NoSQL Databases

PL08 – Introduction to Key-Value Databases

Teacher: Cristiana Neto
Email: cristiana.neto@algoritmi.uminho.pt

Office hours:
Friday 10h-11h

Summary

1 Introduction to key-value databases

2 Redis Installation

4 Laboratory

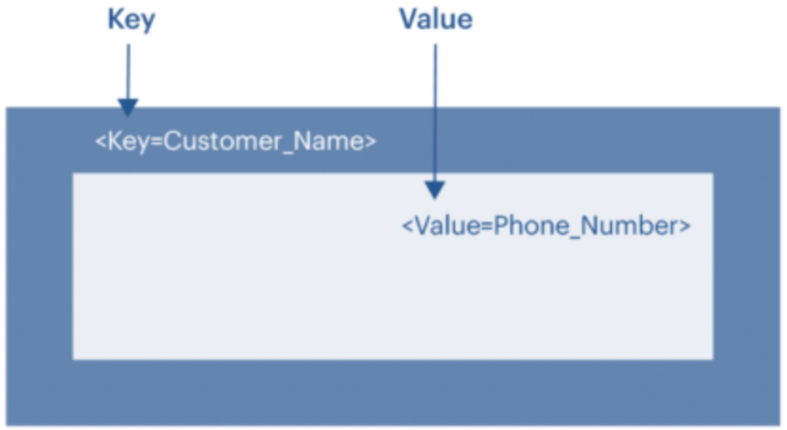
5 FE07 – Worksheet 7

Introduction to key-value databases

- simple key-value pair method to store data;
- contain a simple string (the key) that is always unique and an arbitrary large data field (the value).
- implements a hash table to store unique keys along with the pointers to the corresponding data values.

Phone directory

Key	Value
Paul	(091) 9786453778
Greg	(091) 9686154559
Marco	(091) 9868564334



A simple example of key-value data store.

Introduction to graph databases

When to use?

- Handling large volume of small and continuous reads and writes;
- Storing basic information;
- Applications with infrequent updates and simple queries;
- Key-value databases for volatile data.

Introduction to graph databases

Use cases

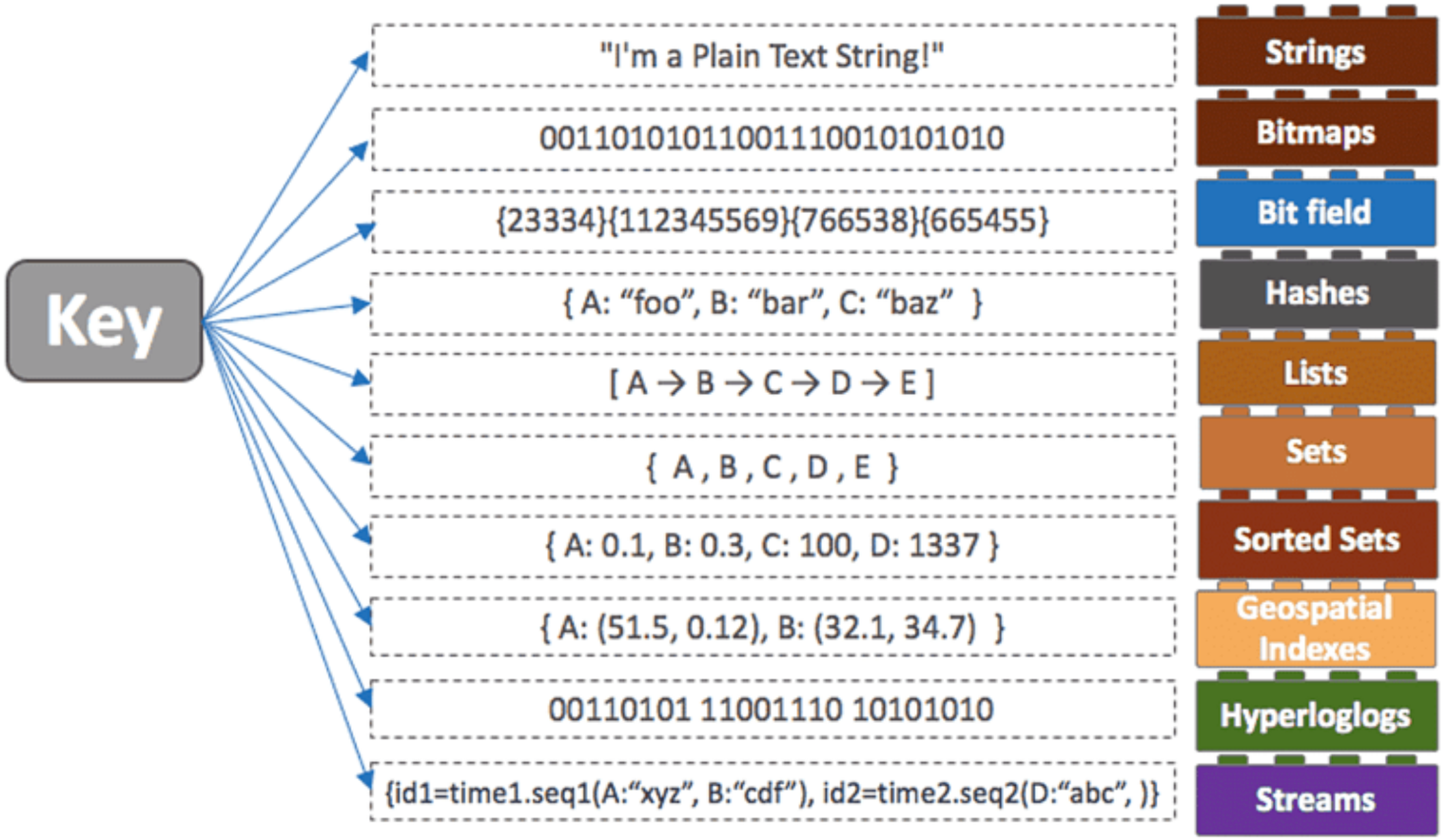
- Session management on a large scale;
- Using cache to accelerate application responses;
- Storing personal data on specific users;
- Product recommendations and personalized lists;
- Managing player sessions in massive multiplayer online games.

Introducing Redis

- **RE**mote **DI**ctionary **S**erver
- First release in 2009;
- Open source;
- Written in C;
- It can handle up to 2^32 keys, and was tested in practice to handle at least 250 million of keys per instance;
- Most popular key-value store .

Redis

Data types



Redis

Operations

Depending on the data types, Redis supports different operations.

The basic operations are similar to a relational database, which supports CRUD (Create-Read-Update-Delete):

- GET: Retrieve the value of a key
- PUT: Create a new key-value pair or update an existing key
- DELETE: Delete a key-value pair

https://developer.redis.com/howtos/quick-start/cheat-sheet/

Redis

➡ Operations

Strings	Used for cache, counter, distributed locks, sessions	GET, SET, MGET, APPEND, SUBSTR, STRLEN
Lists	Used for message queues	LPUSH, LPOP, LLEN, LMOVE, LTRIM
Sets	Used for intersections, unions etc	SADD, SREM, SCARD, SISMEMBER, SINTER
Hashes	Used for caches	HGET, HSET, HMGET, HINCRBY
Sorted Sets (ZSet)	Used for ranking	ZADD, ZRANGE, ZREVRANGE, ZRANGEBYSCORE, ZREMRANGEBYSCORE, ZRANK

https://developer.redis.com/howtos/quick-start/cheat-sheet/

Redis

➡ Operations – Examples

```
[redacted]:~$ redis-cli
127.0.0.1:6379> set foo bar
OK
127.0.0.1:6379> get foo
"bar"
127.0.0.1:6379>
```

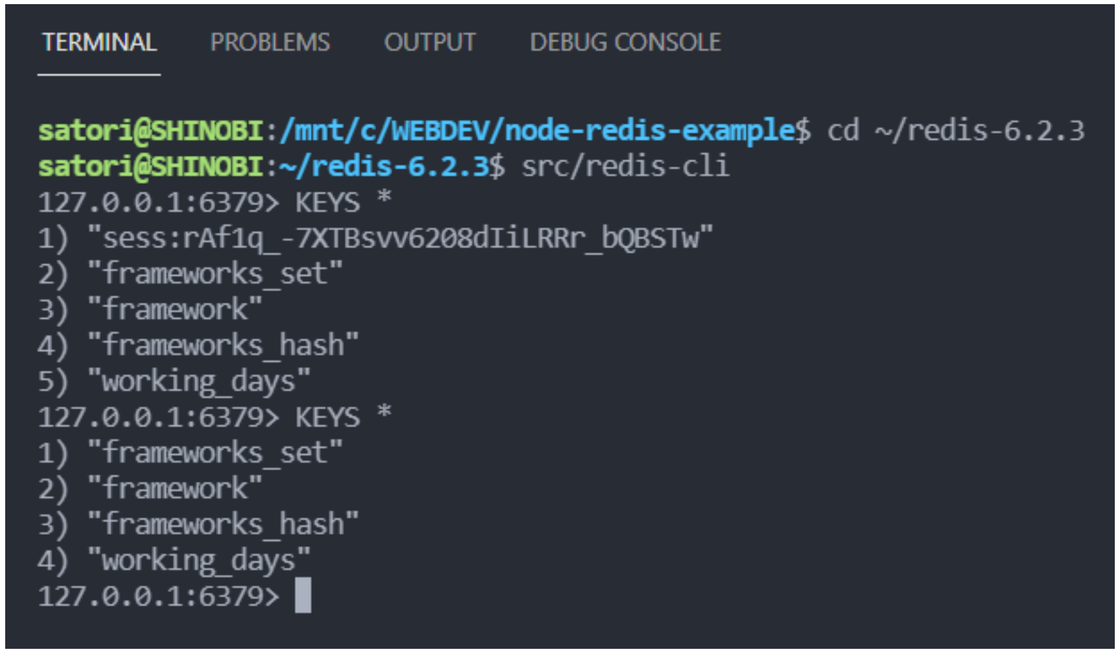
```
# redis-cli
127.0.0.1:6379> RPUSH my-list item1
(integer) 1
127.0.0.1:6379> RPUSH my-list item2
(integer) 2
127.0.0.1:6379> RPUSH my-list item3
(integer) 3
127.0.0.1:6379> LRANGE my-list 0 -1
1) "item1"
2) "item2"
3) "item3"
127.0.0.1:6379> LINDEX my-list 2
"item3"
127.0.0.1:6379> LPOP my-list
"item1"
127.0.0.1:6379> LRANGE my-list 0 -1
1) "item2"
2) "item3"
127.0.0.1:6379>
```

https://developer.redis.com/howtos/quick-start/cheat-sheet/

Redis

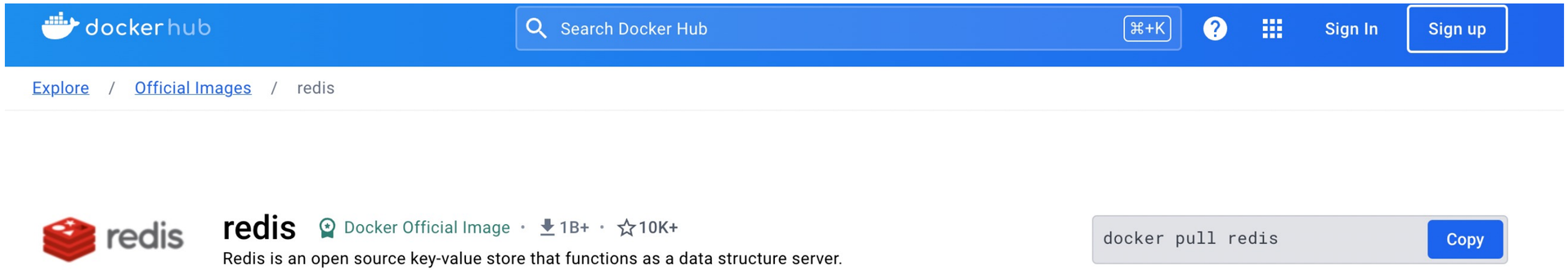
redis-cli

The Redis command line interface (also known as redis-cli) is a terminal program that sends commands to and reads replies from the Redis server.



Redis

Redis container

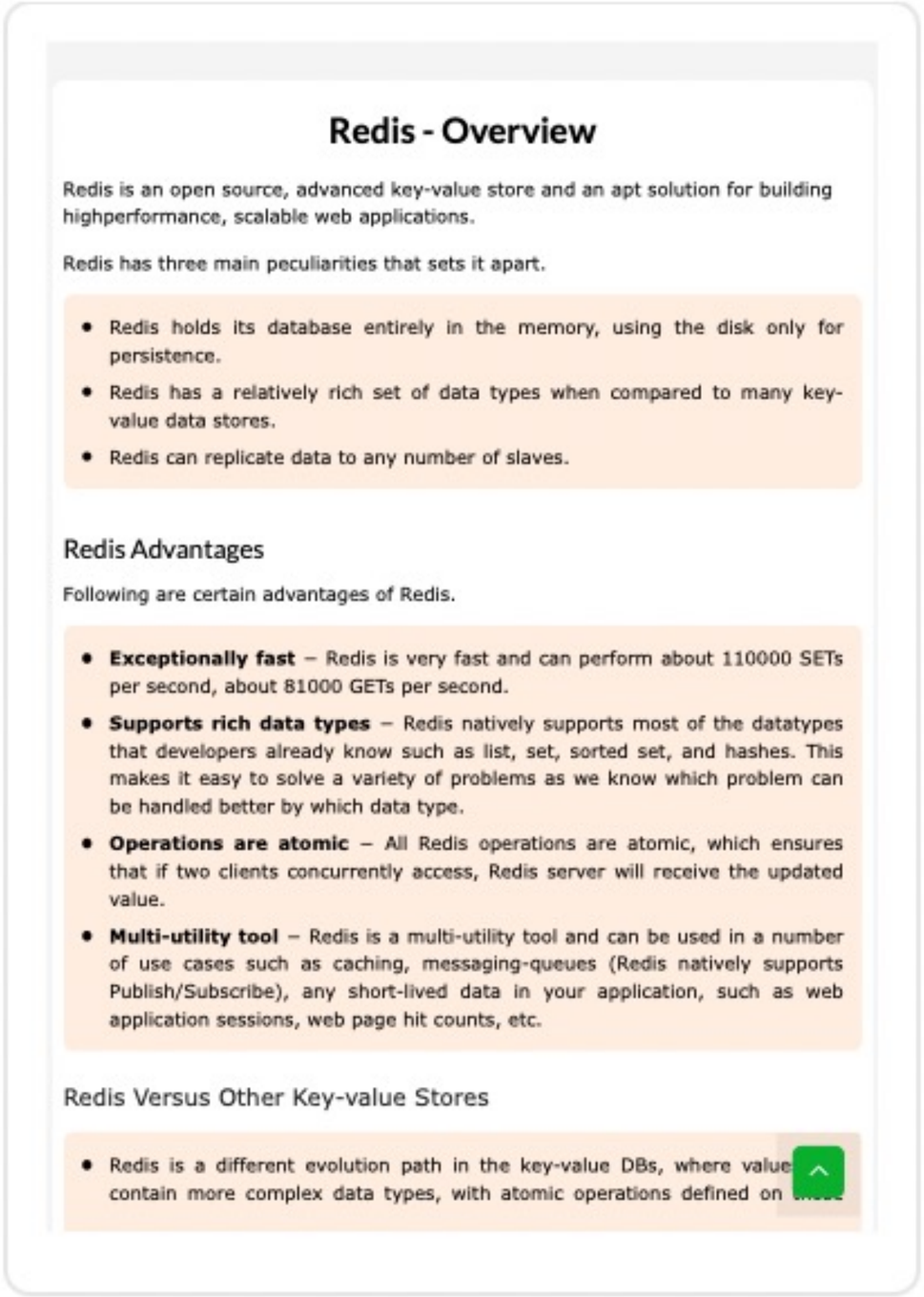



```
$ docker pull redis
$ docker run --name redis -d -p 6379:6379 redis
$ redis-cli
```

https://hub.docker.com/_/redis

Laboratory

➡ **Redis tutorial**



Redis - Overview

Redis is an open source, advanced key-value store and an apt solution for building highperformance, scalable web applications.

Redis has three main peculiarities that sets it apart.

- Redis holds its database entirely in the memory, using the disk only for persistence.
- Redis has a relatively rich set of data types when compared to many key-value data stores.
- Redis can replicate data to any number of slaves.

Redis Advantages

Following are certain advantages of Redis.

- **Exceptionally fast** – Redis is very fast and can perform about 110000 SETs per second, about 81000 GETs per second.
- **Supports rich data types** – Redis natively supports most of the datatypes that developers already know such as list, set, sorted set, and hashes. This makes it easy to solve a variety of problems as we know which problem can be handled better by which data type.
- **Operations are atomic** – All Redis operations are atomic, which ensures that if two clients concurrently access, Redis server will receive the updated value.
- **Multi-utility tool** – Redis is a multi-utility tool and can be used in a number of use cases such as caching, messaging-queues (Redis natively supports Publish/Subscribe), any short-lived data in your application, such as web application sessions, web page hit counts, etc.

Redis Versus Other Key-value Stores

- Redis is a different evolution path in the key-value DBs, where value
contain more complex data types, with atomic operations defined on

FE07 – Worksheet 7

University of Minho
Department of Computer Science

FE07

Course: MSc in Computer Engineering / MSc in Bioinformatics
U.C.: NoSQL Databases

Exercise Sheet PL08	
Teacher:	António Abelha / Cristiana Neto
Theme:	Introduction to Redis
Class:	Laboratory Practice
Academic Year:	2023-2024 – 2nd Semester
Duration of the lesson:	2 hours

NoSQL Databases

PL08 – Introduction to Key-Value Databases

Teacher: Cristiana Neto
Email: cristiana.neto@algoritmi.uminho.pt

Office hours:
Friday 10h-11h