

Redis, 7 years and more

@antirez
Redis Labs

2009

- EXPIRE
- Sets and Sorted Sets
- Non blocking replication
- AOF

2010

- Hash data type
- Virtual Memory project started
- Pub/Sub
- Redis Cluster project started
- VM abandoned, Diskstore project started (!)

2011

- More work to Redis Cluster (alpha at the time)
- IPv6
- Lua scripting
- Diskstore abandoned. No new attempts.

2012

- Bit operations
- Special encodings for small values
- Sentinel V1
- SCAN family commands

2013

- Keyspace events
- Replication PSYNC support
- CONFIG REWRITE
- Performance boosts
- Sentinel V2

2014

- Lexicographical ranges
- HyperLogLog
- Diskless replication
- New List type encoding introduced

2015

- I join RedisLabs !!!
- Redis 3.0 with Cluster released
- Geo indexing API
- Non blocking DEL and FLUSH commands
- Lua debugger

2016

- BITFIELD command

What's new in 3.2

Lua new replication modes

- Now you can write script with side effects, random commands, ...
- Selective replication of certain commands to slaves.

Expires consistency

- Slave and Master now agree much better about what keys already expired.
- Expires are still up to the master, but the slave, for read operations, can show a sane behavior.

RDB improvements

- RDB has now auxiliary informations inside.
- It's faster to load since it resizes the hash table before loading the keys.
- So... no rehashing needed (is costly).

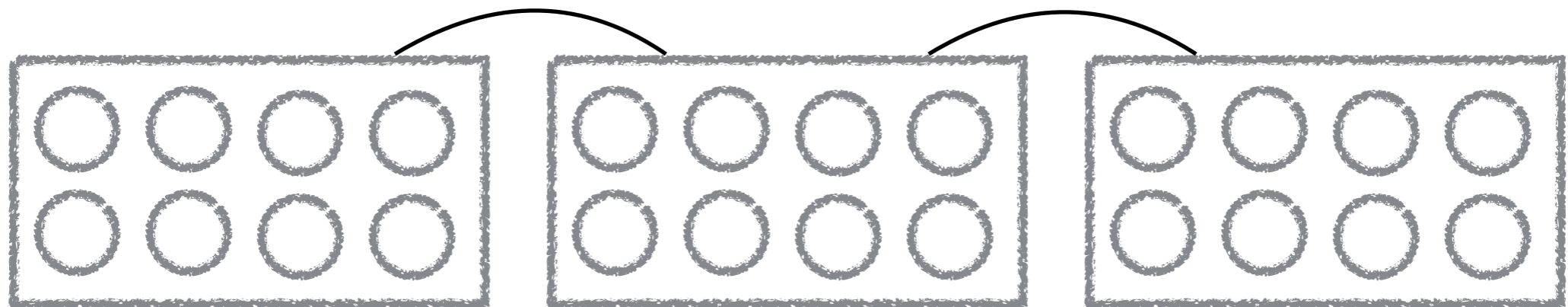
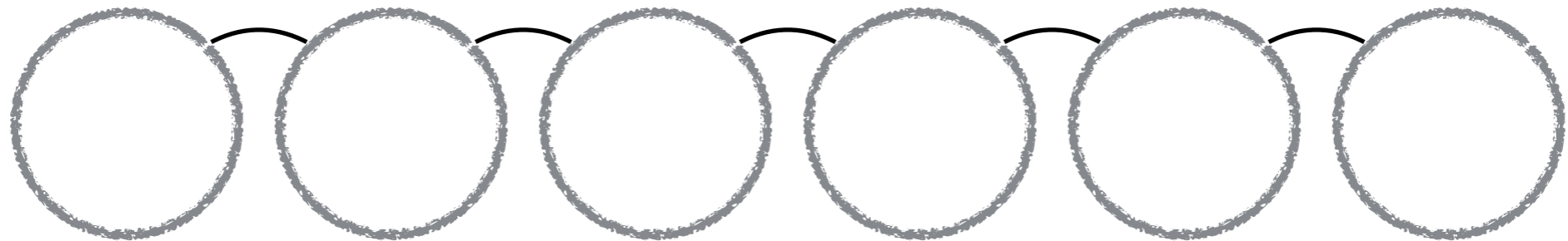
Geo commands

- 49 commits, 899 insertions, **1429** deletions.
- API simplified, JSON API removed entirely.
- An useless indirection layer removed. Geo indexing API taken from Ardb simplified & improved.
- Radius queries fixed to really report items in range.
- Much faster (may be yet a lot faster... work needed)

Quicklists

- Idea initially conceived by Pieter & me, never implemented.
- Later we saw Twitter implementing it, but just the minimum they needed.
- Finally Matt Stancliff provided a good implementation.
- Sun He and I refined the implementation and improved the on-disk serialization.

Quicklists



Improved ZADD

- NX, XX, CH, INCR.
- Better capabilities via options.
- Newcomers can ignore them.
- Complexity is disclosed incrementally.
- Commands list does not becomes huge.
- What about **Markov Chains**?

SPOP with count

- Ability to pop multiple items from set in a single call.
- Small change, but required some reworking of the replication code.

Lua debugger

- Writing complex scripts is no longer a nightmare.

Cluster improvements

- Rebalancing of the cluster.
- CLUSTER API improvements.

BITFIELD

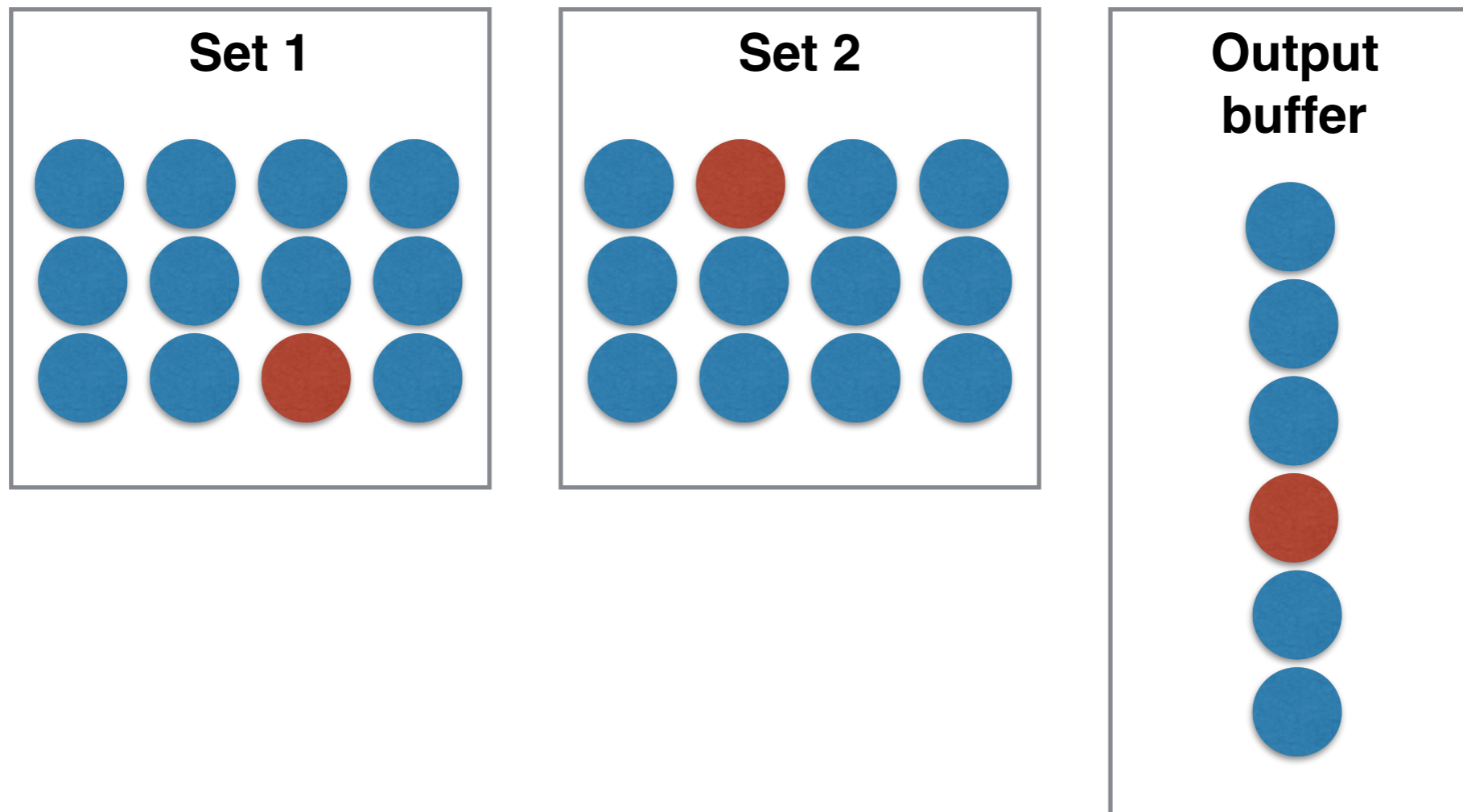
- Arrays of counters
- Overflow handling

Lazy freeing of objects

Lazyfree basics

- It means: **O(1)** DEL **synchronous work**.
- Plus: **O(N)** DEL **asynchronous work**.
- The result: **DEL is no longer blocking**.
- API: **UNLINK** and **ASYNC** option for **FLUSHDB** and **FLUSHALL**.

Shared objects



More stuff

- Better SDS library.
- More useful and polished crash reports.
- CLIENT REPLY to suppress replies for bulk inserts.
- DEBUG RESTART and CRASH-AND-RECOVERY.
- HSTRLEN command.
- Too many to show them all.

Ideas for the future

Threaded I/O

- Spawn N threads at startup.
- Each thread does multiplexing like now.
- Each client is pinned to a thread when connecting.
- We can finally **scale write, read, multiplexing**.
- **Global Lock**. Similar to **memcached**.

Threaded slow ops

- Client is executing **SUNION foo bar**.
- Block it (like in blocked.c).
- Lock the key in a **lock hash table**: no mutex per key! Single mutex for the lock hash table.
- Perform operation, **including sending the output** in the client output buffer, in a different thread.
- Unblock the client.

N-dimensional range queries

- Already implemented as a Ruby library.
- Uses bit to bit interpolation.
- We want to add it as a native API.
- **NADD** myindex 3 1.4 200.4 4321 element
- **NRANGE** myindex [0 1] [10 20] (30 40]

Bloom filters

- API is the thing that matters.
- I would love **no parameters** (like HyperLogLogs).
- Auto scaling, without false negatives.
- Alternatives to bloom filters with same semantics look promising compared to bloom filters.

The end

ask me anything here or mentioning @antirez