REDIS - HOW TO PROFIT FROM ADDING IT TO YOUR STACK

ABOUT ME

Milan Heimschild
github.com/mheimschild
@mheimschild

WHAT IS REDIS?

- in memory data store
- high performance
- publish/subscribe
- replication
- persistence
- no need for switch, just add it to your stack

WHEN TO USE REDIS

- performance
- write-heavy app
- lot of changes
- data fits natural Redis structures

WHEN NOT TO USE REDIS

- You need ACID
- Complex data structures

INSTALLATION

- Windows - Port or Cygwin
- Linux
- OSX
- Docker

REDIS CLIENTS

- http://redis.io/clients
- Java - Jedis/lettuce
- Spring Data Redis

JAVA PROJECT - POM.XML

```
<dependency>
    <groupId>org.springframework.data</groupId>
    <artifactId>spring-data-redis</artifactId>
    <version>1.6.4.RELEASE</version>
</dependency>

<dependency>
    <groupId>redis.clients</groupId>
    <artifactId>jedis</artifactId>
    <version>2.8.0</version>
</dependency>
```

JAVA PROJECT - SPRING CONTEXT

```
<bean id="connectionFactory" class="...connection.jedis.JedisConnecti
    <property name="usePool" value="true"/>
</bean>

<bean id="redisTemplate" class="...redis.core.RedisTemplate">
    <property name="connectionFactory" ref="connectionFactory"/>
</bean>
```

DATA STRUCTURES

- Strings
- Lists
- Sets
- Hashes
- Sorted Sets

STRINGS

- Strings
- Integers
- Floats
- Bitmaps
- Atomic multiples

STRINGS - EXAMPLES

```
SET name RedisTalk
GET name
# RedisTalk
```

```
SET counter 1
INCR counter
GET counter
# 2
```

```
GETSET counter 3
# 2
```

```
SETNX counter 4
GET counter
# 3
```

```
SETBIT flags 0 1
GETBIT flags 0
# 1
```

SETS

- unsorted collections of strings
- add/remove
- membership
- union/intersection/diff

SETS - EXAMPLES

```
SADD products iPad Nexus
SMEMBERS products
# iPad Nexus
SISMEMBER products Nexus
# 1
```

```
SADD offers iPad Galaxy
SINTER offers products
# iPad
```

```
SUNION products offers
# iPad Nexus Galaxy
```

SETS

- good for:
  - collections
  - verifying existence
- complexity O(1)

SORTED SETS

- same as sets but with order
- add/fetch/remove
- scoring
- rank

SORTED SETS - EXAMPLES

```
ZADD access:hours 1457628349333 1500
ZADD access:hours 1457624749333 800
# ZINCRBY access:hours 1457624749333 1
ZADD access:hours 1457621149333 1200
ZREVRANGE access:hours
# 1500 800 1200
```

```
ZREVRANGEBYSCORE access:hours inf 1457624749333
# 1500 800
```

```
ZREVRANGE access:hours 0 0
# 1500
```

SORTED SETS

- good for:
  - leaderboards
  - timestamp data ranges
  - autocomplete
- complexity O(log(N))

LISTS

- Linked list
- push/pop
- search
- remove

LISTS - EXAMPLES

```
LPUSH stack 1
LPUSH stack 2
LPUSH stack 3
LPOP stack
# 3
```

```
RPOP stack
# 1
```

LISTS

- good for:
  - stacks
  - queues
  - last updated
  - sidekiq
- complexity O(1)
- but O(n) for inserting

HASHES

- add
- fetch
- remove
- complex structures

HASHES - EXAMPLES

```
HMSET user:98765 name "Milan Heimschild" logins 0
HINCRBY user:98765 logins 1
```

```
HGET user:98765 logins
# 1
```

```
HGETALL user:98765
# "name" "Milan Heimschild"
# "logins" "1"
```

HASHES

- good for:
  - representing objects
  - storing objects
  - storing objects references

HYPERLOGLOGS

- computes cardinality of a set

EXAMPLE

```
PFADD hll user:1 user:2 user:3
PFCOUNT hll
# 3
```

PUBLISH/SUBSCRIBE

- ! In Memory
- Reliability

EXAMPLE

```
redisTemplate.convertAndSend("chat", "Hello All!");

redisConnectionFactory.getConnection()
    .subscribe((message, bytes) -> {
    sout(valueSerializer.deserialize(message.getBody()));
    sout(stringSerializes.deserialize(message.getChannel()));
}
```

EXAMPLE - FIXED

```
ExecutorService exServ = Executors.newFixedThreadPool(1);

exServ.submit(() -> redisConnectionFactory.getConnection()
    .subscribe((message, bytes) -> {
    sout(valueSerializer.deserialize(message.getBody()));
    sout(stringSerializes.deserialize(message.getChannel()));

    redisConnectionFactory.getConnection().getSubscription().unsubscr
}, "chat".getBytes()));
```

EXPIRING KEYS

- Good for volatile keys
- sessions/caching/quotas
- EXPIRE
- PERSIST
- TTL

EXAMPLE

```
SETEX myValue 1 42
GET myValue
# 42
# delay
GET myValue nil
```

```
SETEX myValue 60 42
TTL myValue
# 59
PERSIST myValue
TTL myValue
# -1
GET myValue
# 42
```

TRANSACTIONS

- MULTI/EXEC
- DISCARD
- WATCH/MULTI/EXEC
- Errors during transactions

EXAMPLE

```
HMSET item:42 desc "Nexus" count 10

HGET item:42 count
# 10
MULTI
HINCRBY item:42 count -1
# QUEUED
LPUSH cart:123 "item:42"
# QUEUED
EXEC
#9
```

EXAMPLE - MORE TRANSACTIONAL

```
HMSET item:42 desc "Nexus" count 10

HGET item:42 count
# 10
WATCH item:42

MULTI
HINCRBY item:42 count -1
# QUEUED
LPUSH cart:123 "item:42"
# QUEUED

# in another client
HINCRBY item:42 count -1
# 9
```

PERSISTENCE

- Snapshot
  - manually/automatically
  - frontend/backend
  - 200ms/1 GB
- AOF
  - transaction log
  - best in RAID
  - system FS

REPLICATION

- Master/Slave
- Sentinels
- Cluster

BENCHMARKS

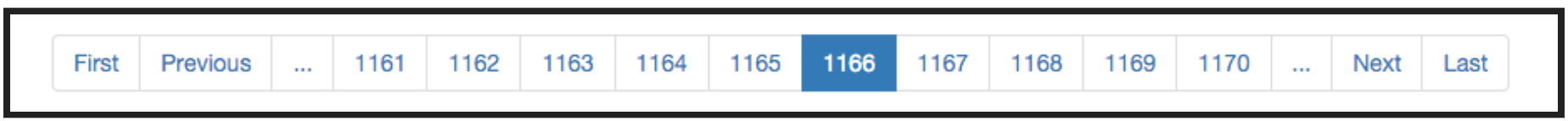
- redis-benchmark

SECURITY

- requirepass config
- AUTH password
- Must be really strong
- Proxy

ADVANCED EXAMPLES

PAGINATION (OR N LATEST ELEMENTS)



- Long lists
- What for? (SCO)
- Why not DB

SOLUTION #1 - LISTS

```
LPUSH lastcomments 1 2 3 4 5
LLEN lastcomments
# 5
LRANGE lastcomments 0 -1
# 1 2 3 4 5
```

```
LPUSH lastcomments 6
LTRIM 0 4
LRANGE lastcomments 0 -1
# 2 3 4 5 6
```

```
LRANGE comments 11 20
```

SOLUTION #2 - SORTED SETS

```
ZADD topcomments 10 1
ZADD topcomments 5 2
ZADD topcomments 15 3
ZREVRANGE topcomments 0 -1
# 3 1 2
```

```
ZADD topcomments 7 4
ZREMRANGEBYRANK topcomments 0 -4
ZREVRANGE topcomments 0 -1
# 3 1 4
```

CACHING

- In-memory caching
- Redis vs. Memcached
- NGINX redis adapter
- Redis via unix sockets
- maxmemory options
- can grow too fast

SOLUTION

```
SET page42 '<div>42</div>'
EXPIRE page42 300
```

```
SETEX page42 300 '<div>42</div>'
```

```
SET object:1234 "some value" # wrong
HSET object 1234 "some value" # right
```

PRODUCTS CATALOG

- Product attributes
- List of products
- Searching

SOLUTION

```
HMSET phone:1234567 company "LG" model "Nexus" price 300

HMSET phone:search "Nexus" 1234567

HSCAN phone:search 0 MATCH "*Nex*" COUNT 5
# Next 5 entries
HSCAN phone:search 5 MATCH "*Nex*" COUNT 5
```

AUTOCOMPLETE

- use ElasticSearch
- IP-to-city
- GEO-to-city

SOLUTION

```
ZADD autocomplete 0 m
ZADD autocomplete 0 mi
ZADD autocomplete 0 mil
ZADD autocomplete 0 mila
ZADD autocomplete 0 milan
ZADD autocomplete 0 milan$
```

```
ZRANK autocomplete mil
# 2
```

```
ZRANGE autocomplete 3 50
#mila milan milan$
```

```
ZRANGEBYLEX autocomplete [mi (miz
```

SESSION MANAGEMENT

- does not require stickiness
- faster than DB
- stabler than Memcached

SOLUTION

```
HMSET session:42 username "milan" locale "de"
EXPIRE session:42 3600
```

LEADERBOARD

SOLUTION

```
ZADD points 200 Milan 300 Sigi
ZREVRANGE points 0 -1
# "Sigi" "Milan"
```

```
ZADD stars 5 Milan 2 Sigi
ZREVRANGE stars 0 -1
# "Milan" "Sigi"
```

```
ZUNIONSTORE leaderboard 2 points stars WEIGHTS 1 100
ZREVRANGE leaderboard 0 -1 WITHSCORES
# "Milan" "700"
# "Sigi" "500"
```

NOTIFICATION CENTER

- PUB/SUB not reliable

SOLUTION

- Publish/Subscribe
- Retrieve N latest entries

COMMENTS

- Nonthreaded

SOLUTION

```
LPUSH article:42:comments comment:12
HMSET comment:12 author "Milan Heimschild" text "Awesome comment"
timestamp "2014-05-17 23:00:34"
```

```
# Deleting
DEL comment:12
LREM article:42:comments 0 comment:12
```

```
# Listing
LRANGE article:42:comments 0 10
# comment:12
HGETALL comment:12
# author: ... text ... timestamp
```

SHOPPING CART

- Product catalog
- Transactions
- Publish/Subscribe
- Key-Space notification

SOLUTION

```
HMSET product:1 desc "iPad" price 500 count 10
HMSET product:2 desc "Nexus" price 300 count 20
```

```
MULTI
HGET product:1 count
# 10
HINCRBY product:1 count -1
# 9
RPUSH cart:42 product:1
EXPIRE cart:42 600
EXEC
```

```
PSUBSCRIBE __keyspace@0__:cart* del
```

TIPS AND TRICKS

- (hash|list|zset|set)-max-zip(map|list)-(entries|value) - factor 10
- Pipelining - factor 5
- Lua Scripting
- Big Data Import (use protocol RESP)
- Data partitioning (key based)

RESOURCES

- http//redis.io/commands
- http://redis.io/clients#java
- http://github.com/mheimschild/redis-talk