Lab01 – Comparando aplicações distribuídas usando RPC e Sockets TCP/UDP

PSPD – Programação para Sistemas Paralelos e Distribuídos Turma A
(Prof. Fernando W Cruz)

Alunos: Vinícius Vieira de Souza - 170115500
Antonio Ruan Moura Barreto - 180030272

Introdução

Este experimento tem como objetivo a construção de aplicações distribuídas incluindo passagem de parâmetros, envolvendo módulos cliente e servidor usando RPC e Sockets TCP/UDP.

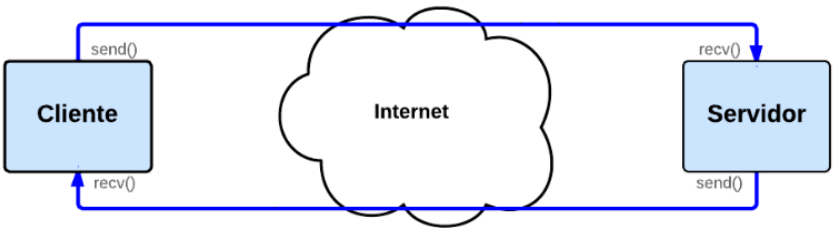
O problema consiste em criar um programa em que o módulo do cliente será responsável por inicializar um vetor de 500000 posições calculadas a partir das seguintes fórmulas:

```
v[i] = (i - (f_aleat(0-X)/2)) ** 2; onde f_aleat(0-X)
é uma função que retorna um valor aleatório entre 0 e X,
sendo X o tamanho definido para o vetor
```

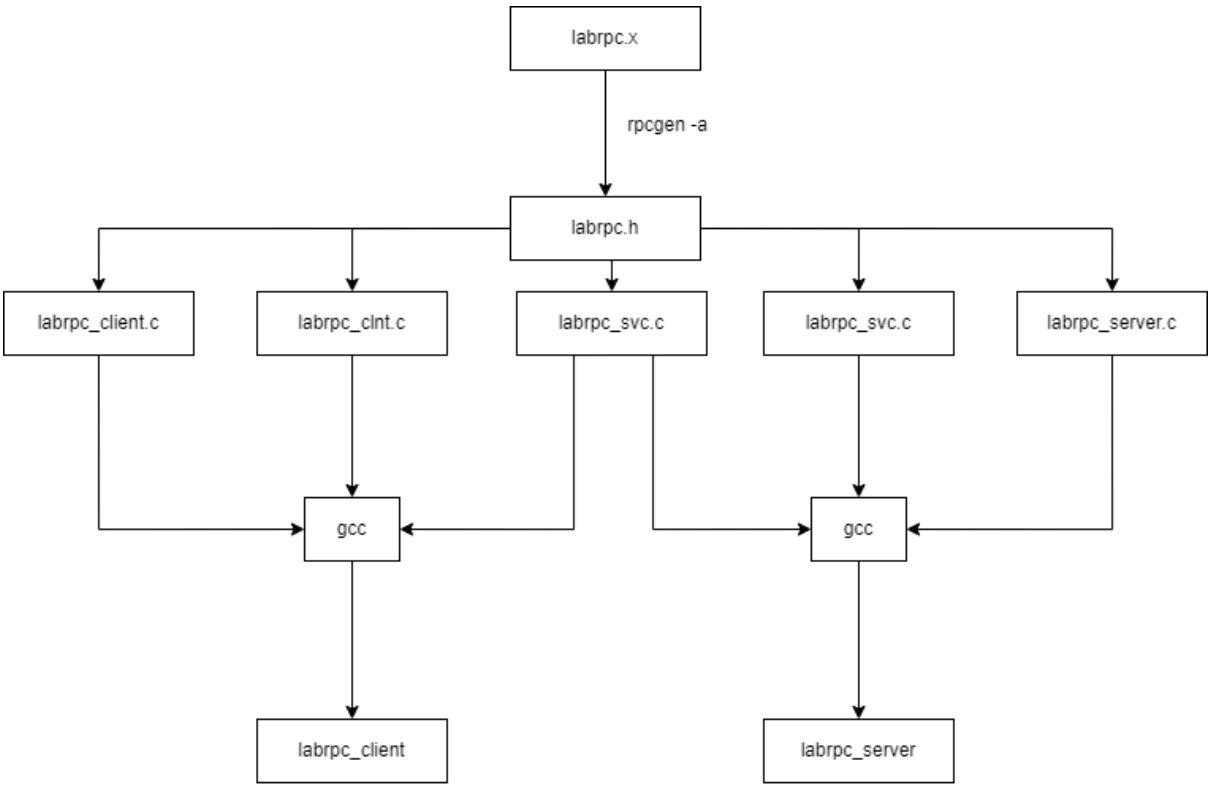
```
v[i] = raiz_quadrada(v[i]);
```

Seguindo essas propostas para a entrega 1 foi gerado um programa com dois módulos: *tcpclien*t e *tcpserver* para a implementação da solução com uso do Socket. E outra solução para a implementação com RPC gerando uma série de arquivos a partir do *labrpc.x*

Segue abaixo um diagrama simplificado das arquiteturas implementadas para as duas soluções:
Arquitetura para Socket:



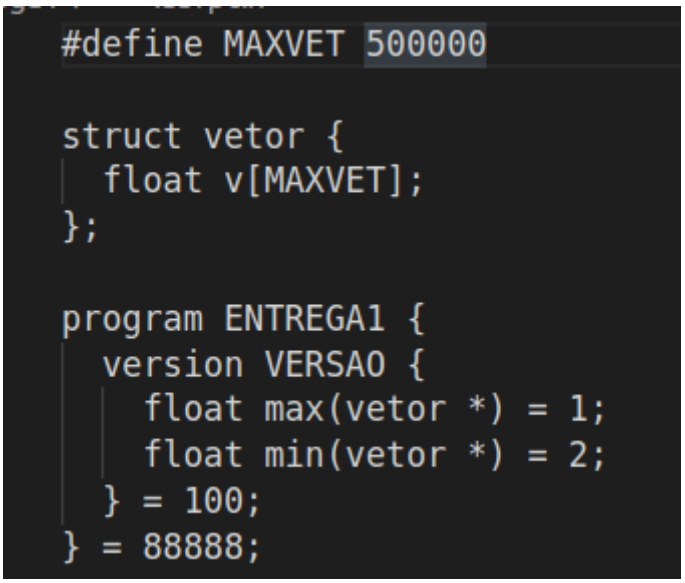
Arquitetura para RPC:



Solução:

Socket: Foi adotado uma comunicação por protocolo TCP entre cliente e servidor. O cliente é responsável por inicializar o vetor com seus valores de forma aleatória conforme solicitado e enviá-los para o tcpserver onde serão descobertos os valores máximos e mínimos (através de duas funções max() e min()) do vetor. Ao realizar esses cálculos o servidor apenas os envia de volta para o cliente onde serão impressos no console.

RPC: Foi criado um arquivo IDF *labrpc.x* que consisti contendo um define para o tamanho do vetor, uma struct utilizada para o próprio vetor de 500000 posições. Assim como a estrutrua do program com suas versoes e o numero de identificação das funções. Veja a imagem abaixo para mais detalhes:



```
#define MAXVET 500000

struct vetor {
  float v[MAXVET];
};

program ENTREGA1 {
  version VERSAO {
    float max(vetor *) = 1;
    float min(vetor *) = 2;
  } = 100;
} = 88888;
```

Detalhes Entrega 1

Socket: Dificuldades. Enviar muitos dados através de um pacote TCP do cliente para o servidor e vice-versa. Limitações: Devido a problemas que a dupla acredita ser de rede. Os valores nem sempre são transmitidos corretamente.

RPC: Dificuldades: A dupla não conseguiu rodar o programa em duas máquinas diferentes e nem através de um container do docker. Portanto essa entrega se limita apenas ao uso por loopback. Ou seja utilizando a própria máquina como servidor e cliente.