UNIVERSIDADE FEDERAL ALFENAS (UNIFAL)
Bacharelado em Ciência da Computação

Disciplina	Método de entrega	Data de entrega
DCE540 - Computação Paralela e Distribuída	Moodle da disciplina	27/01/2022 às 23h59

Professor
Iago Augusto de Carvalho (iago.carvalho@unifal-mg.edu.br)

Exercício 01

Cada aluno deverá submeter um único arquivo .pdf com a resolução da prova.
Este exercício pode ser realizado de duas maneiras:

- Com papel e caneta, sendo posteriormente escaneado e enviado
- Digitado em algum editor de texto, e.g., Word ou LaTeX

O exercício deverá ser entregue **individualmente** no Moodle da disciplina até a data limite.

- Atrasos não serão tolerados

Os códigos aqui referenciados estão postados no Github, junto à descrição deste exercício

Descrição do exercício

O objetivo é que todos analizem a maneira que uma chamada a procedimento remoto (RPC) é realizada. Para isto, utilizaremos alguns códigos desenvolvidos na linguagem *Python*, utilizando a biblioteca *RPyC*. Esta biblioteca pode ser instalada utilizando o *python pip*, executando

```
pip install rpyc
```

Para isto, iremos utilizar um modelo orientado a serviços. O seguinte cógido abaixo, referido no repositório como *servidor.py*, implementa um template para chamadas remotas de procedimento

```python
import rpyc
class MyService(rpyc.Service):
    def on_connect(self, conn):
        # codigo que eh executado quando uma conexao eh iniciada, caso seja necessario
        pass
    def on_disconnect(self, conn):
        #  codigo que eh executado quando uma conexao eh finalizada, caso seja necessario
        pass
    def exposed_get_answer(self): # este eh um metodo exposto
        return 42
    exposed_the_real_answer_though = 43 # este eh um atributo exposto
    def get_question(self): # este metodo nao eh exposto
        return "Qual era a cor do cavalo branco de Napoleao?"
#Para iniciar o servidor
if __name__ == "__main__":
    from rpyc.utils.server import ThreadedServer
    t = ThreadedServer(MyService, port=18861)
    t.start()
```

Devem existir os dois métodos para conectar e desconectar uma conexão e os outros métodos podem ser definidos livremente por você. Neste caso, você pode escolher os atributos que vão ser expostos para outros processos: se o nome começa com *exposed_*, ele poderá ser acessado remotamente, senão ele será acessível somente localmente.

O programa cliente abaixo, referido no repositório como *cliente1.py*, se conecta com o servidor e executa algumas instruções remotamente.

```
import rpyc
import sys

if len(sys.argv) < 2:
   exit("Usage {} SERVER".format(sys.argv[0]))

server = sys.argv[1]

conn = rpyc.connect(server,18861)

print(conn.root)
print(conn.root.get_answer())
print(conn.root.the_real_answer_though)
```

Execute o servidor em uma máquina e execute o cliente na mesma máquina. Para isso basta executar o programa python do servidor sem argumentos e o do cliente com o argumento *localhost*. Após, responda:

Exercício 1

Explique o que foi impresso no cliente

Exercício 2

Execute o código abaixo, referido no repositório como *cliente2.py*, junto com o servidor em uma mesma máquina.

```
import rpyc
import sys
import os

if len(sys.argv) < 2:
   exit("Usage {} SERVER".format(sys.argv[0]))

server = sys.argv[1]

conn = rpyc.connect(server,18861)

print(conn.get_question)
```

Agora, explique o que ocorreu quando você executou este código. Houve algum erro? Como você poderia corrigir este possível erro?

Exercício 3

Escreva um programa cliente que cria um vetor de n posições, onde n é definido pelo usuário, com elementos variando de 0 a $n-1$. Este procedimento chama um procedimento no servidor que soma os elementos do vetor e retorna o resultado da soma. O programa cliente deve imprimir o valor de soma.

Exiba seu código no arquivo .pdf onde você entregará seu exercício.

Exercício 4

Inclua no seu código do cliente e do servidor as instruções abaixo para medir o tempo de execução:

```
start = time.time()
end = time.time()
print(end-start)
```

Execute o cliente e servidor na mesma máquina para um vetor de 10000 posições. Assim, descreva o tempo necessário, tanto do cliente quanto do servidor, para realizar estas operações.