

Redis 4.0 and over

Salvatore Sanfilippo
@antirez

redisday
TLV 2017

What Redis 4.0 brings to the table...

Main features

- Modules!
- Seriously improved replication (PSYNC v2)
- Caching improvements, especially LFU
- Non blocking DEL / FLUSHDB
- Faster & less memory used

Redis 4.0 status

- RC2 released 2 months ago
- RC3 soon to be released
- Surprisingly little serious issues reported so far
- However... a few things to fix / add before 4.0-final
- ETA: about 2 months

Neural Redis

- Redis Modules System example
- Neural networks you can train on-line
- What template this user is likely to pick?
- Is this user likely to upgrade to PRO?
- Non linear regression problems.
- <http://github.com/antirez/neural-redis>

Redis 4.2 roadmap

- Redis Cluster improvements
- Better embedded systems support (Raspberry & more)
- Stream data structure

Cluster improvements

- Used by many, not receiving enough ♥
- Speed improvements (we found a bug that makes it slower than normal Redis instances!)
- Better tools (redis-trib -> redis-cli, backups, ...)
- Faster resharding
- Multiple data centers support as first class citizen

Streams

- It's, basically a log... BUT
- Memory efficient
- Indexed by time
- Supports range queries
- Client groups (Hello Kafka!)
- You can go back in time (Just like Terminator)

Embedded / IOT

- Redis has a very small footprint
- Is reasonably fast in very small computers
- Looks like an ideal Internet Of Things DB! :-)
- Improve ARM support / unaligned addresses accesses & so forth.
- Always test in Raspberry PI 3 & PI zero
- Provide great documentation for this use case

Project into the project: Disque as a module

- Disque is a message broker, a fork of Redis
- Disque is a Clustered-by-default AP system
- Keeping Redis & Disque in sync was challenging
- Let's rewrite it in terms of Redis Module
- Redis Modules need a Cluster API...

Tries, a cool data structure

- Suggested by Dvir because of Redis Cluster speed issue
- Used in Dvir search module
- Tries are a great data structure, compact, fast, ordered!
- May replace dict.c, Redis hash tables
- Current HT: needs incremental rehashing, is a bit memory wasteful, challenging to iterate...

Btw... why should you switch to 4.0?

- MEMORY command! “USAGE” and “DOCTOR” subcommands.
- LFU is better than LRU, being LRU just an approximation of LFU.
- PSYNC2 means peace of mind: resynchronize with slaves after a failover, resynchronize after a slave restart \o/
- Non blocking DEL is very useful sometimes.
- Have access to user-contributed modules for vertical use cases.
- Start testing now, put in production after 4.0-final :-)