University of Aveiro, DETI

Database Add-ons

Practical class guide

LEI – Degree in Computer Engineering

Year: 2022/2023

lab 1 Key-Value Databases

Goals

The objectives of this work are:

- Understand the fundamentals of key-value databases.
- Install and use an open source solution.
- Develop solutions for various use cases

prior note

This module should preferably be developed in Linux. If you intend to use Windows, check the compatibility notes for the software you will be using.
Submit code/results/reports to elearning. Use a folder (1, 2, ..) for each exercise, compressed into a single file.
Good job!

1.1 REDIS

Redis (*REmote DIctionary Service*) was developed in 2009 and has undergone successive updates, being one of the most popular in-memory key-value repositories. It can be used as a database, as *cache* data, or as a messaging system (*message broker*).

a) Install Redis on your personal computer (https://redis.io/). Run the server (*or go to point b if it is installed as an OS service*):

```
$ redis-server
```

b) Study the functioning of the system by testing the most used commands, through the command line.

```
$ redis-cli
```

c) Consult the slides available for the course (*CBS_05_KeyValue*) and websites with documentation about Redis. Some examples:

- *redis web site*, https://redis.io
- *Introduction to Redis*, http://intro2libsys.info/introduction-to-redis/
- *Redis Java*, https://www.tutorialspoint.com/redis/redis_java.htm
- *Redis Tutorial - w3resource*, http://www.w3resource.com/redis/index.php

Some concepts you should pay particular attention to:

- Read and Write, Persistence, TTL, Types
- Structures (Hash, List, Set, Sorted Set, Streams)
- Operations on structures (ranges, unions, intersections, subtractions)
- ACLs, Transactions, and Namespace

At the end of this exercise, in your root directory you will find a file named ".rediscli_history". Copy it to another file named "CBD-11-<NMEC>.txt" (Where<NMEC> must be replaced by its mechanistic number).

You must deliver the files:

CBD-11-<NMEC>.txt–with the command(s) used in redis-cli.

1.2 Redis - Massive Insertion

Redis allows you to insert data from a file, using the command line. To test this functionality, type in the fileCBD-12-batch.txta set of redis commands (between 10 to 20). Then use it to submit these commands to the redis server.

You must deliver the files:

CBD-12-batch.txt

CBD-12.txt–command(s) used to load the file in Redis

1.3 Redis - Programmatic Access

a) Install a Redis driver for Java (egJedi , available in https://redis.io/clients) and create a small program to connect to the Redis server, repeating some of the previous operations. Note that you will have to use the file*jar*directly or through a project*maven*.

Use the following example as a basis (*based on jedi*):

```
packageredis.ex3;
importredis.clients.jedis.Jedis;

public classForum {
	public static voidmain(String[] args) {
		// Ensure you have redis-server running JediJedi
		=newJedis(); System.out.println(jedis.ping());
		System.out.println(jedis.info()); jedis.close();


	}
}
```

b) Create a program that writes and reads using i) a list and ii) a hashmap. Take as a basis the following example that writes and reads about a Set.

(Note: when switching between types, be sure to use different names for the types to avoid collisions between types).

```
packageredis.ex3;
importredis.clients.jedis.Jedis;

public classSimplePost {
	public static StringUSERS_KEY ="users";// Keyset for users' name

	public static voidmain(String[] args) {
		JediJedi =newJedis(); // add users
```

```
            String[] users = {"A-N-A","Pedro","Maria","Luís"};
            // jedis.del(USERS_KEY); // remove if exists to avoid wrong type for(Stringuser:
            users)
                jedis.sadd(USERS_KEY, user);
            jedis.smembers(USERS_KEY).forEach(System.out::println);
            jedis.close();
        }
}
```

1.4 Autocomplete

a) Using the Redis server and client driver (Jedis) create a program to provide a list of terms, sorted in alphabetical order, for a function of *autocomplete*. Use the file "names.txt" as a basis for the terms to search for. The final iteration with the user might be as follows:

```
Search for ('Enter' for quit): susann susann

susanna
susanah
susanne

Search for ('Enter' for quit): zora zora

zorah
bully
```

b) The file "en-names-2021.csv" contains a list of personal names and the total number of records that were registered in 2021, Develop a variant of the previous paragraph where the search result for *autocomplete* is sorted by decreasing popularity of the name.

You must deliver the files:

* . Java – of the two exercises

CBD-14a-out.txt – examples of program interaction and output a)

CBD-14b-out.txt – examples of program interaction and output b)

1.5 Messaging system

a) Build a messaging system using Redis and the client driver (Jedis). It must provide the following features:

- Adding users (identified by name).
- Association between users (eg if *userA* Follow *userB*, then *userA* must have information about all messages sent by *userB* to the system).
- Sending messages, for example:
  storeMsg(userA, "This will be easy!").
- Message reading (by user, etc.).

Develop and develop some additional functionality. Describe the structures you created/used (eg in a file *readme.txt*).

You must deliver the files:

* . Java –

CBD-15.txt–documentation of data structures/keys used

CBD-15a-out.txt–examples of program interaction and output