Summary

I think this assignment builds a distributed key-value storage system with the help of RPC. RPC has a lot of advantages over transport layer protocols, e.g. TCP and UDP. RPC takes care of serialization, threads and it makes the remote program run as if it is on local machine. RPC is a very powerful inter-process communication technique that help me to dive into the basics of distributed system.

Technical impression

As RPC takes care of the transport layer and makes the remote program transparent to local machine, we can just invoke the remote storage methods, e.g. PUT, GET, DELETE.

We need to implement multi-threading server, so it's vital to take care of threads and mutual exclusion. Java RMI already supports multi-threading, and I used Thread.currentThread() to log current server thread. In order to protect data integrity, I used synchronized modifier on PUT, GET and DELETE methods in Server codes. The synchronized modifier in method level actually locks the current instance (of Server) and enables atomic operations on the HashMap.