COMP90015 2022 S1
Assignment1 - Multithreaded Dictionary Server

Hongzhuan Zhu, 1223535

Problem Context

This project is a multi-threaded dictionary that allows multiple clients can access and modify a shared dictionary. The main architecture implemented is a client-server architecture. On the Server side, it implements the thread-per-connection architecture to allow different clients to add, remove, query and update the word concurrently.

The main protocol implemented is TCP, which provides reliable communication between the client and server. JSON file format is the main format implemented to process the dictionary.

In the context of failure, there is two way to handle the error, the first way is displaying the error message in the dialogue, this kind of error can be figured out by the user themselves. For instance: input missing, word existing, query failed, etc… On the other hand, another type of error is the error that can not be fixed by the user, they are mainly about the I/O expectation error, unknown host error, unexpected error, etc…

There are four main functions in this project, which are add, remove, query and update. The logic for all is listening to the event from Client UI, sending command to the server, server process command and dictionary run different operations based on the command, finally, feedback will send back to the client and show the result to the user.

System component

Server

The server is implemented using thread-per-connection architecture. For each Client, it will connect to the server first and the server will create the connection to this client. The thread will not be terminated after one command has been processed, it only terminates when a client has stopped the connection.

All operations about the word proceed in the Server, and each operation is synchronized, which means there is no crash between multiple clients. In terms of process data, the Dictionary helps to process data, load files and converting into the final format. JSON is the only format to store

the data but in the real running, the hash map is also used to help process data because it is more convenient to modify and manage data.

In a logical way, the connection seems like a bridge between client and server. Actually, the connection is created and maintained by the server itself, but in the real running, the input and the output are completed by the buffered reader and buffered writer. The server uses them to receive commands sent from clients and send feedback to the client.

Client

The Client consists of two parts: Client Controller and Client View(GUI). The Client Controller is mainly responsible for establishing connections, creating a buffered reader and a buffered writer for GUI to implement them. The Client View is mainly responsible for listening and processing each operation by the user entering and providing feedback to the user letting them know what's going on with the current step.

UI

There are 2 UI in this application: Client UI and Server UI, all of them are running with the Client and Server as the child thread, they are implemented by the Swing window. The Client UI is providing an interface that allows the user to add, remove, update and query the word. The Server UI is a support UI that let the developer know the current host and port.

System Diagram

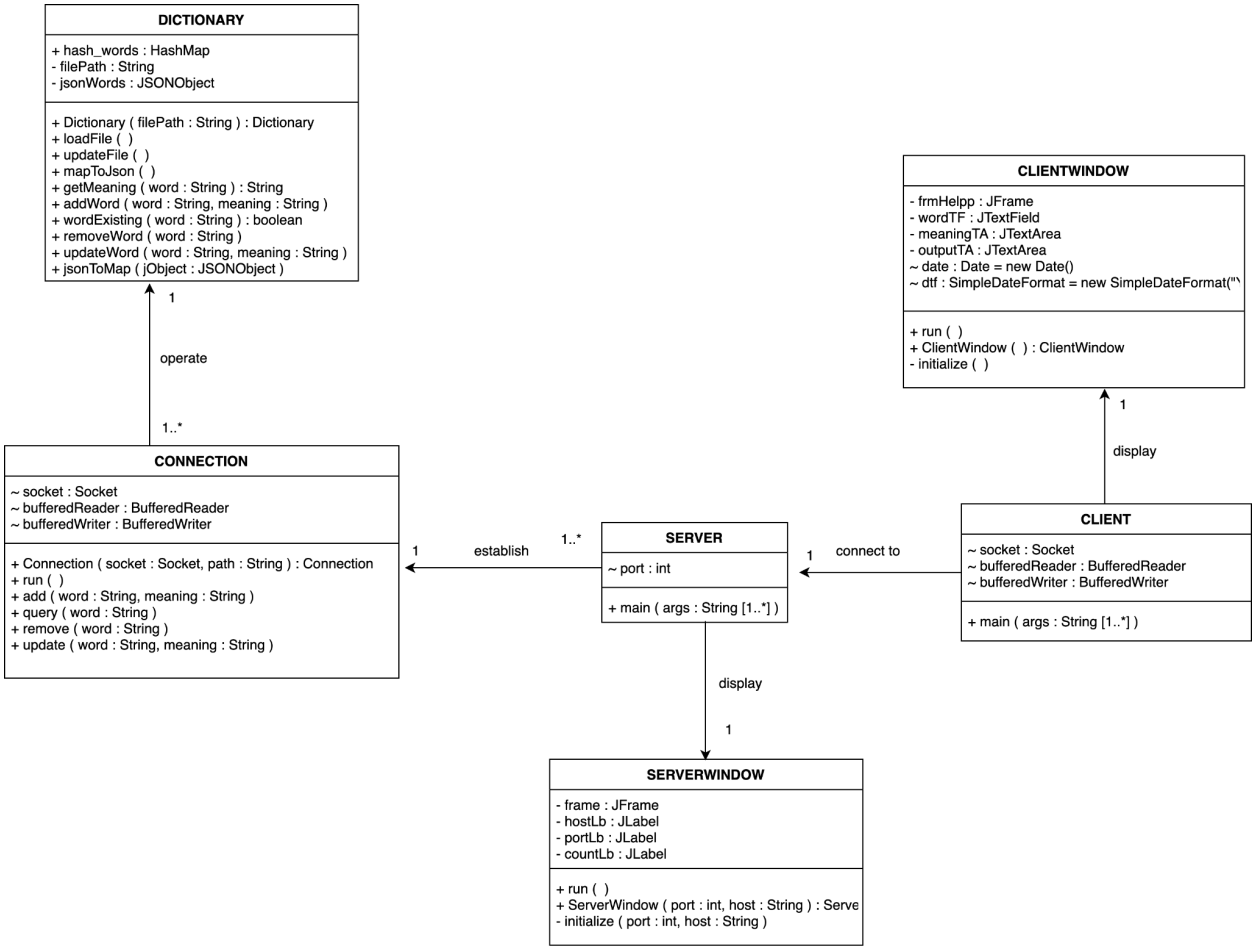

Figure 1: Class Diagram

When the Client starts running, there are two threads running concurrently, the first one is Client UI, which will display the window to the users, and the second one is Client controller, which will build the connection to the server.

When the Server starts running, there are two threads running concurrently, the first one is Server UI, which will display the server window to the developer, and the second one is the logic module of the server, which will create the connection thread after receiving the connection request from the client.

When the Connection thread is running, it will connect the buffer and dictionary first, then read and process the input from the client, write the feedback to the client, and update the JSON dictionary file.

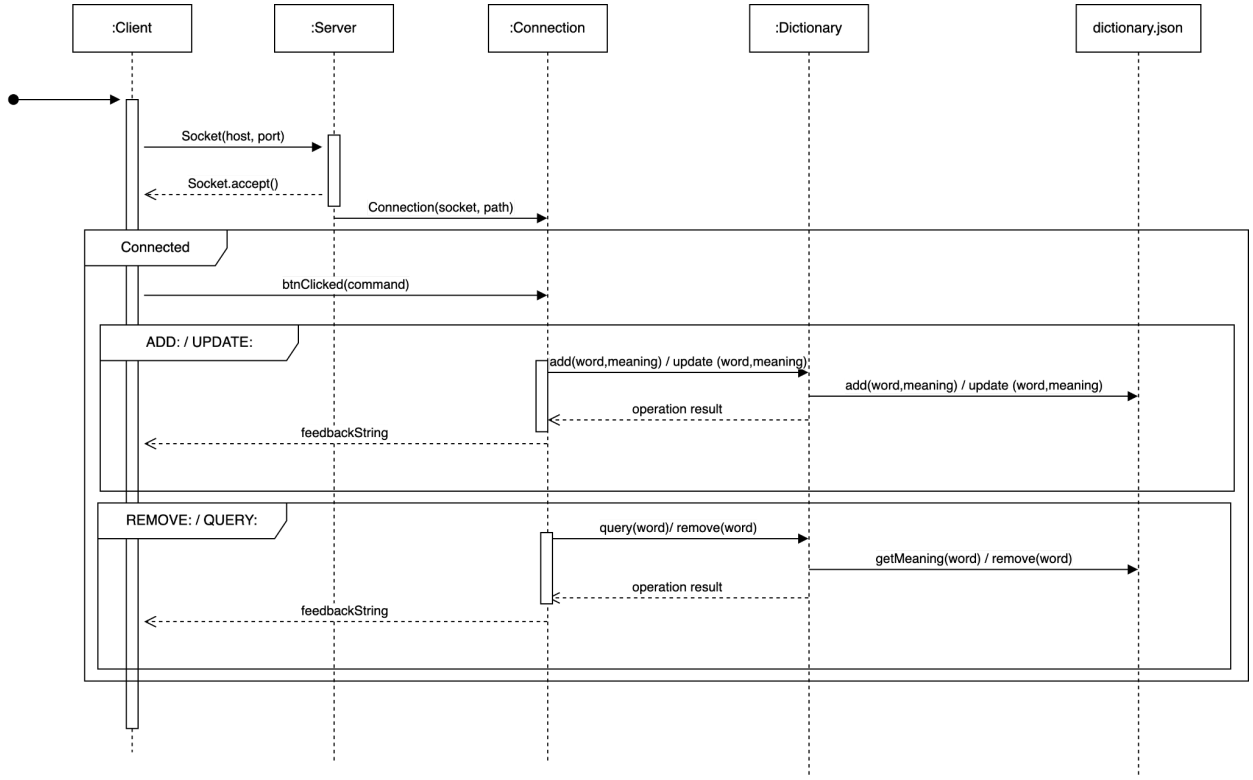

Figure2: Sequence diagram

The process can be summarized as below:

Connection scenario:

1. Start the server
2. Client build connection
3. Connection established, Server runs connection thread to handle transmission between client and server.

Operation scenario:

1. Enter a word or meaning, and click the button to add/remove/update/query.
2. Command send to Server
3. The connection process the command
4. Update dictionary
5. Send back to Client
6. Client display result

Critical Analysis

Advantages

1. The thread-per-connection will not use a new thread to handle a new request, only one thread to handle all requests based on this connection, it can reduce the cost of thread across multiple requests from the same client process, therefore, for a long time duration connection, its performance is better.

2. The dictionary object is unique and all clients are write to and read from this dictionary, therefore there are concurrent issues during the running time, the use of synchronized is to make sure when a thread request one of the operations(add, update, remove and query) which is a synchronized function, the other thread cannot access them concurrently, therefore, it can reduce the crash during the running time

3. During the running time, there are some errors that could occur, almost all errors related to the user's wrong operation such as missing input, searching for the wrong word can display in the way of dialogue, which provides an understandable way to let the user know what went wrong.

4. The use of buffered streams can handle synchronous processes more efficiently, it reads a single character, it will stores the content to fill its buffer, and for further requests, the characters will be read directly from the buffer, therefore, the efficiency can be increased.

Disadvantages

1. Although the thread-per-connection can handle a long-time duration, with the growth of the client, the performance of the thread-per-connection will be declined, it does not support the load balancing effect. Additionally, the thread is a useful resource, it is a waste if a thread has been occupied by a connection with latency.

2. Since the server does not restrict access from the client, anyone can access the server, which increases the risk of being attacked. Additionally, there is no strict restriction on the input of the client, there may be illegal input causing the client or the server to crash. Moreover, the dictionary is not encrypted, and there is a risk of data loss or hacking.