

## Design Description:

All the changes are made on the `client_stub.py` and the `server_stub.py` file.

Two sets of APIs with identical names are implemented on the `client_stub` and the `server_stub`. The client stub marshals the data with pickle library and calls the server api of the same name. The server API interfaces with the Memory and performs read write operation. The return data is marshalled by the server before sending and the client unmarshals it before sending it to the higher layers.

All the APIs implemented on the `client_stub.py` are written within the try catch method. Therefore, any network disruption on result in an error message and the client process will terminate. The failure can be removed by removing the `quit()` API from the catch section and add a `return -1` command to indicate error. However, we observed that in the latter case some of the operation if failed in the middle may corrupt the system.

Since the memory system is a simulated model. If the server crashes, the file system data loses.

Please see the code comments for more information.

### Test results:

We have found that the network file system is slower than the local version. This is predictable since the network apis are not involved in the latter one. The comparison result is shown below (results are performed on a 6 byte data):

**Write** time: 33x slower

**Read** time: 23x slower

**Move** file: 34x slower

**Delete** file: 40x slower