

Project Report: Math Client/Server - Middleware RPC

This project involves the development of both an RPC server and client in Python. The server provides basic arithmetic operations (addition, subtraction, finding minimum and maximum) and keeps track of the count of each operation. The testclient connects to the server and randomly selects operations to perform, simulating a client-server interaction.

The server is implemented using the python ``xmlrpc.server`` library. It exposes four functions (``add``, ``subtract``, ``findMin``, ``findMax``) for the basic arithmetic operations. Additionally, a function ``getOperationCount`` is provided to retrieve the count of each operation. Per the assignment instructions, the functions are exposed to clients as their opposites. So the exposed ``magicAdd`` function calls the ``subtract`` function, ``magicSubtract`` calls the ``add`` function, and so on. ``getOperationCount`` returns the correct counts. There is also error handling if the wrong data type is sent to the functions. Each function verifies that the arguments are of the correct type and returns "ERROR - NO RESULT", if they are not.

The client uses the ``xmlrpc.client`` module to connect to the server. It takes the IP address of the server as input and can optionally specify the number of times to call the server. The client randomly selects an operation and generates random numbers to pass as arguments. It then calls the corresponding function on the server and prints the result. At the end of the run, the client will print the operation counts and produce a graph with the same data. Counts will be from the startup time of the server and not the client.

To use the project, first start the server using the provided code by running ``python mathServer.py``. The server will output its current IP so it can be used as an argument. Then, run the client script, providing the IP address of the server as an argument. The client is executed by running ``python testClient.py <IP of server>``. Optionally it can be executed with a count argument. It will default to ten iterations. The client will connect to the server and perform the specified number of operations.

This project demonstrates the implementation of an RPC server and client for basic arithmetic operations. It showcases how RPC can be used for simple client-server communication in Python. The project can be expanded to include more operations or additional functionality as required.

The following graph shows the Operation Counts for a run of 2000 operations.

