported datatypes and operations
- Performance
- Adopters

The Background

- Advancement of Web 2.0, the read/write web
- Exponential data growth in social networks
- With very high magnitude of data, IO become very costly.

Memory is the new disk,
disk is the new tape.
— Jim Gray

Why Memory?

- Relative latency for SSD is 1000x than memory
- Relative latency for disk is 100x than SSD

Concept of NoSQL

An informal, loosely-defined term for non-relational, structured data storage systems.

Examples:

Hbase, Cassandra, Tokyo Tyrant, MongoDB, memcached, CouchDB, and Redis

Memcached

- Everything is a string
- Set or Get data
- Non-persistence

An Italian solution

- Main developer Salvatore 'antirez' Sanfilippo with Pieter Noordhuis
- Key-value store
- In memory
- Data structure server
  - *Redis is a collection of data structures exposed over the network*

Redis comes with

- Memcached like simplicity
- More datatypes
- More commands
- Persistence with non-blocking I/O
- Publish-subscribe channels

Installation

- Current version is 2.2.8
- Clone https://github.com/antirez/redis.git
  or
- wget http://redis.googlecode.com/files/redis-2.2.8.tar.gz
- tar xzf redis-2.2.8.tar.gz
- cd redis-2.2.8
- make

Redis Data Types

- Strings
- Lists
- Sets
- Sorted Sets
- Hashes

Clients

- src/redis-cli
- Full list of clients at http://redis.io/clients
- For Ruby there's *redis-rb*
  - Pure Ruby
    - require 'redis'
  - Can work with C client HiRedis as a C extension
    - require 'redis/connection/hiredis'
    - Use hiredis when you have large array replies and/or large pipelines of commands.

Keys

- EXPIRE / EXPIREAT
- PERSIST
- TTL
- TYPE
- RENAME
- KEYS
- etc...

Strings

- SET
- GET
- INCR
- DECR
- etc...

Lists

- LLEN
- LPOP / RPOP
- LPUSH / RPUSH
- RPOPLPUSH
- etc...
- Implementation of Queue
  - Not a native type, but can easily be implemented
    - RPUSH
    - LPOP

Sets

- SADD
- SPOP
- SINTER
- SDIFF
- SUNION
- etc...

Sorted Sets (zsets)

Commands are almost similar to ***set*** commands, except a ***score*** is needed for sorting.

- ZADD
- ZREM
- ZRANGE
- ZRANGEBYSCORE
- etc...

Hashes

- HLEN
- HGETALL
- HKEYS
- HGET
- HINCRBY
- etc...

Pub/Sub

- Implement the Publish/Subscribe messaging paradigm
- ***SUBSCRIBE***, ***UNSUBSCRIBE*** and ***PUBLISH*** commands

Persistence

- Snapshotting
  - ***save N M*** in configuration file
    - Save the dataset every N seconds if there are at least M changes in the dataset
  - ***BGSAVE*** commands
- Append-only file
  - ***appendonly yes*** in configuration file
- Log rewriting
  - ***BGREWRITEAOF***

Performance

- src/redis-benchmark -q -n 1000000

  PING: 62167.10 requests per second

  SET: 62774.89 requests per second

  GET: 61993.86 requests per second

  INCR: 63360.45 requests per second

  LPUSH: 62849.98 requests per second

  LPOP: 59762.75 requests per second

Into the wild

- Big names using Redis:
  - Github
  - EngineYard
  - Digg
  - StackOverflow
  - and many more...

Future Ideas

- Redis Cluster
- Redis Disk Store
- Redis Scripting

References

- A CONVERSATION WITH JIM GRAY
  - http://queue.acm.org/detail.cfm?id=864078
- redis.io

Thank you

@bratishgoswami

bgoswami@crri.co.in

Questions...