

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

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

A) Objetivo: O objetivo desse experimento é que o aluno compreenda as características inerentes à construção de aplicações distribuídas, incluindo passagem de parâmetros, envolvendo módulos cliente e servidor usando RPC e Sockets TCP/UDP.

B) Detalhes do laboratório

Visão geral da aplicação - No ambiente local (Linux), construir uma pequena aplicação distribuída, que descobre o maior e o menor valor em um vetor de 500 mil posições, considerando os seguintes passos:

1. O vetor deve ser gerado no cliente, deve ser de números do tipo *float*, e inicializado pelo uso da fórmula a seguir:

$v[i] = (i - (f_aleat(0-X)/2)) ** 2;$ 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

2. Em seguida, a aplicação do lado cliente deve calcular cada posição do vetor, de acordo com a seguinte fórmula:

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

3. A aplicação deve ser organizada, de modo a encontrar o maior e o menor valor do vetor, envolvendo cliente (inicialização do vetor) e servidor ou *worker* (que faz a descoberta do maior e menor valor do vetor). Ao terminar o trabalho o *worker* deve repassar esses números de volta para o cliente, de modo que esse cliente possa imprimir o resultado para o usuário.
4. Ou seja, o cliente apresenta para o usuário solicitante, o resultado da operação. Perceber que a solicitação foi realizada de maneira distribuída, ou seja, envolvendo unidades de processamento e espaços de memória distintos.

Entrega 1 - Os alunos devem gerar a aplicação especificada, em duas versões: (i) uma versão no qual o cliente e o servidor dialogam via socket TCP ou UDP; (ii) uma versão no qual o cliente e o servidor interagem via RPC.

Obs1.: Para cada uma das versões, os alunos devem entregar os códigos devidamente documentados, em pastas separadas.

Obs2.: No relatório de entrega, os alunos devem comentar as dificuldades e facilidades encontradas com essas duas soluções, comparando as possibilidades.

Entrega 2 - Os alunos devem implementar o problema descrito usando RPC, de modo que o cliente faça a solicitação envolvendo pelo menos dois servidores (*workers*), a fim de reduzir o tempo de resposta da aplicação. Nesse caso, é importante fazer testes e mostrar a diferença de performance entre a versão tradicional e a versão desenhada.

C) Questões de ordem

- A atividade pode ser feita por grupos de até 2 alunos.
- Caso solicitados, os alunos devem estar preparados (slides e apresentação do código) para demonstração da solução em sala de aula, conforme definido pelo professor em data oportuna.
- A entrega desse lab será feita pelo envio de um arquivo zipado no ambiente Moodle da disciplina disponível em <http://aprender3.unb.br>. O arquivo zipado deve conter: (i) os arquivos .c das aplicações criadas separadas em pastas, uma para cada uma das entregas definidas, (ii) instruções de instalação e uso da aplicação criada, e (iii) um relatório cujo conteúdo está descrito mais adiante.
- Os códigos de cada uma das entregas, devidamente comentados e identados, e em pastas específicas.
- Deve ser entregue também um relatório (formato pdf) a ser entregue junto com o experimento, contendo os seguintes pontos:
 - a) Título do experimento, dados da disciplina e identificação do(s) aluno(s) participantes
 - b) Introdução – pequena descrição do problema e da arquitetura RPC
 - c) Descrição da solução, visão geral - mostrando a ideia da solução adotada (incluindo o arquivo de definição de interface (IDL) no caso da versão RPC)
 - d) Descrição dos detalhes da Entrega 1 – apontar problemas e soluções encontradas até chegar à versão final do problema proposto. Incluir aqui eventuais limitações dos códigos entregues.
 - e) Descrição dos detalhes da Entrega 2 - apontar problemas e soluções encontradas até chegar à versão final do problema proposto. Incluir aqui eventuais limitações dos códigos entregues. Importante: descrever quais foram os mecanismos adotados de modo a coordenar os trabalhos dos *workers* e integrar os resultados numa solução única.
 - f) Descrição da versão-bônus, caso tenha sido implementada. Nesse caso, descrever o que foi modificado no código para atender essa demanda. Importante também relatar os problemas e as soluções encontradas para tratar o vetor. Mostrar também uma tabela apontando o desempenho da aplicação, em função da quantidade de *workers* testada.
 - g) Opinião geral e nota - aqui, cada aluno do grupo deve manifestar sua opinião sobre o laboratório, apontando eventuais aprendizados novos, as dificuldades encontradas e possíveis limitações percebidas no experimento. Deve ainda deixar claro como foi a sua participação no Laboratório (onde colaborou mais) e atribuir uma nota, no intervalo de zero a dez, em função da sua participação e aprendizado.