Proseminar Distributed Systems

University Innsbruck — Institute of Computer Science

Manfred Moosleitner

12.01.2021

Exercise Sheet 08

Introduction

The purpose of this exercise sheet is to introduce the key-value store Redis to you and make you familiar with its functionalities. This is also the basis for working with a Redis cluster, as you will work with a single node for this exercise sheet.

The exercise sheet consists of 3 exercises, upload your solutions to OLAT before midnight on the day before the next proseminar.

Exercise 1 (Install Redis)

a) Start up and configure an EC2 instance run Redis using the AWS SDK for Java[1]. At the initialization of the instance, the source code of redis needs to be downloaded, build, and run. You can do this via AWS SDK by setting the userdata[2] in the `RunInstanceRequest` (see file `userdata.txt` provided in OLAT) or use an SSH Java library to connect to the instance execute the necessary commands. Further instructions and informations about installing and running Redis can be found here: https://redis.io/download and here https://redis.io/documentation.
Make sure that your code reflects the following criteria:

- Use Amazon Linux 2 AMI (e.g. `ami-0be2609ba883822ec (64-bit x86)`) as AMI.
- Use t2.micro as EC2 instance type.
- Create a new Amazon EC2 security group and make sure to add the necessary ports or reuse a security group with the necessary ports open.
- Create a new Amazon EC2 key pair or use an already existing EC2 key pair.
- Download and install needed software to build Redis, study the documentation and `userdata.txt` to find out what commands need to be run, and use one of the following two ways to execute the commands when starting the EC2 instance:
  a) Use the content of the supplied file `userdata.txt` as user data to automatically set up `redis-server` on your instance[3], or
  b) Use a Java ssh library to execute the necessary commands on the instance.
- Request and store the public and private IPv4 address and DNS of the instance in the Java code and print these information when the initialization is done.

Upload all of your java code, including any utility classes you may have written.

[1] https://aws.amazon.com/sdk-for-java/
[2] https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/user-data.html
[3] Hint: Remember to encodeBase64 the userdata, as hinted in `userdata.txt`

b) An instance type from the general purpose family may not be the best choice when running a Redis server. Which family type of EC2 instances do you think has an advantage when running a Redis server? Why do you think the instance types from your chosen EC2 family are better suited for that task?

   Upload your answers as textfile or PDF.

Exercise 2 (Redis Benchmark)

a) Log into your started EC2 instance, run the command `redis-benchmark -q -n 500000` and answer the following questions.

   - Which commands are executed by `redis-benchmark`?
   - Briefly describe the commands used by the benchmark.
   - What can you infer from the values displayed?

b) To test your answer from Exercise 1b, start up another instance with the code of Exercise 1 and use a different instance type this time. Choose an instance type which belongs to the EC2 family you gave as answer in the previous exercise. Arrange within your project team that each member uses a different instance type.

c) Log into the new instance, run `redis-benchmark` again with the same parameters as before, and compare the results of the individual runs. Does the comparison support your hypothesis from Exercise 1b about which family gives a better performance?

Upload the output of each benchmark run and your answers as textfile or PDF.

Exercise 3 (Inserting Data and Working with Redis)

Use Redis the command line interface to insert the data provided in the file `salary.txt` (provided in OLAT) into your Redis server. The data contains pairs of names as keys and a random real value between 10.00 and 99.99 as value. Use `redis-cli` to query your Redis server to answer the following questions and list the query/commands that were executed and their output.

- How many key-value pairs are exactly stored on your server?[4]
- Is your name included as key? Which value is stored for your name? If your name is not included, choose one of the names in the data and give the stored value for your chosen name.
- Insert a new entry (use the concatenation of your first name and last name as key), and use real value between 10.00 and 99.99 as value.
- List names that are similar to your own first name.
- Is it easy to make a query to retrieve ***all names with value between 40.00 and 50.00*** in Redis? Why or why not?

Upload your queries, their output and your answers as textfile or PDF.

[4] Hint: If your answer to this question is 0, then something went wrong with the mass insertion.