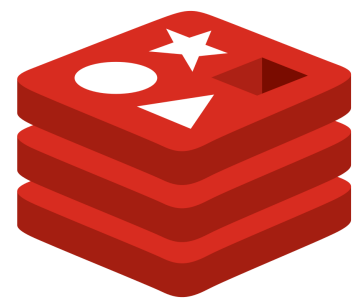
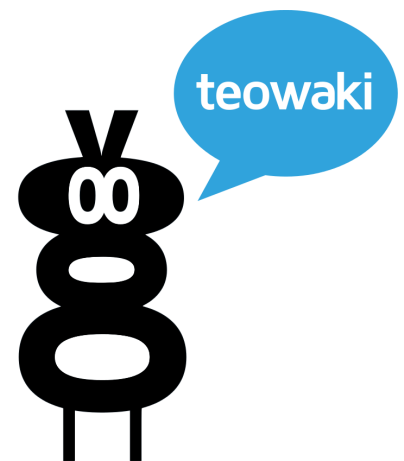




How you can benefit from using



redis



javier ramirez
@supercoco9











**I was
Lois Lane**

redis
has
super
powers

myth: the bottleneck

```
redis-benchmark -r 1000000 -n 2000000 -t get,set,lpush,lpop,mset -P 16 -q
```

On my laptop:

SET: **513610** requests per second
GET: **641436** requests per second
LPUSH: **845665** requests per second
LPOP: **783392** requests per second
MSET (10 keys): **89988** requests per second

On a small digital ocean server (\$5/month)

SET: **227816** requests per second
GET: **228258** requests per second
LPUSH: **251098** requests per second
LPOP: **251572** requests per second
MSET (10 keys): **43918** requests per second

<http://redis.io>

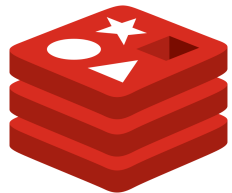


open source, BSD licensed, advanced key-value store.

It is often referred to as a data structure server since keys can contain strings, hashes, lists, sets, sorted sets and hyperloglogs.

started in 2009 by Salvatore Sanfilippo @antirez

112 contributors at
<https://github.com/antirez/redis>



redis



Redis makes
you think

Redis data types

Strings

Hashes

Lists

Sets

Sorted Sets

HyperLogLogs

basic commands

Setting/getting/deleting/expiring

Increment/decrement

Pushing/popping

Checking membership, size...

much cooler commands

Setting only if exists

Blocking pop

(Blocking) pop from one list, push to another

Get/set string ranges (and bit operations)

Set intersections/diffs (and store)

Pub/sub

A chat in 6 lines of code

```
r=Redis.new driver: :hiredis
r.subscribe "chatrooms:42" do |on|
  on.message do |topic, msg|
    puts "#{topic}: #{msg}"
  end
end
end
```

```
r.publish "chatrooms:42", "Hello berlin buzzwords"
```

atomicity

single threaded, so no
concurrency problems

transactions and lua
scripts to run multiple
operations atomically

**Redis keeps
everything
in memory
all the time**

**Does that mean if
the server goes
down I will lose
my data?**

NO *

***unless you didn't configure it properly**

High Availability & scaling out

one master, several
read-only slaves

sharding

Twemproxy & redis cluster

what's being used for

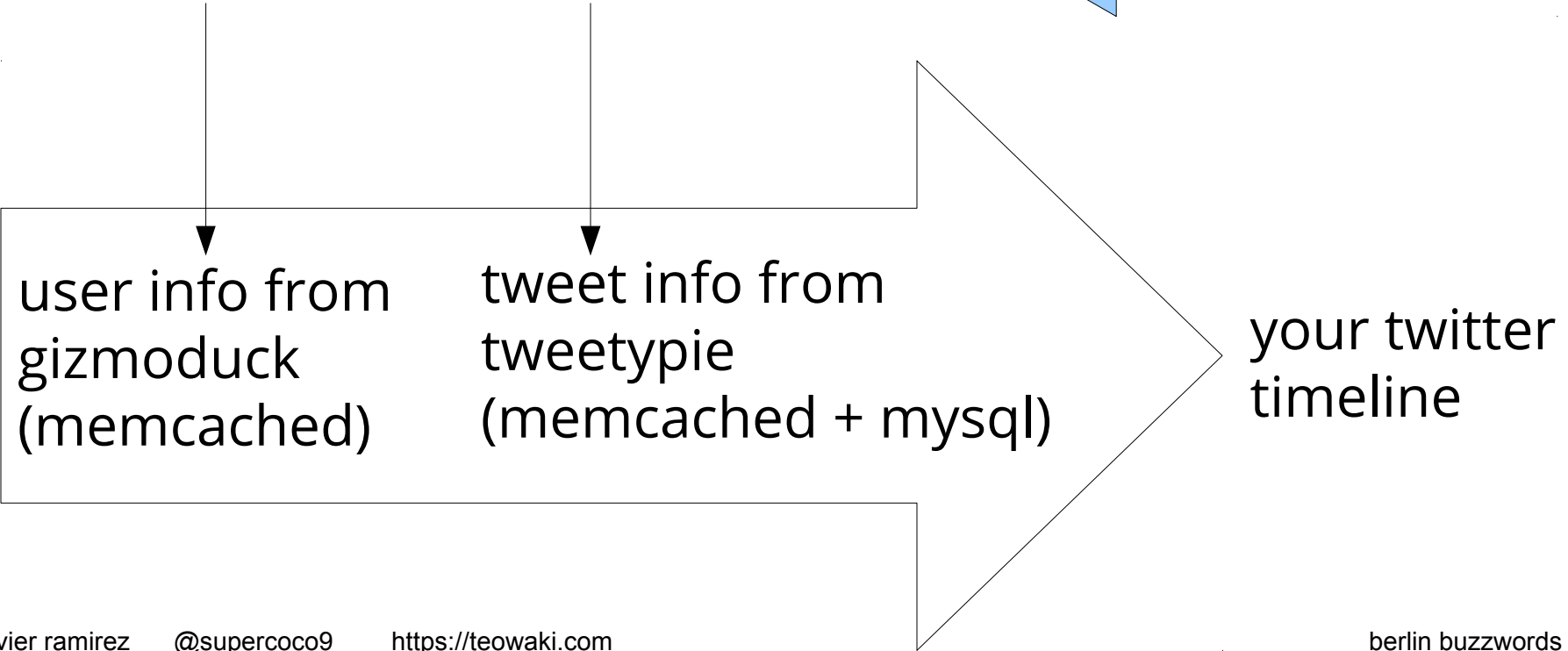
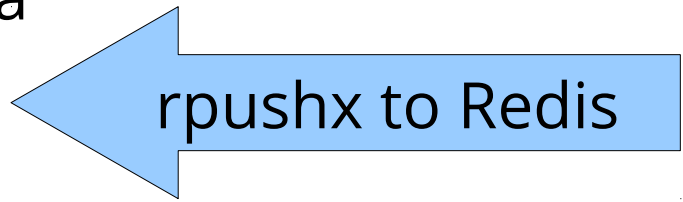
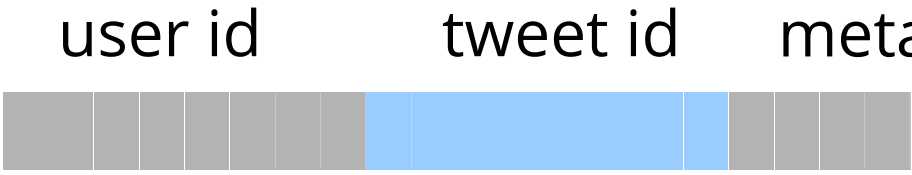
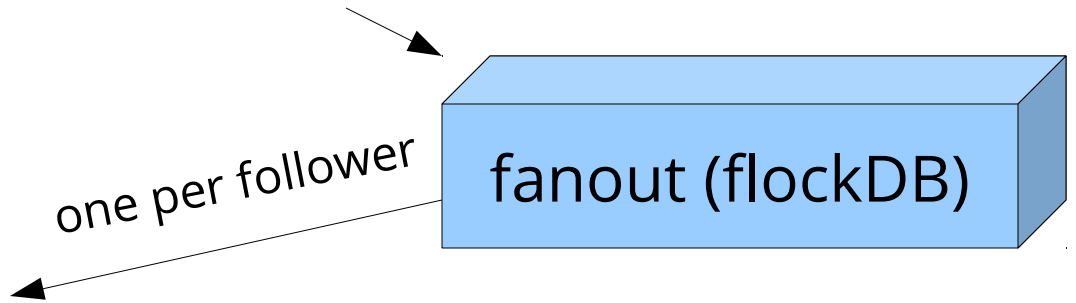
twitter

Every time line (800 tweets per user) is on redis

5000 writes per second avg
300K reads per second

twitter

write API (from browser or client app)



soundcloud

Roshi implements a time-series event storage via a LWW-element-set CRDT with inline garbage collection.

Roshi is a stateless, distributed layer on top of Redis and is implemented in Go. It is partition tolerant, highly available and eventually consistent.

<https://github.com/soundcloud/roshi>

World Of Warcraft

Blizzard is quite secretive about it

Known to have at least 8 servers with 64Gb each to serve user avatars

Known to have seen 1.000.000 users concurrently in the Asian version of the game

stack overflow

Three level cache:

local cache (no persistence)

sessions, and pending view count updates

site cache

hot question id lists, user acceptance rates...

global cache (separate Redis DB)

Inboxes, API usage quotas...

instagram object graph

per user

- Redis SortedSet, with timestamp as the score to store the users followed explicitly
- store the users followed implicitly
- store the user's explicit followers
- store the user's implicit followers
- store boards followed explicitly

Redis set to store boards unfollowed explicitly

per board

- Redis Hash to store a board's explicit followers
- Redis Set to store a board's explicit unfollowers

youporn

Most data is found in hashes with ordered sets used to know what data to show.

```
zInterStore on: videos:filters:released,  
Videos:filters:orientation:straight,Videos:filters:categories:  
{category_id}, Videos:ordering:rating
```

Then perform a zRange to get the pages we want and get the list of video_ids back.

Then start a pipeline and get all the videos from hashes.

snapshot

Redis cluster of 216 master and 216 slaves.
Over 400MM messages per day.
Running on Google App Engine.
No need to have an operations team.

github

Git as a service internally.

Redis is used for storing routing info
matching user repositories to server names

hipchat

Redis for caching.

Information like which users are in which rooms, presence information, who is online...

Redis to balance XMPP, so it doesn't matter which XMPP server you connect to.

when things go wrong



the instagram case

moving from redis to
cassandra: 75% savings on
servers

lesson learnt:
know when redis is not appropriate

pick the right superhero



the twilio case

credit card hell

lesson learnt:

know what you are doing.

Don't change config on the fly



**take the time to master
your super powers**

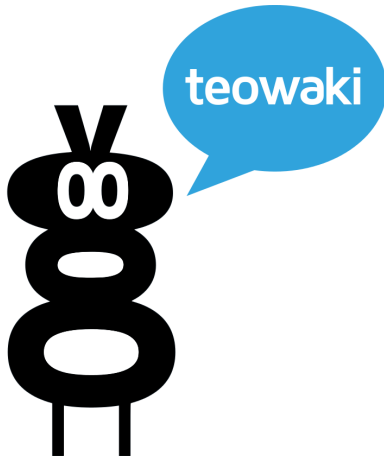
everybody fails, sometimes

Salvatore Sanfilippo
lost his own blog*
because he forgot
to configure persistence

* he recovered it in 30 minutes

you can go
a long way
before having
scary numbers

how teowaki is using redis



Happier development teams.

teowaki improves communication between developers and helps sharing technical information, best practices, gossip and lifehacks with your developer friends.

It's free!

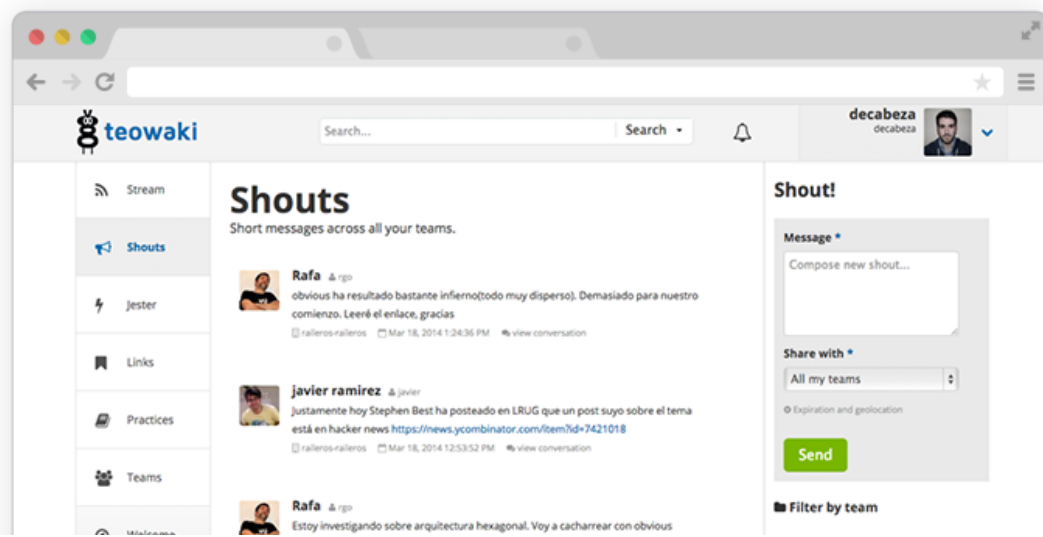


Sign up for teowaki

Discover all the features

What's teowaki. Why you need to start using it.







Communicate with teammates, remote peers, friends from other companies, freelancers, ex colleagues, open source contributors, local group members, or conference attendees in a simple and effective way. [Learn more about the features](#)





javier

<http://javier-ramirez.com>




-  London, UK
-  [blog](#) [blog](#)
-  [github](#) [javier](#)
-  [linkedin](#) [ramirez](#)
-  [stackoverflow](#) [javier ramirez](#)
-  [twitter](#) [supercoco9](#)



javier

Web developer, daydreamer and all around happy person. Founder of <http://teowaki.com>

Teams

-  **2014**
 FOSDEM  javier
-  **fosdem**
 FOSDEM  javier
-  **brecknockroad**
 brecknockroad  javier
-  **community**
 javier  javier
-  **support**
 teowaki  javier
-  **feedback**
 teowaki  javier
-  **The G(uinea) team**
 rgo  javier
-  **raileros**
 raileros  supercoco9
-  **Alumni**
 ASPgems  supercoco9
-  **community**
 teowaki  javier
-  **teowaki**
 teowaki  javier

Taglines



Diego Rodriguez  diec123

Always reading about new technologies and thinking about how to apply them at @teowaki

 Dec 17, 2013 11:57:03 AM

Recent user links

API pivotal tracker  www.pivotaltracker.com

Pivotal Tracker is a simple, effective agile project management tool that allows your team to collaborate around a shared backlog of stories in real time.

 tools



code-collective

Code Collective is a meetup group to share tips and tricks about how to learn programming effectively.

<http://www.meetup.com/CodeCollective>

Code Collective

London, United Kingdom

Invite new members

Shouts

Jester

Links

- autolinks
- Beginner start point
- Github
- HTML/CSS and JS
- Javascript ninjas
- MOOCs and Tutorials
- Pythonists
- Rubysts
- tools
- ux

Members

Leave this team



rosario

rosario



javier ramirez

javier



julienbrun



José María Ruiz

jose-maria



dmacedo



Admin

joansenent

Recent activity



javier

javier shouted Sorry about my previous message! I was demoing the tool at a conference and hit the send button by accident before setting the geographical scope. Sorry about the noise :(

Apr 12, 2014 12:39:57 PM see more...



javier

javier shouted jajaja.. justo estaba haciendo una demo de mandar un mensaje asociado a una location y le he dado al ratón por accidente antes de pinchar en el enlace de geo. perdón!

Apr 12, 2014 12:39:56 PM see more...



javier

javier shouted estoy en codemotion, quien se tona unas birras?

Apr 12, 2014 11:45:18 AM see more...




joansenent


joansenent shared a link. Makers Academy - Launch a Web App in 10 hours

<https://makersacademy-public.hackpad.com/Makers-Academy-Launch-a-Web-App-in-10-hours->





 Stream


 Shouts

 Jester

 Links

 Practices

 Teams

 Welcome

Recent activity

This is what your teammates have been up to.



PivotalTracker published Diego Rodriguez finished this bug


<https://www.pivotaltracker.com/story/show/68791012>

 Apr 3, 2014 3:59:48 PM [see more...](#)



Bitbucket published delicious import search indexing fixed . 94f09f807f2a

<https://bitbucket.org/teowaki/api/>

 Apr 3, 2014 11:13:44 AM [see more...](#)



 diec123

diec123 shared a link. Ruby 2.0 Garbage Collection

<http://blog.newrelic.com/2013/05/28/unicorn-rawk-kick-gc-out-of-the-band/>

 Apr 3, 2014 11:09:19 AM [see more...](#)



Bitbucket published Update to sidekiq 3.0 . ac139970c6a5

<https://bitbucket.org/teowaki/api/>

 Apr 2, 2014 11:20:40 PM [see more...](#)

notifications  



[Help Centre](#)

[The Teowaki Blog](#)

[Engineering Blog](#)

[Privacy and Cookie Policy](#)

[Terms and Conditions](#)

Follow us on  

© 2014 teowaki

Abusing queues

keep track of every activity in the system, even if you don't need them all right now:

- every page view
- every API request
- every time a record is created/updated/deleted

benefits:

- highly decoupled system
- easier to divide into services
- you can add behaviour without changing your app

intermediate cache

- * As a very fast lightweight storage for analytics data before sending them to our google bigquery based solution
- * As a cache for attributes frequently looked up in join tables (names, nicknames, guids, delegated or included model names...)

Some of our uses of Lua

Expiring attributes inside a Redis hash

Inserting notifications into a list only if there are not pending notifications from the same user for the same scope

Paginating a list by an attribute

Manipulating JSON directly at the Redis layer

counters

Atomic counters can be safely invoked concurrently from anywhere, so you can implement “like” features, global sequences or usage monitoring systems in highly concurrent applications for free.

You can share your counters with any other internal application and still be sure they won't collide.

Temporary data

Redis allows us to self expire keys after a time has passed. You can use this mechanism for a simple cache

If you operate on a key with expiration time, you can change its value and still keep the expiration going. By combining a decrementing counter with an expiration time, implementing usage quotas is trivial

Also, you can inspect which keys you have in your server efficiently using SCAN commands

nginx + lua + redis

**Multiple levels of cache by using Redis on the webserver/
middleware layer**

<http://wiki.nginx.org/HttpRedis>

summarizing

- * Redis is more powerful than it seems
- * Very fast, easy to use, simple, good documentation
- * In-memory data structures, distributed, shared and persisted
- * Good as data store, intermediate data store, cache or queue
- * Lots of use cases, both in huge and smaller systems

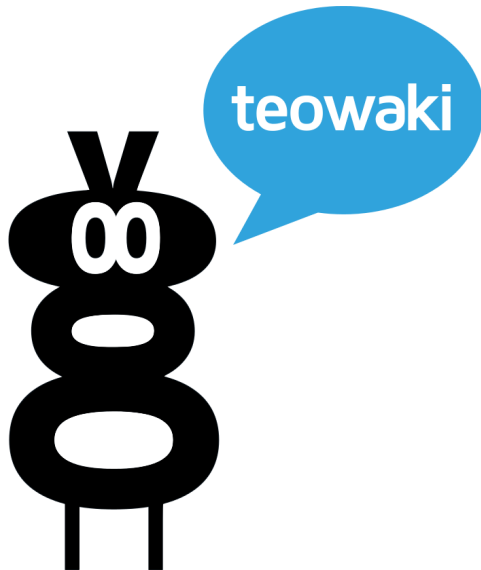
You should probably use it a lot more

Find related links at

<https://teowaki.com/teams/javier-community/link-categories/redis>

Danke.

If you enjoyed this talk, please sign up for
<https://teowaki.com>
Don't forget to invite your friends too!



Javier Ramírez
@supercoco9