Remote Procedure Call and Code Execution

The goal of this recitation is to help you prepare for the final homework assignment, in which you will build a work scheduling framework involving remote execution of code. In this recitation you will build a client/server system in which a server respond to requests of a remote client and potentially execute code on behalf of a remote client.

Imagine you have several files stored on a remote server, and you want to perform operations on the fiels. You should be able to download a file if you would like to, but it will also be great if you can instruct the servers to execute some code on your behalf on the file, and send you back the result as opposed to having to download the files. Hence, the server should be able to respond to two types of requests: a fetch where a client downloads the file, and an exec, where the client sends a piece of code over in the form of a Function object to be performed on the file.

To successfully implement a remote execution framework, remote worker servers must be able to execute code on behalf of clients. This is hard because the server program does not know about the code it will execute in its JVM. Simply sending the task over the socket will result in a ClassNotFoundException. Instead, you will need to send the actual java bytecodes over instead of an object, and instantiate the object from the class file using reflection on the server. We have taken care of most of the complicated logic involved in doing this, in `PickledClass` under `util`.

Remote Procedure Call

It's often normal to send over an opcode at the beginning of any RPC communication. The types of message we permit are written out in the enum MessageType. Implementall functionality involving the Fetch procedure following the instruction provided in the code file. You should try to download the art of war under the folder assets to the folder local.

Remote code execution

Now continue and finish the Exec procedure by following the instruction provided in the code file. You should perform a word count operation, implemented as a function from FileInputStream to integer, on Kafka's Metamorphosis under the assets folder.

Putting it together

You can now start up the Server with a single command line argument of the port number, and run the Client program with the appropriate hostname and port number. You can try to talk to a Server written by your friend, or have your TA run a Server instance for you to talk to. Considering that this is a toy project, no security measure is in place, so be nice to your peers and TA when doing this. We will know who you are if you do bad things.

It is worth noting that in practice, most of this is done by using the Java Remote Method Invocation(RMI) framework, which, underneath the hood, does similiar work to what you just did, but provides a nicer abstraciton. You are welcome to learn and use RMI in homework 6.