
A few (more than one and less than ten) reasons Redis is awesome!

Redis describes itself as an:

"Open source, BSD licensed, advanced key-value cache and store. It is often referred to as a data structure server since keys can contain strings, hashes, lists, sets, sorted sets, bitmaps and hyperloglogs"

— redis.io

This is not a Redis tutorial, just a rant. If you want to learn more, here's a few links:

- The little Redis Book: A great free ebook. Somewhat outdated (ca 2012), but a good start:
 - [PDF](#)
 - [Read online](#)
- Redis: Swiss army knife: [Slides \(2012\)](#)
- Official documentation: redis.io/documentation

1 Small

When you download the latest version from the website, the sourcecode, tests, helper scripts etc. is 6.3 MB. The source itself is ~ 2 MB. The server binary is ~ 4 MB.

2 Simple

Compiling is as simple as one *make*. It takes less than a minute on my laptop, and I have never seen it fail (although I'm sure it can happen).

After compiling you get a bunch of *redis-** files in your *src/* folder. Run *./src/redis-server* and it starts! Done!

There's a lot of config options to set, but adjust when needed, no need to worry about that when trying it out.

3 Versatile

Redis is often associated with caching, but that's just one of its many, many uses. People sometimes call it a "swiss army knife", and that's not a bad analogy. As a low-level tool, it's best seen as a one little program you can use to build a lot of cool stuff upon.

Even the basic datatypes become super useful when you have programs in written in different languages that want to work on the same data. Redis has clients for most languages and platforms. ([seriously](#)). And being fast, Redis actually fills the role of data migrator and interpreter nicely.

Redis supports different data types, each with a slew of uses:

- Strings (basic key-value)
- Hashes
- Sets
- Sorted Sets
- Lists (Stacks, Queues)
- Publish / Subscribe (Message queue)

See [Redis commands](#) for the full, scary list of commands, grouped by data type.

I'm way too lazy to try to summarize the different usecases, play around with them and you'll probably get some ideas. Or try google.

There's also this: [Redis data types and abstractions](#) (long but useful).

4 Fast

A single-server, single-process Redis server does approximately 50 000 - 100 000 operations a second. This means that you can do 50 Redis operations per page load, and still have 1000 concurrent users. Obviously this is not Facebook or Google scale, but it's more than enough for many, probably most, usecases. Scaling beyond this can be a little tricky, but there's examples of people making it work.

5 Useful

Each of the data types supported by Redis has its usecases. Combine this with the cross-platform nature and ease of use, and you get something really useful.

[Redis Is The Most Important Tool In My Toolbelt \(2012\)](#)

6 Honest

There's refreshingly little marketing-bs in the documentation and community. The documentation lists time complexity, in Big-O notation, for each command. And, AFAIK, nobody pretends Redis is going to replace everything else and make RDBMS's obsolete.
