

Redis

In-memory data structure store



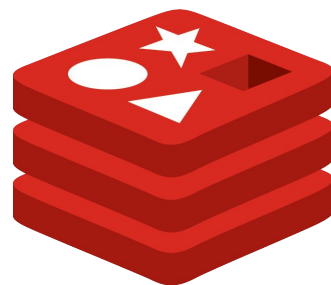
Sergej Kurakin

Kas yra Redis?

Duomenų struktūrų saugykla serverio atmintyje

Naudojama kaip:

- duomenų bazė
- kešas (angl.: cache)
- žinučių brokeris (angl.: message broker)



redis

Kaip aš susipažinau

Tai buvo prieš keletą metų

Projektas su “cache” kietajame diske per kelis serverius

Apribotas iš OS pusės: RHEL4

Apribotas pagal laiką: reikia čia ir dabar

PHP 5.2 (o gal PHP 5.1)

Reikalavimai

[?] Centralizuotas

[?] Restart'o atveju neprarasti bent dalies "cache"

[?] Minimalus laikas apšilimui po restart'o

[?] Patikimas

[?] Lengvai diegiamas (RHEL4)

[?] "Failover" galimybė

Ką žiūrėjau?

Memcached

MemcacheDB

MySQL (key-value)

MongoDB

Redis

Kažkas labai sudėtingo diegiant

Reikalavimai

[+] Centralizuotas

[+] Restart'o atveju neprarasti bent dalies "cache"

[+] Minimalus laikas apšilimui po restart'o

[+] Patikimas

[+] Lengvai diegiamas (RHEL4)

[+] "Failover" galimybė

Pasirinkau Redis

Lengvai diegiamas

Lengvai konfiguruojamas

Turi kelis PHP “extension” ir kelis “vanilla php” klases

Greitas – duomenys saugomi RAM

Duomenų kopija periodiškai saugoma diske

Palaiko lengvai diegiamą replikaciją

Tai kas yra ten tokio unikalaus?

Kokias duomenų struktūras palaiko?

strings

hashes

lists

sets

sorted sets with range queries

bitmaps

hyperloglogs

geospatial indexes with radius queries

Papildomos galimybės

Replication

High availability via Redis Sentinel

Lua scripting

LRU eviction

Automatic partitioning with Redis
Cluster

Transactions

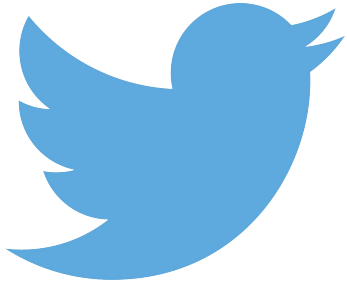
On-disk persistence

Informacijos šaltiniai

- <http://redis.io/>
- <https://twitter.com/redisfeed>
- <https://twitter.com/antirez>
- <https://github.com/antirez/redis>
- <https://groups.google.com/forum/?fromgroups%23!forum/redis-db#!forum/redis-db>

Truputi faktų

- Pradėtas kurti 2009
- Redis – REmote DIctionary Server
- Išaugo iš “logų analitikos” LLOOGG
- Gali palaikyti iki 250 milijonų raktų
- 1 Milijonas “Hello World” užims 100MB atminties (32bit)
- String maksimalus ilgis 512MB



<http://techstacks.io/tech/redis>



Node.js

Java

Lua

ActionScript

C

C+

Go D

PHP

Haskell Erlang

Dart +

Python

C#

Perl

Ruby

Redis funkcionalumo pristatymas

Key-value komandos

- SET key value
- GET key
- DEL key
- EXPIRE key seconds
- RENAME key newkey

Key-value komandos

- SETEX key seconds value
- SETNX key value
- MGET key key key ...
- MSET key value key value

Key-value su skaičiais

- INCR key
- DECR key
- INCRBY key increment
- DECRBY key decrement

Kam galima panaudoti?

Tipinis “Cache” - rezultatams, puslapiams.

Distributed Locking - kai keli procesai turi sinchronizuoti savo veiksmus.

Rate Limiting - kai reiki apriboti API kvietimus, puslapių atidarymus.

Symfony 2 gali naudoti sesijų saugojimui tarp kelių serverių.

Doctrine ORM gali naudoti Metadata, Query, Result kešavimui.

“Hash”ai

- HSET key field value
- HGET key field
- HDEL key field
- HGETALL key
- HVALS key
- HKEYS key

“Hash”ų pavyzdys

- HSET myhash field1 “Value1”
- HSET myhash field2 “Value2”
- HGETALL myhash

```
field1  
Value1  
field2  
Value2
```

```
$var = [  
    'field1' => 'Value1',  
    'field2' => 'Value2',  
];
```

Kam galima panaudoti?

Objektų “Cache” kur:

- atskirus objektų laukus tenka dažniau atnaujinti/apskaičiuoti.
- skirtingiems tinklapio dalim reikia atskirų laukų

Sąrašai / List

- LPUSH arba RPUSH key value
- LPUSHX arba RPUSHX key value
- LPOP arba RPOP key
- LINDEX key index
- LLEN key
- LRANGE key start stop

Sąrašai / List

- LREM key count value
- LSET key index value
- LTRIM key start stop

Kam galima panaudoti?

Eilių organizavimas - bet kokia FIFO ar LIFO eilė.

- Darbų eilė
- Paskutiniai sukurti straipsniai, parašyti komentarai, padaryti užsakymai, užsiregistravę vartotojai

Rinkiniai / Set

- SADD key member
- SREM key member
- SCARD key
- SISMEMBER key member
- SMEMBERS key

Rinkiniai / Set

- SPOP key
- SRANDMEMBER key
- SDIFF key key ...
- SINTER key key ...
- SUNION key key ...

Kam galima panaudoti?

Unikalių duomenų saugojimas ir operacijos

- Pasirinkti atsitiktinai narį iš sąrašo

Rikiuoti rinkiniai / Sorted Set

- ZADD key score member
- ZREM key member
- ZCARD key
- ZCOUNT key min max
- ZINCRBY key increment member
- ZRANK key member
- ZREVRANK key member

Rikiuoti rinkiniai / Sorted Set

- ZRANGE key start stop
- ZREVRANGE key start stop
- ZRANGEBYSCORE key max min
- ZREVRANGEBYSCORE key max min

Kam galima panaudoti?

Nuolat atnaujinami TOP sąrašai

- TOP žaidėjai pagal rezultata
- TOP žaidėjai pagal padaryta žalą
- TOP tinklapio straipsniai pagal komentarus

Pub/Sub

- PUBLISH channel message
- SUBSCRIBE channel
- UNSUBSCRIBE channel

- PSUBSCRIBE pattern
- PUNSUBSCRIBE pattern

Kam galima panaudoti?

Realaus laiko komunikacija

- Chat
- Fanout job queue
- Node.JS + Socket.IO procesų sinchronizavimas
- Redis Sentinel naudoja vidiniai komunikacijai

Transakcijos!?

- MULTI
- ...
- ...
- EXEC
- DISCARD

Dar komandos

- SELECT index
- FLUSHALL
- FLUSHDB
- SAVE ir BGSAVE
- INFO
- MONITOR

Replikacija

- Konfigūracinė rinkmena
- Automatiškai kopijuoja duomenis
- Automatiškai sinchronizuoja duomenis po prisijugimo

- `SLAVEOF 192.168.1.1 6379`
- `CONFIG set masterauth <master password>`

Ko dar neteko panaudoti

Redis Sentinel

High-Availability sprendimas

Labai įdomus sprendimas, kai yra atskiri “stebėtojai” vadinami Sentinel.

Atskiri “procesai” kurie stebi Redis procesus, automatiškai konfiguruoja replikacija.

Atsirado nuo 2.4 versijos.

Redis Cluster

Automated Sharding

Automatiškai paskirsto įrašus pagal raktą ir užtikrina High-Availability.

Turi įdomių apribojimų ir reikalavimų.

Atsirado nuo 3.0 versijos.

Nusipelno atskiros temos.

Atsiliepimas apie Redis iš mano praktikos

Roman Getman: идею с редисом ты придумал вроде, да?

Roman Getman: в общем за это тебе спасибо 😊 редис супер!

Roman Getman: panaudoti Redis tu sugalvojai, taip?

Roman Getman: ačiū tau 😊 Redis super!

Laikas diskusijai

Sergej Kurakin

Darbinis paštas: sergej.kurakin@nfq.lt

Asmeninis paštas: sergej@kurakin.info

<https://www.linkedin.com/in/sergejkurakin>

