Universidade Federal
do Espírito Santo

CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA

Processamento Paralelo e Distribuído – Turmas 01 e 02 (EARTE) – 2021/1
Prof. Rodolfo da Silva Villaça – rodolfo.villaca@ufes.br
Trabalho T1.2 – Uso de Middleware RPC/RMI

Objetivo:
Experimentar a implementação de sistemas cliente/servidor por meio de middleware RPC e/ou RMI. Juntar os conceitos de paralelismo e middleware em um mesmo sistema distribuído. Comparar o tempo de execução com diferentes quantidades de processos ou *threads*. Avaliar a necessidade de controle de concorrência quando há mútiplos clientes e um único servidor.

Instruções:

Parte I

1. O servidor deverá implementar uma tabela hash de inteiros com tamanho M (M>1000000). O servidor deverá disponibilizar pelo menos 2 interfaces de acesso remoto: put() e get();
2. No servidor, o método put() serve para inserir um inteiro v na hash table a partir de uma chave k que pode ser gerada no cliente ou no servidor (a critério do grupo). O método get() deve receber uma chave k como entrada e retornar o inteiro v associado à chave (ou NULL, caso não haja valor associado à k);
3. O cliente deverá gerar m (m<M) valores inteiros aleatórios v e inserir esses números na tabela hash a partir da função put(). Em seguida, a partir da chave k gerada para cada v, recuperar os m valores por meio da função get(). Contar o tempo para todo esse processo de inserção/recuperação dos valores;
4. put() e get() devem ser implementados como chamadas de procedimentos ou métodos remotos, RPC ou RMI;

Parte II
1. Paralelizar os clientes em i=[2, 4, 8] instâncias simultâneas, cada uma deverá repetir o processo da Parte I com m/i valores. Todos os clientes concorrem para o mesmo servidor, as mesmas funções put/get e a mesma tabela hash;
2. Contar o tempo requerido para inserção e recuperação dos m valores v na tabela hash com 2, 4 e 8 instâncias cliente em paralelo;
3. Ao final, responder (e justificar) à seguinte pergunta: é necessário implementar algum controle de concorrência no acesso aos métodos e à tabela hash por parte dos diferentes clientes?

Requisitos:
1. O trabalho pode ser feito em grupos de 2 ou 3 alunos: não serão aceitos trabalhos individuais ou em grupos de mais de 3 alunos;
2. Os grupos poderão implementar os trabalhos usando qualquer uma dentre as três linguagens de programação: C, Java ou Python. Os grupos também poderão usar quaisquer algoritmos ou bibliotecas para implementar tanto as chamadas de RPC/RMI quanto as tabelas hash. Outros métodos podem ser disponibilizados para acesso remoto, a critério do grupo;
3. A submissão deverá ser feita por meio de um *link* com a disponibilização do código no Github, em modo de acesso público ou, minimamente, que meu e-mail rodolfo.villaca@ufes.br tenha direito de acesso ao código;
4. A documentação para uso/teste do seu programa deverá ser feita na própria página do Github através do arquivo README[1];
5. Avaliação: Adequação aos Requisitos (30%), Legibilidade do Código (30%), Documentação (40%), Extra (10%);
6. Data de Entrega: 27/07/2021 pela Sala Virtual da disciplina no Google Classroom;

Extra:
1. Valendo 10% a mais na nota para os grupos que fizerem cliente e servidor com linguagens distintas. Neste caso sugiro usar a biblioteca grpc[2]

Bom trabalho!

1 https://guides.github.com/features/wikis/
2 https://grpc.io/