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

Lab01 – Construindo aplicações distribuídas usando RPC

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 a arquitetura RPC.

B) Detalhes do laboratório

Versão com um 1 cliente e um *worker* (servidor RPC) - No ambiente local (Linux), construir uma pequena aplicação distribuída (linguagem C), que descobre o maior e o menor valor em um vetor de 10 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 – tamanho_do_vetor/2) ** 2;
```

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 que a função que encontra o maior e o menor valor de um vetor esteja instanciada numa função remota (*worker*), num servidor RPC. Essa função, portanto, deve analisar o vetor, encontrar o maior e o menor valor e repassar esses números de volta para o cliente, de modo que ele 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.

Versão com dois *workers* (função instanciada em dois servidores RPC) - Nesse versão, a aplicação deve ser construída, de modo que o cliente faça a solicitação envolvendo pelo menos dois servidores (*workers*).

Versão-bônus - Essa versão é opcional. Pontos extras para os alunos que construírem uma versão da aplicação com vários *workers*, de modo a encontrar o maior e o menor valor, considerando um vetor de 1.000.000.000 de posições de números do tipo *float*.

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 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 versões comentadas, *(ii)* instruções de instalação e uso da aplicação criada, e *(iii)* um relatório cujo conteúdo está descrito mais adiante.

- A versão-bônus é opcional, mas sugere-se a sua implementação. Nesse caso, observar o seguinte:

    a. Se os códigos dessa solução forem diferentes em relação às soluções anteriores, os respectivos códigos devem ser incluídos na entrega. Nesse caso, é permitida a alteração completa do código, inclusive a lógica e os parâmetros passados entre os lados cliente e servidor, de modo a atender o problema.

    b. Nessa solução, fazer testes com vários *workers*, de modo a reduzir o tempo de processamento do problema até encontrar uma configuração de *workers* cujo tempo de resposta seja aceitável para essa aplicação.

- Os códigos de cada uma das versões descritas devem ser entregues, 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, mostrando a ideia da solução adotada, incluindo o arquivo de definição de interface (IDL) desenhado para resolver o problema

    d) Descrição da solução com um *worker* – apontar problemas e soluções encontradas até chegar à versão final do problema proposto. Incluir aqui eventuais limitações dos códigos entregues.

    e) Descrição da solução com dois *workers* - 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.