
redis-rb-examples Documentation

Release 0.1.2

Vladimir Botka

Sep 17, 2019

Contents:

1	Introduction	1
2	Examples	3
2.1	Create frequency graph from a log	3
2.2	List 10 most used words in a text	5
3	Indices and tables	7

CHAPTER 1

Introduction

Redis is popular in-memory data structure store, used as a database, cache and message broker. [redis-rb](#) is Ruby client library for Redis documented at [rubydoc](#) .

Intention of this documentation is to create examples how to use redis-rb.

2.1 Create frequency graph from a log

Task: Read `/var/log/dpkg.log` and create a graph to visualize how often packages are installed, upgraded and removed.

Solution: The loop (34) calls function `read_log` which reads the log line by line (13), splits the fields (14) and concatenate date `l[0]` and time `l[1]` in minutes (15). Third field of the log `l[2]` is status of the dpkg operation (install, upgrade, remove ...). Method `zincrby` (16) increments by 1 the score of `word` in the key `l[2]`. As a result the database contains keys (install, upgrade, remove ...) and associated lists of `words` sorted by score. Next loop (38) calls the function `write_csv` with all keys. As a result `status.csv` files are created in the current directory with the `(date;score)` pairs.

[create-graph-01.rb]

```
1  #!/usr/bin/ruby
2  # Tested with ruby 2.3.3, ruby-redis 3.3.3 and redis 4.0.1
3
4  require 'redis'
5
6  LOG_FILES = ['/var/log/dpkg.log', ]
7  LOG_SEPARATOR = ' '
8  CSV_SEPARATOR = ';'
9
10 def read_log(log_file, r)
11   # This function reads log_file and put the status into the database
12   f = File.open(log_file, "r")
13   f.each do |line|
14     l = line.split(LOG_SEPARATOR)
15     word = l[0] + " " + l[1][0..-4]
16     r.zincrby(l[2], 1, word)
17   end
18   f.close
19 end
20
```

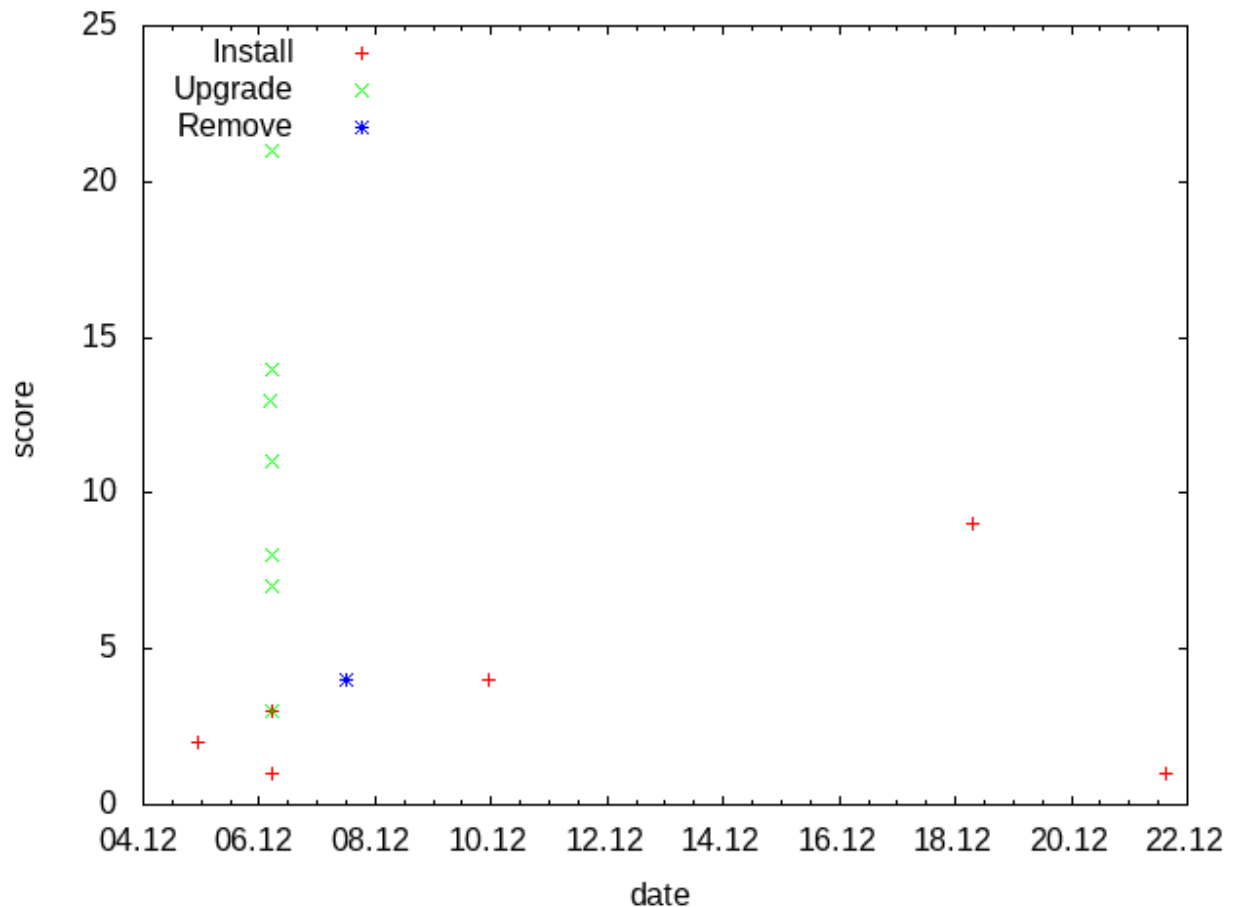
(continues on next page)

(continued from previous page)

```
21 def write_csv(status, r)
22   # This function reads the database and writes the status CSV file
23   f = File.open(status + ".csv", "w")
24   l = r.zrange(status, 0, -1, :with_scores => true)
25   l.each do |x|
26     f.write( x[0] + CSV_SEPARATOR + x[1].to_s + "\n")
27   end
28   f.close
29 end
30
31 r = Redis.new(host: "localhost", port: 6379, db: 0)
32 r.flushdb
33
34 for log_file in LOG_FILES
35   read_log(log_file, r)
36 end
37
38 for status in r.keys
39   write_csv(status, r)
40 end
```

Result: The *status.csv* files can be used to create a graph with *gnuplot*.

[create-graph-01.gnuplot]



2.2 List 10 most used words in a text

Task: Read text from a file and list 10 most frequently used words in it.

Solution: Let's use article about Redis at wikipedia.org as a text.

[create-topchart-text.bash]

```
#!/bin/bash
lynx -dump -nolist https://en.wikipedia.org/wiki/Redis > redis.txt
```

`zincrby` (12) increments by 1 the score of *word* in the key *topchart* and `zrange` (14) returns top 10 words with scores.

[create-topchart.rb]

```
1  #!/usr/bin/ruby
2  # Tested with ruby 2.3.3, ruby-redis 3.3.3 and redis 4.0.1
3
4  require 'redis'
5
6  FILE = 'redis.txt'
7
8  r = Redis.new(host: "localhost", port: 6379, db: 0)
9  r.flushdb
10
11 f = File.open(FILE, "r")
12 f.read.scan(/\w+/).each { |word| r.zincrby("topchart", 1, word) }
13
14 ranking = r.zrange("topchart", -10, -1, :with_scores => true)
15
16 for x in ranking
17   puts ( x[1].to_s + " " + x[0] )
18 end
```

Result:

```
> ./create-topchart.rb
11.0 Retrieved
13.0 edit
23.0 in
24.0 a
24.0 is
26.0 and
33.0 of
34.0 to
37.0 the
69.0 Redis
```


CHAPTER 3

Indices and tables

- `genindex`
- `modindex`
- `search`