CSE-411: Advanced Programming Techniques

Yifeng Xu

10/07/2023

Homework 2

1.Introduction

In this assignment, the goal is to implement a multi-threaded server and client, by inputting data in a fixed format in the client to the server, and then the server distinguishes which operation is to be performed and performs the corresponding operation before returning the result to the client for display.

2.System Design

On the server side, it uses its own constructed structure (fig 1) to save the data passed in by the client that needs to be saved and adds the structure to the Vec (fig 2) to facilitate the operations of searching for saving and deleting.

```
struct Node {
key-len,
key,
value-len,
value
}
```

Figure 1. struct Node

N ode1	N ode2	N ode3	N ode4	N ode5

Figure 2. Add the struct to Vec

A thread pool is used to manage and control the threads for concurrent tasks, and the maximum number of threads is set to 4 to ensure the safety of the whole program. Parses the input stream and reads out different requests for saving, modifying, reading and deleting (fig 3).

PUT key-len key value-len value	GET key-len key	DEL key-len key

Figure 3. Different types of input to server

3.Implementation

On the client side, the method of prompting the user for input is used to obtain the input request and data, and the input request is judged whether it belongs to put, get, and del; if not, it is re-entered, and if it is a space, the client is terminated. Whatever data is entered can be transmitted to the server side for related operations. The server side splits the input stream and recognizes the type of operation, and then performs the corresponding operation on the remaining data.

About testing, my test case first needs to open four clients at the same time：

1. perform put operation, get operation and del operation from four clients at the same time to observe whether the thread pool in the server can handle concurrent operations correctly

2. perform put operation, get operation and del operation in different clients to observe whether it is possible to perform more operations on them after performing put and del operation in other clients

3. observe the relationship between the number of operations and time, because the time of each thread is certain, to observe whether the operation time rises smoothly because the number of operations increases. Observe the relationship between the number of operations and time, since each thread has a certain amount of time, to see if the operation time rises smoothly due to the increase in the number of operations instead of exponentially increasing due to resource contention or deadlock conditions.

4.Analysis

Successfully connected to the server through four clients and through test cases 1 (fig 4) and 2 (fig 5) found that thread concurrency is useful, there is no resource contention and deadlock phenomenon.

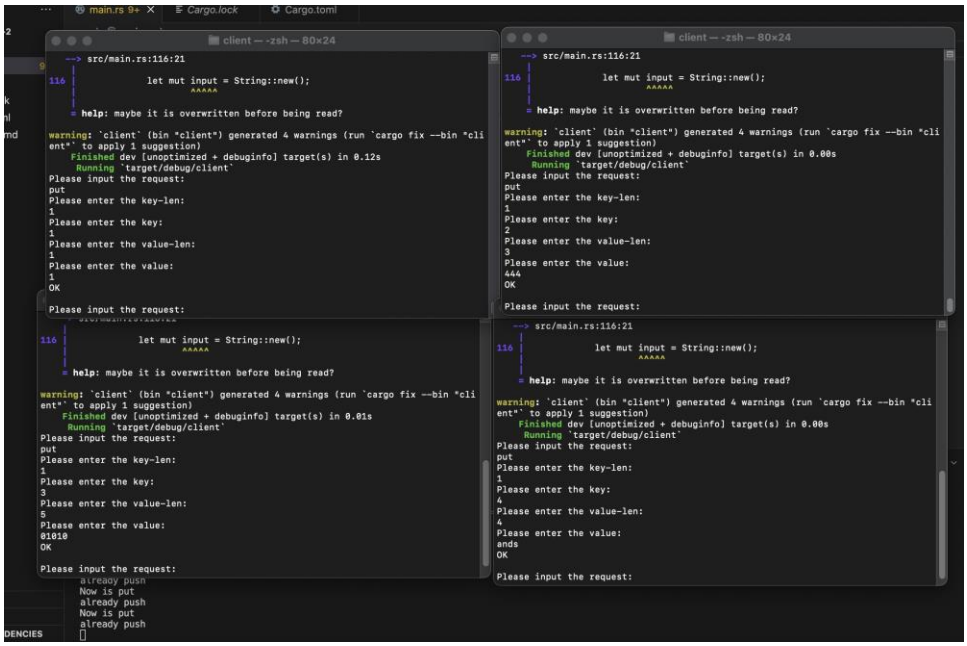

Figure 4. The result of test case 1

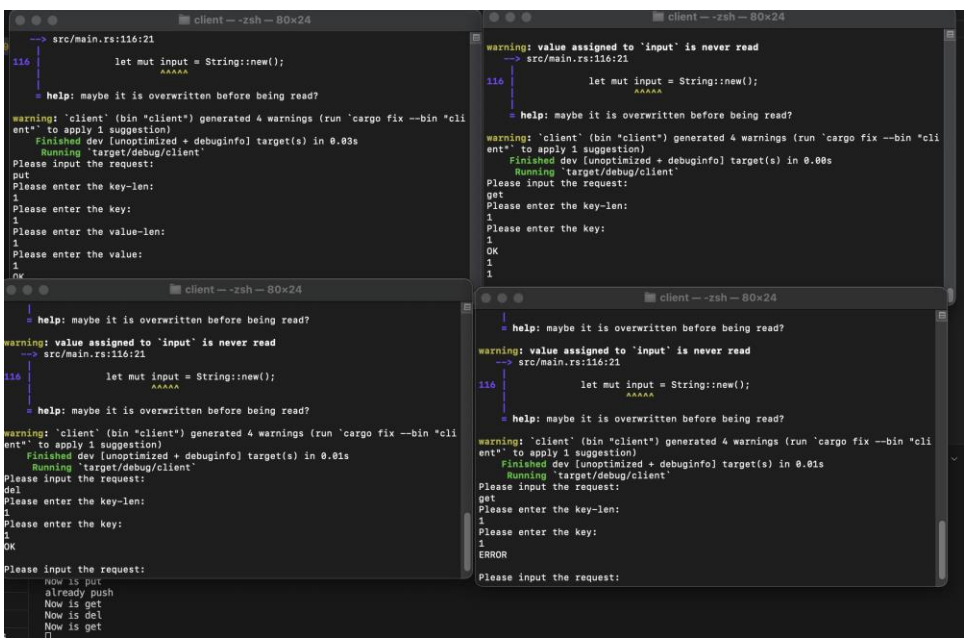

Figure 5. The result of test case 2

Completion of test case 3 by Instant testing the cost time of operations (fig 6).

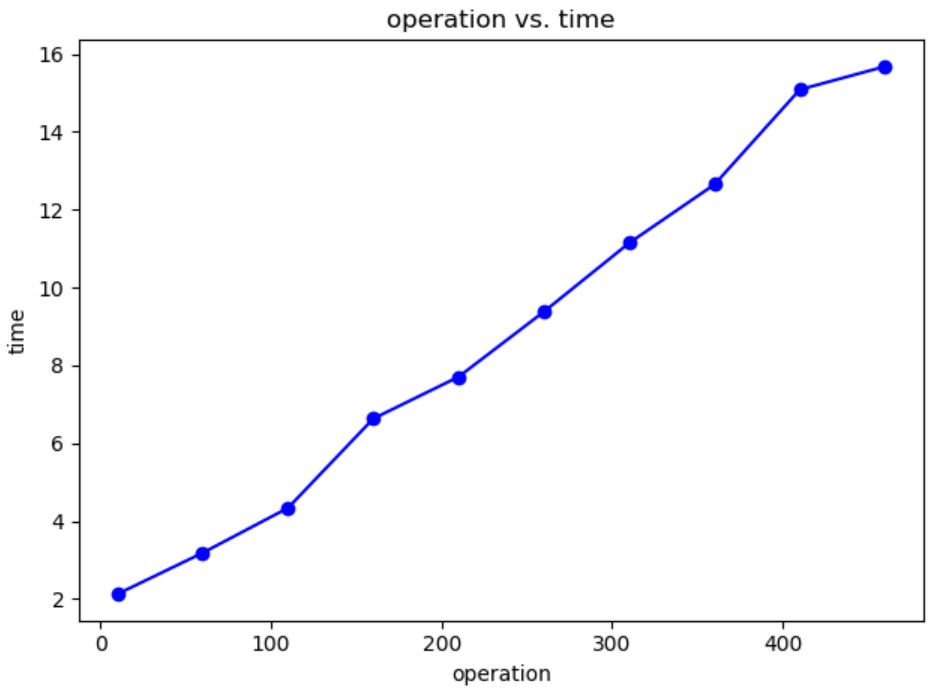

Figure 6. The result of test case 3

5.Conclusion

With this assignment, network communication between the client and server is successfully performed through port 1895. I also found that we need to pay more attention to the problems between the threads in the actual production development, so that our program can be more secure.