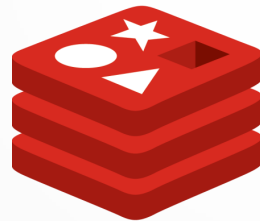


redis

Róbert Móra



nosql-database.org



redis



Redis is an open source,
advanced **key-value store**. It is often
referred to as a **data structure
server** since keys can contain
*strings, hashes, lists, sets and sorted
sets.*

Čo to je?

- Úložisko typu **klíč-hodnota**

Klíč (String)	Hodnota (String, Set, ...)
image:redis.png	iVBORw0KGgoAAAAN...
login_count	4247
logged_in:2012-04-30	{ 1, 17, 33, 42, 127, 512 }
latest_post_ids	[105, 109, 121, ...]
user:1234:session	{ login: xmoror, aiscid: 47879 }
best_scores	{ (134.12, 1), (11.31, 103), ... }

Rovno si to vyskúšajme!

<http://try.redis-db.com>

* TRY REDIS *

<http://redis.io/commands>

Ako si to nainštalujem?

- Linux:

```
$ wget http://redis.googlecode.com/files/redis-2.6.12.tar.gz
```

```
$ tar xzvf redis-2.6.12.tar.gz
```

```
$ cd redis-2.6.12
```

```
$ make
```

```
$ make install
```

- Windows

- pozri odkaz na konci prezentácie

Ako to spustím?

- Spustenie Redis serveru
\$ redis-server
- Spustenie konzoly na interakciu s Redisom
\$ redis-cli

Kľúč

- „binary-safe“
 - Môže to byť akákoľvek sekvencia bajtov
- Nepoužívať príliš dlhé kľúče (> 1KB)
- Zbytočne nepoužívať príliš krátke kľúče, dobré riadiť sa konvenciou

`object-type:id:field`

- DEL, TYPE, ...

Reťazec (String)

- „binary-safe“ (JPEG obrázkov, serializovaný objekt)
- Max. 512MB
- INCR, DECR, APPEND, SET, GET, GETSET, SETBIT, ...
 - > SET counter 1
 - > INCR counter
 - > INCRBY counter 2
 - > GET counter

Zoznam (List)

- Zoznam reťazcov (max. $2^{32} - 1$ prvkov)
- Usporiadané v poradí, v akom boli vložené
- LPUSH, RPUSH, LLEN, LRANGE ...
 - > RPUSH queue 1
 - > RPUSH queue 2
 - > LPUSH queue 3
 - > LRANGE queue 0 -1

Množina (Set)

- Neusporiadaná kolekcia unikátnych reťazcov
- Pridávanie, odoberanie prvkov a testovanie ich existencie v konštantnom čase $O(1)$
- SADD, UNION, SINTER, SDIFF, SCARD, ...
 - > SADD dbs jozko petko vladko
 - > SADD psi katka palko petko
 - > UNION dbs psi
 - > SINTER dbs psi
 - > SDIFF dbs psi

Hash

- Mapa dvojíc kľúč-hodnota (oba typu reťazec)
- Vhodné na reprezentáciu objektov
- HGETALL, HSET, HMSET, HKEYS, HVALS, ...

> HMSET user:1 login xmoror psswd heslo age 23

> HSET user:1 psswd sezamOtvor_sa

> HGETALL user:1

Usporiadaná množina (Sorted Set)

- Prvky množiny sú usporiadané podľa skóre
- Hodnoty skóre sa môžu opakovať
- Náhrada indexov, vhodné tiež pre štatistiky
- ZRANGE, ZRANGEBYSCORE, ZCOUNT, ZRANK, ...
 - > ZADD page_views 500 a 2000 b 5000 c 1200 d
 - > ZREVRANGE page_views 0 -1
 - > ZRANGEBYSCORE page_views 1000 2000 WITHSCORES

Kde sa uchovávajú dáta?

- Dáta uchovávané **v pamäti** (RAM)
 - Veľmi rýchle (redis-benchmark)
- Umožňuje však perzistenciu
 - **RDB (snapshots)**: asynchrónny zápis celého obsahu DB na disk v definovaných intervaloch
`save <seconds> <changes>`
 - **AOF (append-only log file)**: zaznamenáva každú operáciu zápisu => je možné zrekonštruovať obsah DB

Ako to funguje? (1/2)

- **Atomické** vykonávanie operácií
 - V jednom vlákne
- Podpora **transakcií**
 - Všetky príkazy vykonané ako jedna atomická operácia
 - Začiatok transakcie: `MULTI`
 - Commit: `EXEC`
 - “Rollback”: `DISCARD`

Ako to funguje? (2/2)

- Klient-server architektúra
 - Komunikácia Redis protokolom nad TCP
- Master-slave replikácia
 - Na oboch stranách neblokujúca
 - Škálovateľnosť: urýchlíuje operácie čítania

```
slaveof <hostname> <port>
```


Čo mi to ešte poskytuje? (1/3)

- **Expirácia kľúčov**
 - Výhodné pri využití ako session cache
 - EXPIRE, TTL, SETEX
- Implementácia vzoru **Vydavateľ – Odberatelia** (Publisher/Subscriber)
 - Vydavateľ zapisuje správy do kanála
 - Odberatelia sa registrujú na kanál a dostávajú správy
 - Vhodné napr. pri implementácii chatu
 - SUBSCRIBE, UNSUBSCRIBE, PUBLISH

Čo mi to ešte poskytuje? (2/3)

- **Skriptovanie v jazyku Lua**
 - Od verzie 2.6
 - Vykonávané atomicky
 - EVAL, EVALSHA
 - Vyvolanie príkazov Redisu: `redis.call`
 - Musí byť čistá funkcia
 - Pre danú kombináciu vstupných argumentov vždy vráti rovnaký výsledok
 - Nezávisí od vonkajšieho stavu

Čo mi to ešte poskytuje? (3/3)

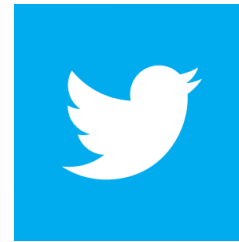
- **SORT**
 - Usporiadúva zoznam, množinu a usporiadanú množinu
 - Čísla aj reťazce (lexikograficky)
 - Dá sa použiť ako **EQUI-JOIN**

`SORT mylist BY nosort GET object_*`

Kde všade to viem použiť?

- Podporované jazyky
 - Ruby, Python, Java, C, C++, C#, PHP, Javascript (Node.js), Objective-C, Scala, ...
- Kde sa používa

github



flickr®
from YAHOO!



 stackoverflow



Ako sme použili Redis v ALEFe?

- **Session cache**
 - Uchovávanie stavu widgetov
 - Predchádzanie „race conditions“ pri asynchrónnej inicializácii widgetov
- **Spracovanie úloh na pozadí**
 - Logovanie, prepočítavanie skóre, aktualizácia modelu používateľa, príprava sumarizácií, ...
 - resque + resque-pool

Ako sme použili Redis v ALEFe?

- Live demo

Ďalšie možnosti použitia?

- Štatistiky
- Progress bar / polling
- Tagovanie
- Chat, komentáre, posielanie správ (MQ)
 - Resque, RestMQ
- Prúdy aktivít (Facebook, Twitter)
- Autocomplete
- Full-textové vyhľadávanie

Kedy použiť Redis?

- Keď je kritická rýchlosť
- Potrebujeme škálovať
 - Redis cluster
- Dáta sa zmestia do RAM
 - Neplatí, ak použijeme v kombinácii s klasickou relačnou DB

Kam d'alej? (1/3)

- <http://redis.io/>
- Tutoriály
 - <http://try.redis-db.com/>
 - <http://no.gd/redis-presentation.pdf>
- Knihy
 - <http://openmymind.net/2012/1/23/The-Little-Redis-Book/>

Kam ďalej? (2/3)

- Užitočné gemy
 - <https://github.com/redis/redis-rb>
 - <https://github.com/resque/resque>
 - <https://github.com/nevans/resque-pool>
- Redis na MS Windows
 - <https://github.com/MSSOpenTech/Redis>
- Porovnanie NoSQL databáz
 - <http://kkovacs.eu/cassandra-vs-mongodb-vs-couchdb-vs-redis>

Kam ďalej? (3/3)

- Pre tých, čo chcú vedieť viac
 - <http://antirez.com/>
 - <http://antirez.com/post/autocomplete-with-redis.html>
 - <http://redis.io/topics/twitter-clone>
 - <http://www.enjoythearchitecture.com/redis-architecture>
 - <http://blog.getspool.com/2011/11/29/fast-easy-realtime-metrics-using-redis-bitmaps/>
 - [http://redis.io/presentation/Redis Cluster.pdf](http://redis.io/presentation/Redis%20Cluster.pdf)
 - <http://restmq.com/>
 - <http://www.infoq.com/presentations/Real-Time-Delivery-Twitter>
 - <http://highscalability.com/blog/2012/4/9/the-instagram-architecture-facebook-bought-for-a-cool-billio.html>