

Disciplina: INF16179 - Sistemas Distribuídos

Alunos: Thiago da Silva Meireles de Souza / Marcio Merçon de Vargas / Maria Morena Muzini Esperidião

Laboratório III

Metodologia

Este trabalho realiza a implementação de um “minerador de criptomoedas”, no modelo cliente/servidor baseado em RPC. A implementação foi realizada em Python/gRPC.

Conforme especificado, o trabalho foi dividido em duas partes:

- O **Servidor**, que fica em loop, atendendo as solicitações na porta TCP/IP 8080 e só será interrompido com um <ctrl+c>. Durante a execução ele manterá uma tabela com as seguintes informações: Identificador inteiro da transação, que é iniciado com o valor 0 (zero); Valor inteiro referente ao desafio atual, gerado aleatoriamente entre 0 (zero) e 6 (seis); String com o valor da solução; Valor inteiro que representa o ID do cliente que solucionou o desafio. Também foram implementadas as seguintes chamadas de procedimento remoto aos clientes: `getTransactionID()`, para retornar o ID da transação atual; `getChallenge()`, para retornar o valor do desafio associado a um determinado ID; `getTransactionStatus()`, para informar se um determinado desafio (ID) foi solucionado (0 - resolvido / 1 - não resolvido); `submitChallenge()`, para enviar a solução para a função de hashing que resolve o desafio e retornar 1 se a solução foi válida, 0 se for inválida e 2 se o desafio (ID) já foi solucionado; `getWinner()`, para retornar o vencedor de um determinado desafio (ID); e `getSolution()`, retorna uma estrutura com o status, a solução e o desafio de um determinado ID.
- O Cliente, que permite informar o endereço do servidor RPC e disponibiliza as seguintes opções em formato de menu: `getTransactionID`, para solicitar o ID do desafio atual; `getChallenge`, para solicitar o valor do desafio de um determinado ID; `getTransactionStatus`, para solicitar o estado do desafio de um determinado ID; `getWinner`, solicita o ID do cliente vencedor de um determinado desafio (ID); `GetSolution`, solicita a solução de um determinado desafio (ID); e `Mine`, para buscar o desafio atual (ID), buscar o valor de um determinado desafio, buscar a solução para o desafio proposto, imprimir a solução encontrada, submeter a solução encontrada ao servidor e aguardar a resposta, e imprimir a resposta do servidor após decodificá-la.

Testes e Resultados obtidos

Os testes foram executados localmente, abrindo uma instância do servidor em um terminal e várias instâncias clientes, que se conectam ao servidor e executam as chamadas de procedimento remota desenvolvidas e descritas acima.

Os resultados ficaram dentro do que se esperava. Ou seja, o servidor ao ser iniciado gera o desafio e deixa disponíveis as 6 (seis) chamadas RPC que foram desenvolvidas e descritas e fica em loop atendendo as conexões. O cliente, por sua vez, se comunica com o servidor através do endereço fornecido, executa a interação com o servidor pelas chamadas disponíveis e soluciona o desafio criptográfico lançado pelo servidor.