Report 5: Groupy - A Distributed Hash Table

Michail Roussos, michailr@kth.se

October 10, 2023

1 Introduction

In this exercise we learnt about the Chord scheme and we implemented some modules to implemented a distributed hash table following this scheme. The first implementation was more simple and could only support the creation of the structure of a distributed hash table ring. The second implementation could also handle actually storing keys and values as well.

2 Main problems and solutions

We had to implement the following modules:

1. The first module we had to implement was *key*. This module contains two basic functions that we will need.
2. The second module we had to implement was *node1*. In this module we will implement a basic node of the system.
3. The third module we had to implement was *node2*. In this module we will implement an enhanced node of the system, that will be able to store keys and values as well. In this module we will have to add some extra functionality to handle the keys an the values (e.g. merge, handover).
4. The fourth module we had to implement was *storage*. In this module we will implement the storage that will be used in the enhanced node along with some basic functions.
5. The fifth module is *test*, where some functions that help with the execution of the project.

A ring comprised of nodes created using *node1* can only form the structure of a distributed hash table and it can not store keys or values. When using the *node2* you can actually store keys or values and perform some operations using the *storage* module as well.

3 Evaluation

As expected the first version only has some basic functionality.

In the second version it can be used to store things as well!

4 Conclusions

In this assignment we had the opportunity to further our Erlang skills, we learnt about the Chord scheme and had the opportunity to implement a hash table in Erlang.