

Redis

Redis

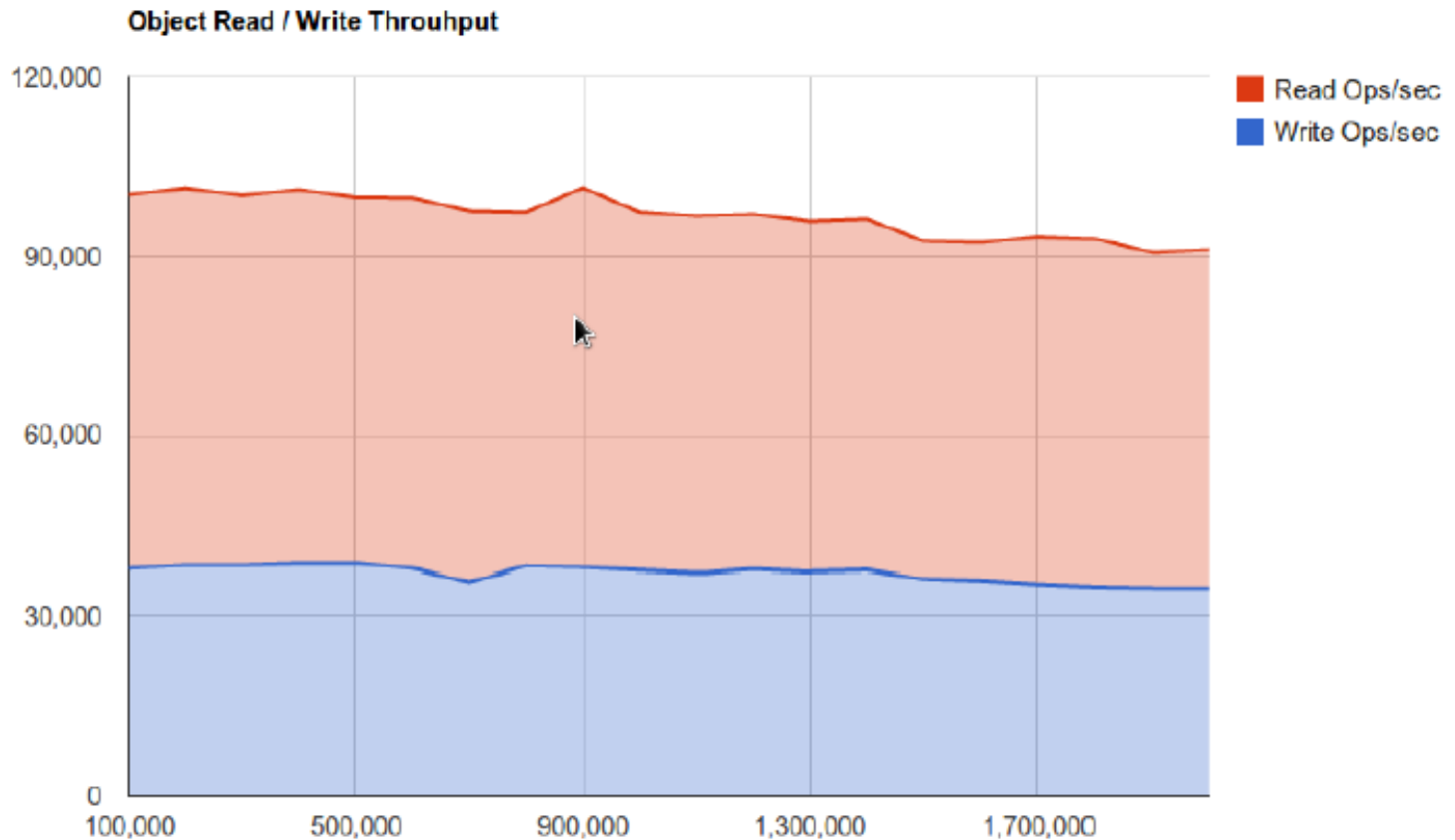
- REmote DIctionay Server
- „advanced, fast, persistent key-value database”
- Adatstruktúra szervernek is szokták nevezni

Redis

- C nyelven írt
- In-memory
- Kevés függőség, kis méret (1,3 MB)
- Gyors
- Egy-szálú
- Rengeteg kliens library
- Első verzió: 2009 március

Gyors

Performance with growing data

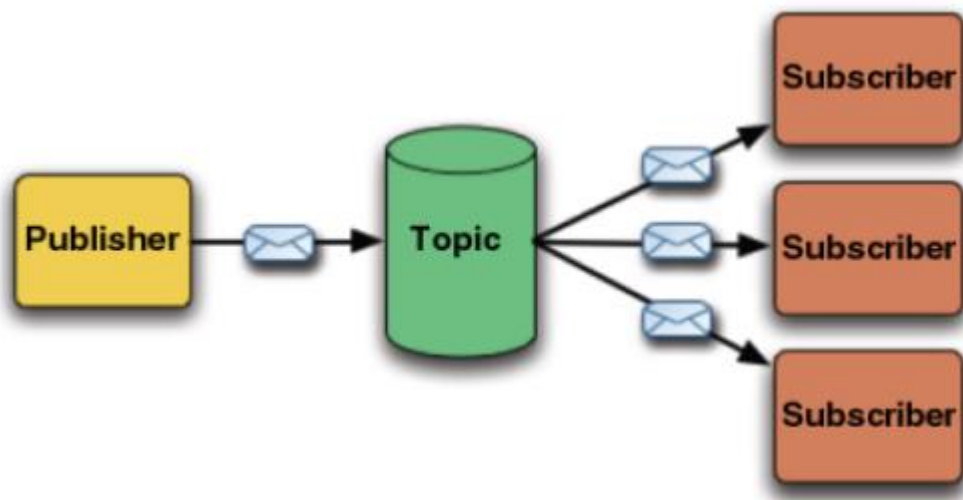


Jellemzők

- Kulcs-érték adatbázis, ahol az érték:
 - string, list, set, sorted set, hash
- Replikáció: master-slave
- Publish – Subscribe
- Idő limités adat
- Tranzakciós

Publish/Subscribe

- Publisher adatot küld a topikba
- Kliensek feliratkoznak a topikokra
- Ha új adat érkezik a topikban a kliensek megkapják



Perzisztencia

- Snapshotting mode
 - Bináris dumpok, minden x másodpercben vagy y művelet után
- Append Only File (AOF)
 - Minden művelet ki lesz írva egy fájlba
 - Újraindításnál minden művelet újra lefut
- Replikálás

Partícionálás

- Redis cluster
 - Minden node-nak elérhetőnek kell lennie a kliensek felé
 - Minden node kommunikál a többi node-dal
 - A shard-olás hash slot-okkal történik

Hash slots

- 16384 hash slot van
- Egy adat tárolása: adat CRC16 modulo 16384
- pl.:
 - A node : 0 5500
 - B node : 5501 11000
 - C node : 11001 16384
- node hozzá vétel
 - pár hash slot másolása az új node-ra
- node eltávolítása
 - a rajta lévő hash slotok másolása a többi node-ra
- Nincs szükség a cluster leállítására

Kik használják?

- stackoverflow
- github
- blizzard
- twitter
- pinterest
- snapchat
- digg
- flickr
- vmware (sponsor)
- ...

Támogatott nyelvek

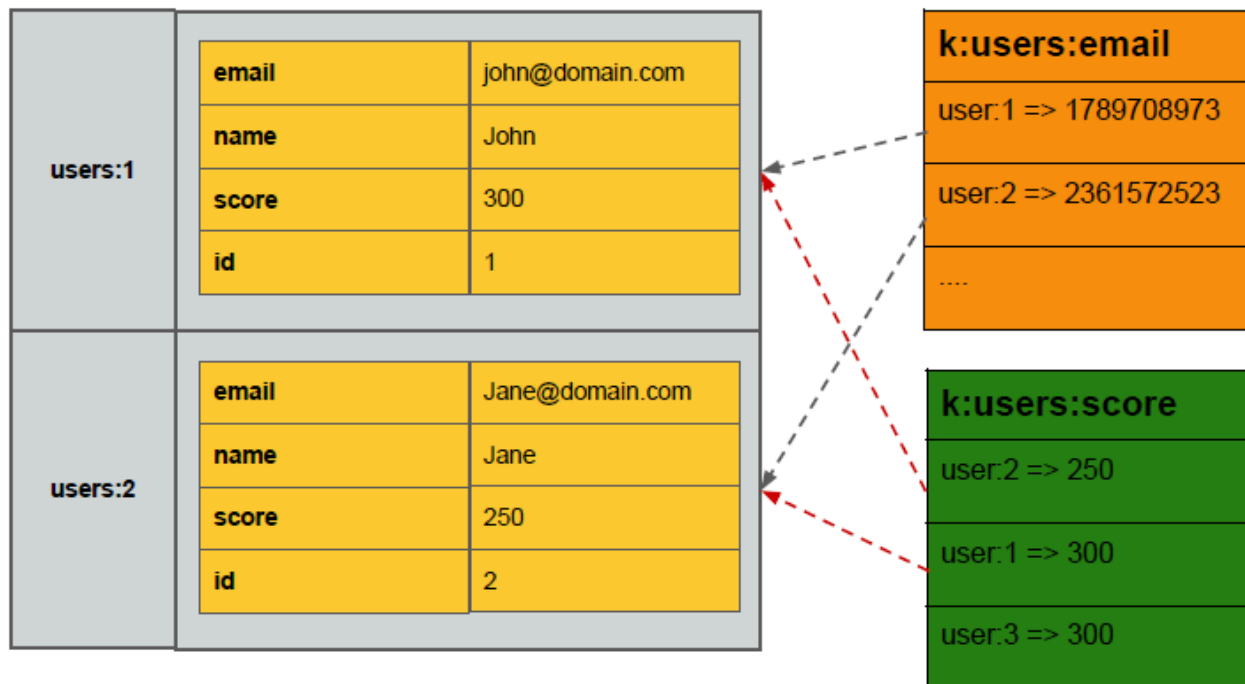
ActionScript	Bash	C	C#	C++	Clojure
Common Lisp	Crystal	D	Dart	Delphi	Elixir
emacs lisp	Erlang	Fancy	gawk	GNU Prolog	Go
Haskell	Haxe	Io	Java	Javascript	Julia
Lasso	Lua	Matlab	mruby	Nim	Node.js
Objective-C	OCaml	Pascal	Perl	PHP	Pure Data
Python	R	Racket	Rebol	Ruby	Rust
Scala	Scheme	Smalltalk	Swift	Tcl	VB
VCL					

Kulcsok

- Hosszú kulcsok nem jók
- Szóköz, újsor karakter ne használjunk
- „object-type-id:id:field” jó megoldás, pl.:
„user:1001:name”

Indexek

- Manuális indexek rendezett halmazzal
- Gyorsabb adatelérése a score és az emailnek (hashelt)



Alap parancsok

- Változó típus lekérdezése
 - TYPE mylist
- Help lekérdezése
 - HELP
- Redis információk
 - INFO

String érték

- Érték beállítása
 - SET mykey somevalue
- Érték lekérdezése
 - GET mykey
- Érték törlése
 - DEL mykey

String érték

- Ne állítsa be ha létezik
 - SET mykey somevalue **nx**
 - SETNX mykey somevalue
- Akkor is állítsa be a létezik
 - SET mykey somevalue **xx**

Szám értékek

- Beállítás
 - SET counter 100
- Növelés
 - INCR counter
 - INCRBY counter 50
- Csökkentés
 - DECR counter
 - DECRBY counter 30

Miért kell a növelésre függvény?

- Mi is meg tudnánk oldani:
 - $x = \text{GET count}$
 - $x = x + 1$
 - $\text{SET count } x$
- A probléma a következő:
 - A lekérdezi a countot, B lekérdezi a countot
 - A növeli, B is növeli
 - Mi lesz az eredmény

Miért kell a növelésre függvény?

- Mi is meg tudnánk oldani:
 - $x = \text{GET count}$
 - $x = x + 1$
 - $\text{SET count } x$
- A probléma a következő:
 - A lekérdezi a countot (10), B lekérdezi a countot (10)
 - A növeli (11), B is növeli (11)
 - Helyes: 12

Kulcs érték

- Több érték beállítása egyszerre
 - mset a 10 b 20 c 30
- Több érték lekérdezése
 - mget a b c

Idő limit

- Érték beállítása
 - SET key some-value
- Idő beállítása
 - EXPIRE key 5
- Vagy egyszerre
 - SET key 100 ex 10
- Idő limit lekérdezése
 - TTL key

List

- Érték hozzá vétele a listához
 - RUSH mylist A
 - LPUSH mylist first
 - RUSH mylist 1 2 3 4 „foo”
- Lekérdezés
 - LRANGE mylist 0 -1

List

- Adat kivétele a listából
 - RPOP mylist
 - LPOP mylist
- Blokkolt olvasás a listából
 - BRPOP mylist 5 //5 mp-ig vár, ha nincs benne elem
- Utolsó X mező megtartása a listából
 - limitekre hasznos
 - LTRIM 0 2

Hash

- Érték beállítása
 - HSET map key value
 - HMSET user:1000 username antirez birthyear 1977 verified 1
- Érték lekérdezése
 - HGET user:1000 username
 - HGET user:1000 username birthday
 - HGETALL user:1000

Set

- Objektumok közötti kapcsolatokra használják
- Érték beállítása
 - SADD myset 1 2 3
- Lekérdezés
 - SMEMBERS myset
- Megtalálható-e az érték a halmazban
 - SISMEMBER myset 3

Set

- SET példa: hírek és a hozzá tartozó címkék
 - hírekhez a címkék kapcsolása
 - SADD news:1000:tags 1 2
 - címkékhez a hírek kapcsolása
 - SADD tag:1:news 1000
 - SADD tag:5:news 1000
 - SMEMBERS news:1000:tags
 - SMEMBERS tag:1:news

Set

- Halmazok uniója
 - `SUNIONSTORE newSet Set1 Set2`
- Halmaz másolása
 - `SUNIONSTORE newSet Set1`
- Elem kinyerése a halmazból – véletlen-szerű
 - `SPOP newSet`
- Halmaz elemszáma
 - `SCARD newSet`

Sorted Set

- Set halmaz score értékkel
- pl.: ZADD hackers 1912 „Alan Turing”
ZADD hackers 1965 "Yukihiro Matsumoto,,
- Itt a score most a születési idő.

Sorted Set

- Lekérdezés

- ZRANGE hackers 2 4 //2-4 elemek lekérdezése
- ZRANGE hackers 0 -1 //össze elem 0-tól
- ZREVRANGE hackers 0 -1 //csökkenő sorrendbe
- ZRANGE hackers 0 -1 withscore //score kiiratása
- ZRANGEBYSCORE hackers -inf +inf //végtelenek közötti
- ZRANGEBYSCORE hackers (1 2 //1< <=2
- ZREMRANGEBYSCORE hackers 1 3 //1 és 3 közötti értékek eltávolítása

Tranzakció kezelés

- Tranzakciók használata több parancs végrehajtásához
 - MULTI //tranzakció kezdete
 - //parancsok
 - EXEC //tranzakció végrehajtása
- Amíg nincs futtatás, addig nem olvas értéket!

Köszönöm a Figyelmet!