CS 5523: Bonus Programming Project
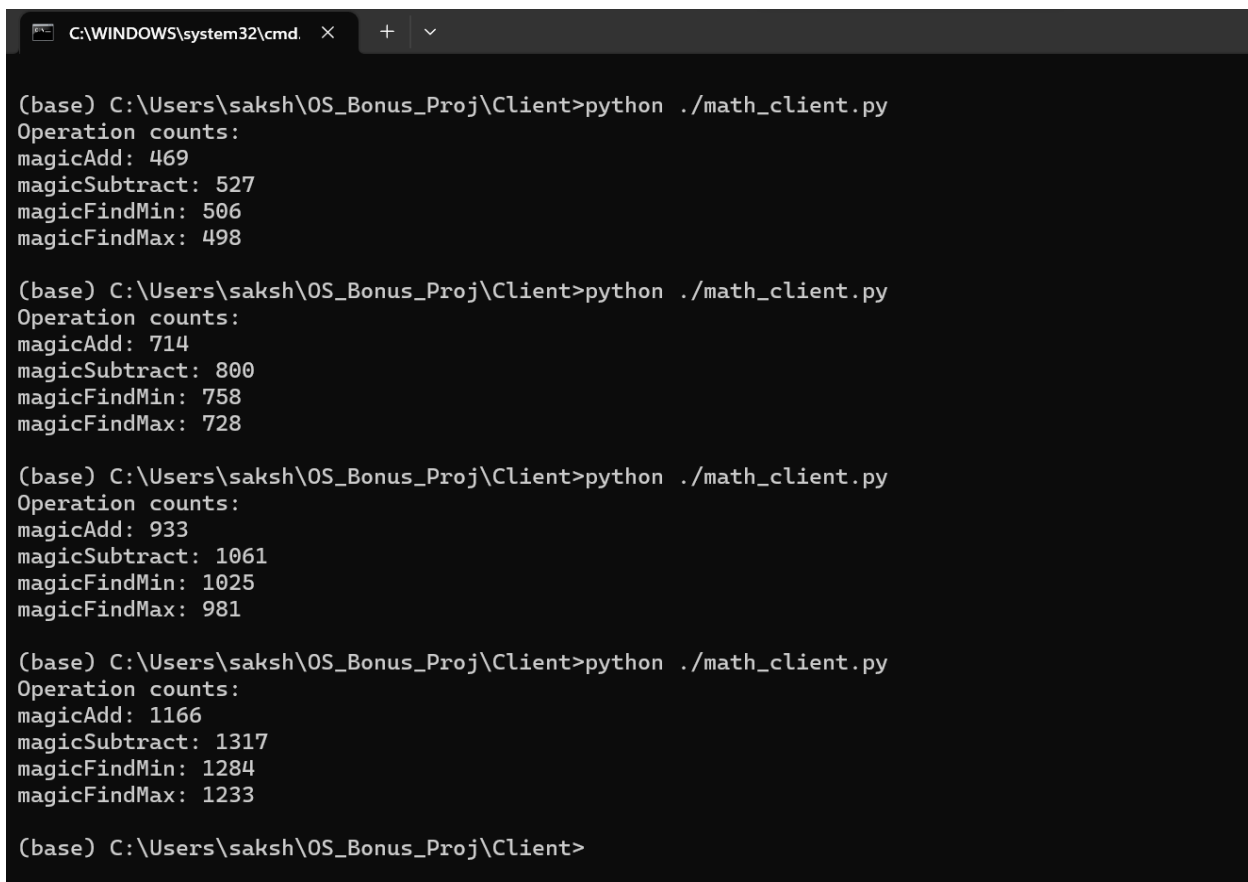
- Sakshi More (ibo836)

The objective of this project is to improve the comprehension of client-server architecture in distributed systems and gain experience in using Remote Procedure Calls (RPC). The project involves implementing a Math server with four remote procedures: addition, subtraction, finding the minimum, and finding the maximum.

The server is designed to handle multiple clients concurrently and track the number of requests made by each client for each procedure. The project currently supports a maximum of 10 clients, each using a different computer.

Python was used for implementation, and the process was simplified with the assistance of the gRPC library. To ensure efficiency and client isolation, multithreading was enabled, allocating a separate thread for each client.

- Figure 1 shows the server running on port 50051


```
(base) C:\Users\saksh\OS_Bonus_Proj\Client>python ./math_client.py
Operation counts:
magicAdd: 469
magicSubtract: 527
magicFindMin: 506
magicFindMax: 498

(base) C:\Users\saksh\OS_Bonus_Proj\Client>python ./math_client.py
Operation counts:
magicAdd: 714
magicSubtract: 800
magicFindMin: 758
magicFindMax: 728

(base) C:\Users\saksh\OS_Bonus_Proj\Client>python ./math_client.py
Operation counts:
magicAdd: 933
magicSubtract: 1061
magicFindMin: 1025
magicFindMax: 981

(base) C:\Users\saksh\OS_Bonus_Proj\Client>python ./math_client.py
Operation counts:
magicAdd: 1166
magicSubtract: 1317
magicFindMin: 1284
magicFindMax: 1233

(base) C:\Users\saksh\OS_Bonus_Proj\Client>
```
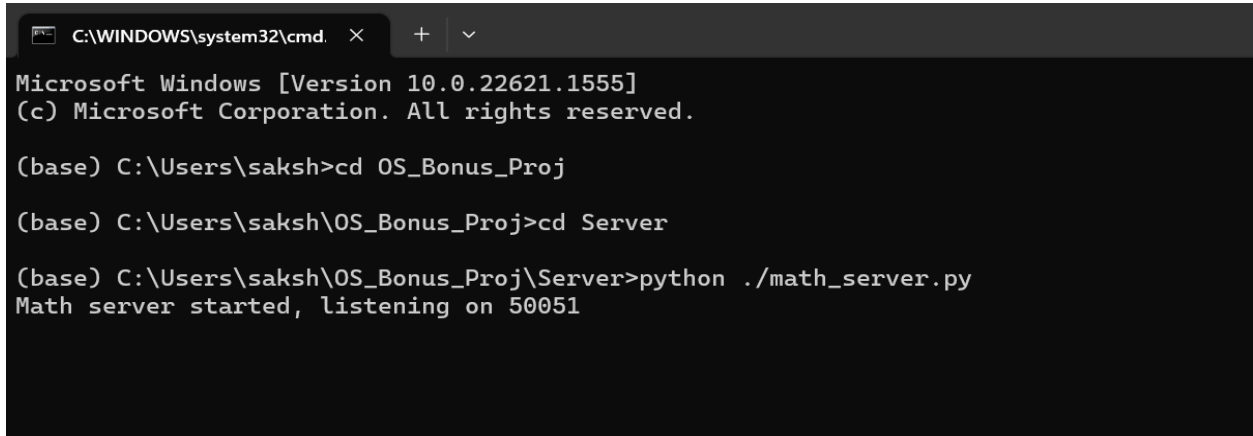
(Figure 1)

- In Figure 2, the results of running the client on the local computer using the localhost are depicted. The first six lines of code in Figure 2 demonstrate the program's output, indicating the total number of operations executed. Additionally, the last six lines of the code illustrate the program being rerun, with the previous operation counts being combined with the new counts. This ensures that the client can accurately track the total number of operations performed.

```
C:\WINDOWS\system32\cmd    ×    +   ∨

Microsoft Windows [Version 10.0.22621.1555]
(c) Microsoft Corporation. All rights reserved.

(base) C:\Users\saksh>cd OS_Bonus_Proj

(base) C:\Users\saksh\OS_Bonus_Proj>cd Server

(base) C:\Users\saksh\OS_Bonus_Proj\Server>python ./math_server.py
Math server started, listening on 50051
```

(Figure 2)

The project includes two folders:

Client Folder:

- Within this folder, you will find the script math_client.py, intended to be executed to establish a connection with the desired server.
- For math_client.py to run successfully, it should be placed in the same folder as math_service_pb2_grpc.py and math_service_pb2.
- The server must be actively running for the client to establish a connection and interact with it.
- Modifying the code is necessary on line 11, where the server's IP address needs to be entered.
- It is essential for both the server and the client to be on the same network in order to communicate effectively.

Server Folder:

- Within this folder, you will find the script math_server.py, which is intended to be executed to start listening to clients and handling their requests.
- For math_server.py to function correctly, it should be placed in the same folder as math_service_pb2_grpc.py and math_service_pb2.
- No modifications are required in this code as it automatically retrieves the local host IP address to set up the server.

Steps to implement:

1. Install the following libraries:

   pip install --upgrade google-api-python-client
   python -m pip install grpcio
   python -m pip install grpcio-tools

2. Run the following from the Server folder C:\Users\saksh\OS_Bonus_Proj\Server> python .\math_server.py

3. Open a new terminal and from the Client folder C:\Users\saksh\OS_Bonus_Proj\Client> python .\math_client.py