

redis

# Introduction

HumanTalks Nantes - 13 Mai 2014

Francois-Guillaume Ribreau



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

REmote DIctionary Server (Redis) is created by @antirez, written in C, released in 2009, BSD licensed, was supported by VMWare and now Pivotal.

# Key features



- Atomic operations
- Lua Scripting
- Pub/Sub
- Transactions
- Master/Slave replication
- Cluster (with automatic sharding)\*
- Automatic failover (Redis Sentinel)
- Append Only File (AOF) persistence
- Snapshot (RDB file) persistence

# Key features



- Atomic operations
- Lua Scripting
- Pub/Sub
- Transactions
- Master/Slave replication
- Cluster (with automatic sharding)\*
- Automatic failover (Redis Sentinel)
- Append Only File (AOF) persistence
- Snapshot (RDB file) persistence

\* currently in 3.0.0 beta-3

# Key advantages



- › Blazing\* Fast
- › Robust
- › Easy to setup, use and maintain
- › Extensible with LUA scripting

# Key advantages



- › Blazing\* Fast
- › Robust
- › Easy to setup, use and maintain
- › Extensible with LUA scripting

\* as the cool kids say these days.

# Key disadvantages



- › Persistence consumes lot of I/O when using RDB
- › all your data must fit in memory

# Key disadvantages



**all** your data **must**  
**fit** in **memory**



**again.**

all your data

must fit in

memory

# What for ?



- Cache out-of-process
- Duplicate detector
- LIFO/FIFO Queues\*, Priority Queue
- Distributed HashMap
- UID Generator
- Pub/Sub
- Real-time analytics & chat apps
- Counting Stuff
- Metrics DB
- Implement expires on items
- Leaderboards (game high scores)
- Geolocation lookup
- API throttling (rate-limits)
- Autocomplete
- Social activity feed
- ...



## Keys

cache:/

users:l

users:l:followers

users:l:tweets

users:mostfollowed



## Keys

## Value Type

cache:/

String

users:l

Hash

users:l:followers

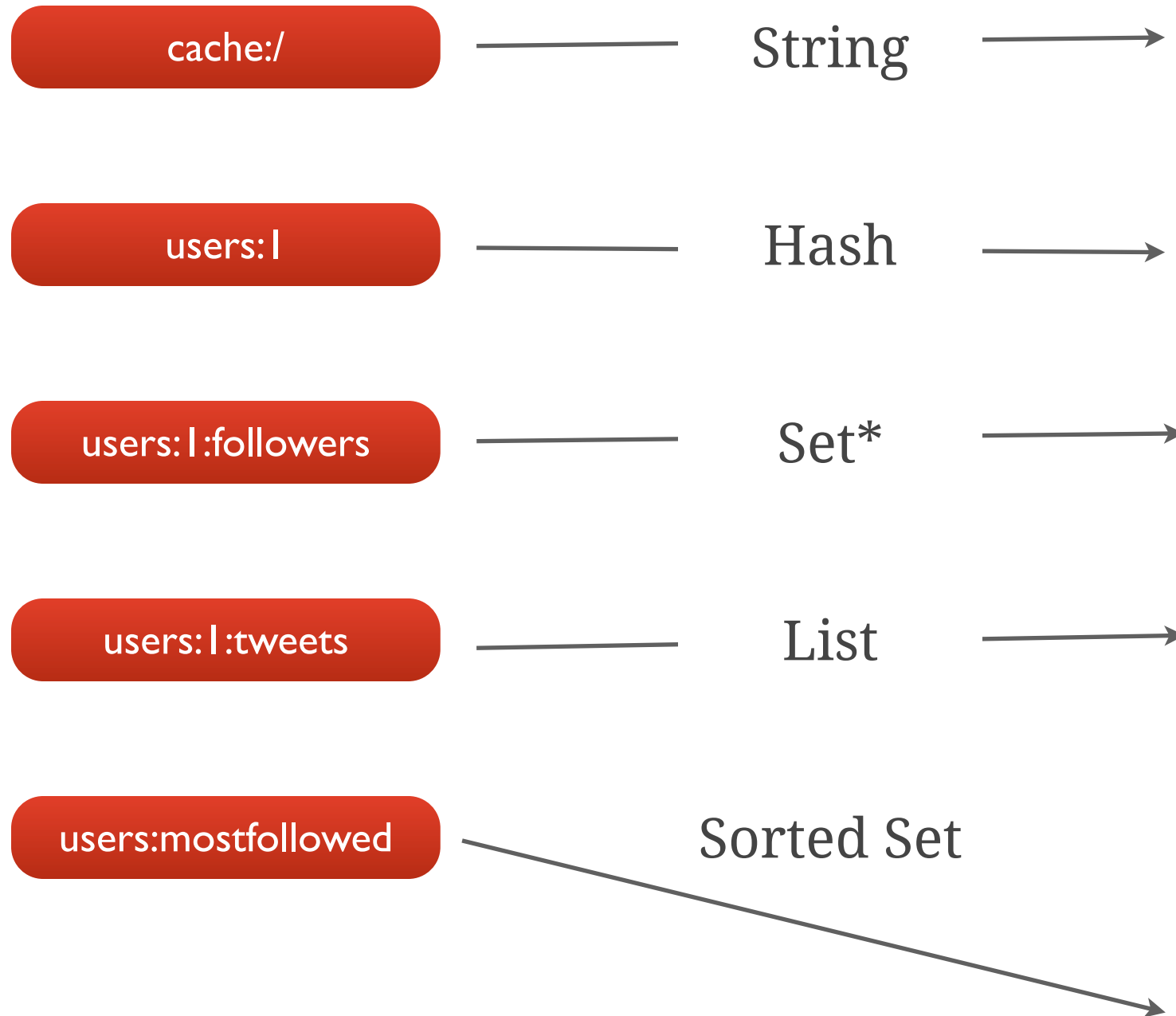
Set\*

users:l:tweets

List

users:mostfollowed

Sorted Set





## Keys

## Value Type

## Example

cache:/

String

<!DOCTYPE html><html><head><meta ...

users:l

Hash

<i>field</i>	<i>member</i>
username	fgribreau
followers	2204

users:l:followers

Set\*

89	8	55	5	34	13	21
----	---	----	---	----	----	----

users:l:tweets

List

0	1	2	3
hello...	hey!..	@Crea...	New pro...

users:mostfollowed

Sorted Set

<i>field</i>	<i>score</i>
@ladygaga	41359356
@youtube	41770322
@barackobama	42872391
@justinbieber	51428544
@katyperry	52811148

\* Since 2.8.9, Redis supports a new type: HyperLogLog (an memory efficient way to store and count unique elements in a set)



## Keys

## Value Type

## Example

cache:/

String

<!DOCTYPE html><html><head><meta ...

```
$ redis-cli
redis> get talk:414:description
(nil)
redis> set talk:414:description "Je propose ce sujet suite à une suggestion de Quentin."
OK
redis> get talk:414:description
"Je propose ce sujet suite \xc3\xa0 une suggestion de Quentin."

redis> expire talk:414:description 5
(integer) 1
# then after 5 seconds
redis> get talk:414:description
(nil)

redis> getrange talk:414:description 47 53
"Quentin"

redis> incr talk:414:views
(integer) 1
redis> incrby talk:414:views 10
"11"
```

<http://bit.ly/1kRJn0N>

append bitcount bitop bitpos decr decrby get getbit getrange getset incr incrby incrbyfloat mget  
mset msetnx psetex set setbit setex setnx setrange strlen



## Keys

## Value Type

## Example

users:1

Hash

<i>field</i>	<i>member</i>
username	fgribreau
followers	2204

```
redis> hmset users:1 username fgribreau followers 2204
OK
```

```
redis> hgetall users:1
1) "username"
2) "fgribreau"
3) "followers"
4) "2204"
```

```
redis> hincrby users:1 followers 1
(integer) 2205
```

```
redis> hget users:1 followers
"2205"
```

```
redis> hset users:1 username humantalks
(integer) 0
```

hdel hexists hget hgetall hincrby hincrbyfloat hkeys hlen hmget hmset hset hsetnx hvals hscan





Keys

Value Type

Example

users:1:followers



```
redis> sadd users:1:followers 89 8 55 5 34 13 21
(integer) 7
```

```
redis> sadd users:2:followers 0 1 1 2 3 5 8 13 21
(integer) 8
```

```
redis> sinter users:1:followers users:2:followers
1) "5"
2) "8"
3) "13"
4) "21"
```

```
redis> scard users:1:followers
(integer) 7
```

sadd scard sdiff sdiffstore sinter sinterstore sismember smembers smove spop srandmember srem  
union unionstore sscan



Keys

Value Type

Example

users:1:followers

Set

89

8

55

5

34

13

21

```
redis> sadd users:1:followers 89 8 55 5 34 13 21
(integer) 7
```

```
redis> sadd users:2:followers 0 1 1 2 3 5 8 13 21
(integer) 8
```

```
redis> sinter users:1:followers users:2:followers
1) "5"
2) "8"
3) "13"
4) "21"
```

```
redis> scard users:1:followers
(integer) 7
```

sadd scard sdiff sdiffstore sinter sinterstore sismember smembers smove spop srandmember srem  
union unionstore sscan



## Keys

## Value Type

## Example

`users:1:tweets`

```
redis> rpush users:1:tweets hello...  
(integer) 1  
redis> rpush users:1:tweets hey!...  
(integer) 2
```

```
redis> lindex users:1:tweets 1  
"hey!..."
```

```
# Shell 1  
redis> brpop queue 0  
# blocking until
```

```
1) "queue"  
2) "my_task"  
(14.27s)
```

```
# Shell 2  
redis> lpush queue my_task  
(integer) 1
```

`blpop brpop brpoplpush lindex linsert llen lpop lpush lpushx lrange lrem lset ltrim rpop rpoplpush  
rpush rpushx`



## Keys

## Value Type

## Example

`users:1:tweets`

List

0	1	2	3
hello...	hey!...	@Crea...	New pro...

```
redis> rpush users:1:tweets hello...
(integer) 1
redis> rpush users:1:tweets hey!...
(integer) 2
```

```
redis> lindex users:1:tweets 1
"hey!..."
```

```
# Shell 1
redis> brpop queue 0
# blocking until
```

```
1) "queue"
2) "my_task"
(14.27s)
```

```
# Shell 2
redis> lpush queue my_task
(integer) 1
```

blpop brpop brpoplpush lindex linsert llen lpop lpush lpushx lrange lrem lset ltrim rpop rpoplpush  
rpush rpushx



## Keys

## Value Type

## Example

users:byFollowers

```
redis> zadd users:byFollowers 51428544  
@kant 42872391 @einstein # etc. ...  
(integer) 5
```

```
redis> zrevrange users:byFollowers 0 2  
1) "@clesoleil"  
2) "@kant"  
3) "@einstein"
```

```
redis> zrevrange users:byFollowers 0 1 withscores  
1) "@clesoleil"  
2) "52811148"  
3) "@kant"  
4) "51428544"
```

```
redis> zrevrank users:byFollowers @kant  
(integer) 1
```

```
redis> zscore users:byFollowers @kant  
"51428544"
```

```
zadd zcard zcount zincrby zinterstore zlexcount zrange zrangebylex zrangebyscore zrank zrem  
zremrangebylex zremrangebyrank zremrangebyscore zrevrange zrevrangebyscore zrevrank zscore  
zunionstore zscan
```



## Keys

## Value Type

## Example

**users:byFollowers**

Sorted Set

```
redis> zadd users:byFollowers 51428544  
@kant 42872391 @einstein # etc. ...  
(integer) 5
```

```
redis> zrevrange users:byFollowers 0 2  
1) "@clesoleil"  
2) "@kant"  
3) "@einstein"
```

```
redis> zrevrange users:byFollowers 0 1 withscores  
1) "@clesoleil"  
2) "52811148"  
3) "@kant"  
4) "51428544"
```

```
redis> zrevrank users:byFollowers @kant  
(integer) 1
```

```
redis> zscore users:byFollowers @kant  
"51428544"
```

<i>field</i>	<i>score</i>
@echouard	41359356
@tesla	41770322
@einstein	42872391
@kant	51428544
@clesoleil	52811148

```
zadd zcard zcount zincrby zinterstore zlexcount zrange zrangebylex zrangebyscore zrank zrem  
zremrangebylex zremrangebyrank zremrangebyscore zrevrange zrevrangebyscore zrevrank zscore  
zunionstore zscan
```

# Who is using Redis?



<http://bit.ly/lfp4ept>



<http://bit.ly/XX0yHy>



<http://bit.ly/lf0GWez>



<http://bit.ly/lfp4l4t>



<http://bit.ly/lf0HlIU>



<http://bit.ly/lf0GkWi>



<http://bit.ly/lf0Gq06>



<http://bit.ly/lfp4qVl>



<http://bit.ly/lf0GDjW>

<http://bit.ly/lbGQqJf>

# Who is using Redis?



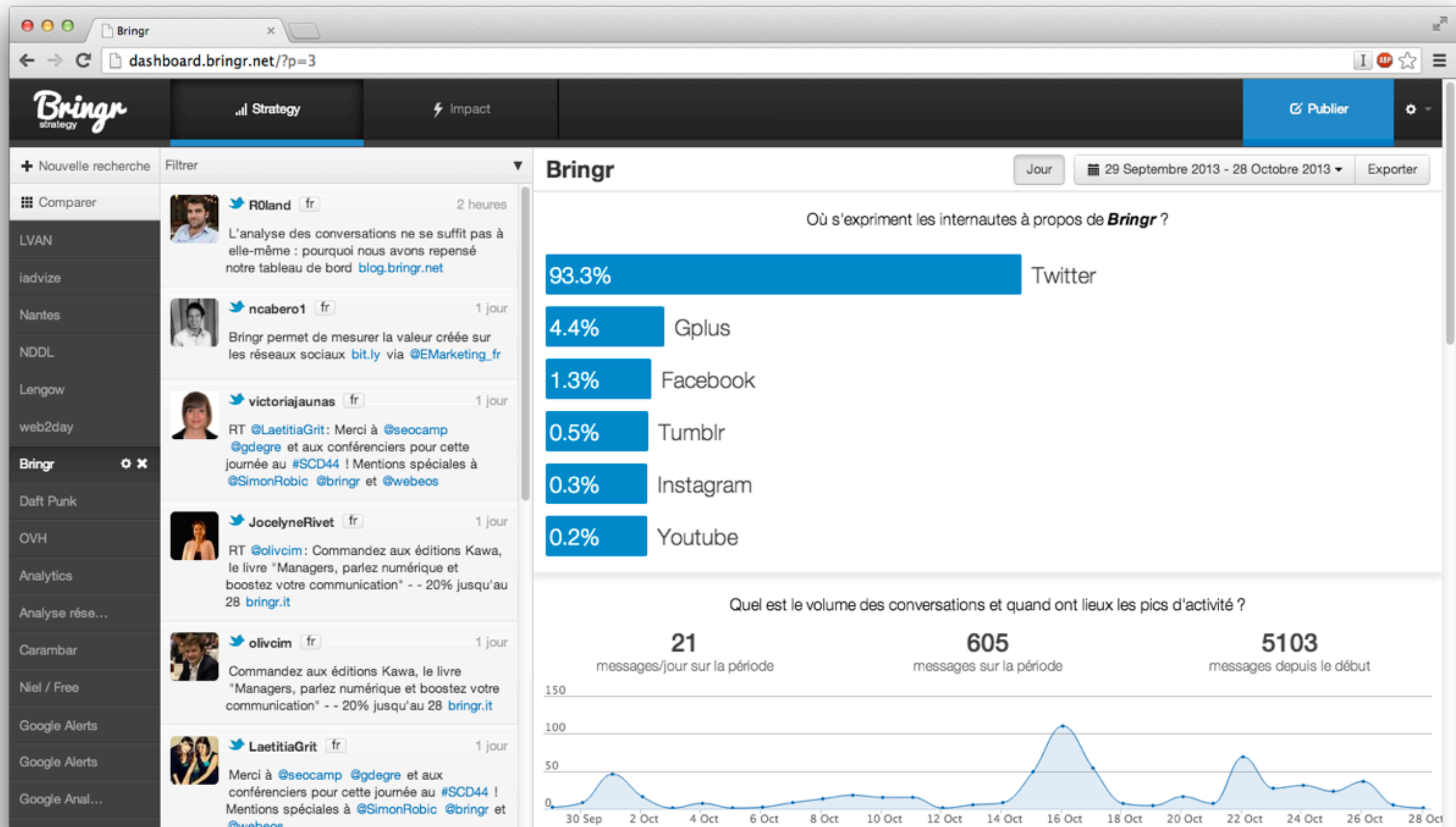
220.000.000.000 commands per day  
500.000.000.000 reads per day  
50.000.000.000 writes per day,  
500+ servers  
2000+ Redis instances!



Why am I talking  
about Redis ?

# Cofounder & CTO of *Bringr*

bringr.net



*“ Create value for your business on Social Media,  
from discussion to conversion ”*



Founder of  **redmin**<sup>™</sup> [redmin.com](https://redmin.com)

*“Administrate everything, monitor in real-time.  
Visualizing and editing Redis data-structures has never been so simple.”*

*“ A free, once-weekly e-mail round-up of Redis news, articles, tools and libraries. ”*

## Redis Weekly

A free, once-weekly e-mail round-up of Redis news, articles, tools and libraries.

Sign Me Up!

ONE e-mail each Friday. Easy unsubscribe. No spam — your e-mail address is safe.

redis weekly

Issue #27  
Feb 14 2014

### featured

#### [Redis 3.0.0 beta1 is out! First beta of Redis Cluster !](#)

The first beta of Redis 3.0.0, or should we say "Redis Cluster"? is out! Our road to get 3.0.0 stable starts officially today.

#### [Redis 2.8.5 is out](#)

If you have scripts dealing with expired keys especially touching a given key with an expire multiple times in the context of the same script, it is a good idea to upgrade ASAP since your replicas / AOF may get out of sync for those keys. Redis-cli now supports the SCAN command. Redis cluster API now supports quorum reconfiguration at runtime.

### reading

#### [Redis Cloud free 1GB beta on @googlecloud](#)

#### [How we cut down memory usage by 82% in our cluster from 340GB to 60GB](#)

RAM memory in comparison to disk storage is not so cheap. Having big Redis cluster could lead to some expenses on such memory. Octavi share some



Thank you.

Questions ?

@FGRibreau

bringr.net redmin.com redisweekly.com