

Sistemas Distribuídos — Trabalho Prático

Cloud Computing

Engenharia Informática
Universidade do Minho

2023/2024

Informações gerais

- Cada grupo deve ser constituído por 4 elementos, obrigatoriamente inscritos no *eLearning* até 31 de outubro de 2023.
- Deve ser entregue o código fonte e um relatório de até 6 páginas (A4, 11pt) no formato PDF.
- O trabalho deve ser entregue até às 23:59 do dia 30 de dezembro de 2023 no *eLearning*.
- A apresentação do trabalho será em dia e hora a agendar posteriormente por cada grupo.
- Cada grupo deve organizar a sua apresentação de forma a que todos os elementos participem espontaneamente.

Resumo

Neste projeto pede-se a implementação de um serviço de *cloud computing* com funcionalidade *Function-as-a-Service (FaaS)*¹: Usa-se um cliente numa máquina local para enviar o código de uma tarefa de computação a ser executado num servidor, logo que haja disponibilidade, recebendo de volta o resultado. Assume-se também que o fator limitante nos servidores é a memória disponível.

O código da tarefa a executar bem como o seu resultado são simples `byte[]`. A execução é feita com a função `JobFunction.execute()` fornecido no `sd23.jar`, tal como demonstrado em `Example.java`.

A essência do serviço é a capacidade de manter uma fila de espera de tarefas para executar e de garantir que faz uma boa utilização dos recursos disponíveis (memória), sem que os pedidos em execução concorrente ultrapassem o máximo disponível.

Funcionalidade básica (até 12 valores)

Este serviço deverá suportar a seguinte funcionalidade:

1. Autenticação e registo de utilizador, dado o seu nome e palavra-passe. Sempre que um utilizador desejar interagir com o serviço deverá estabelecer uma conexão e ser autenticado pelo servidor.
2. Pedido de execução, enviando o código da tarefa e indicando a quantidade de memória necessária. Depois de executada, devolve o resultado ou a indicação do número e mensagem de erro.

¹https://en.wikipedia.org/wiki/Function_as_a_service

3. Consulta do estado atual de ocupação do serviço (memória disponível) e da sua fila de espera (número de tarefas pendentes).

Funcionalidade avançada (mais 4 valores)

Como funcionalidades avançadas pretende-se que:

1. O cliente permita submeter novos pedidos, de consulta e execução de tarefas, sem ter recebido as respostas dos anteriores. O cliente deve continuar a ter acesso às respostas e saber a que pedido se referem.
2. Garanta uma ordem de execução das tarefas que impeça uma tarefa de ficar para sempre à espera, mesmo que haja sempre novos pedidos concorrentes.

Implementação distribuída (mais 4 valores)

É valorizada uma implementação distribuída: um serviço que permita gerir um parque de vários servidores para fazer a execução de tarefas, cada um com uma configuração própria de memória. Neste caso:

1. O serviço usado pelos clientes para enviar tarefas a executar deve ser assegurado por apenas uma máquina, que gere a fila de espera e atribui as tarefas a servidores de execução de tarefas. Neste caso, a máquina que gere a fila não deve executar tarefas.
2. Cada máquina disponível como trabalhadora corre um programa que se limita a executar as tarefas, não participando na gestão da fila de espera nem na interação com os clientes.

Para implementar esta funcionalidade comece por planear a arquitetura a utilizar e o protocolo de comunicação entre a fila e os trabalhadores.

Programas do serviço

Os programas do serviço devem ser implementado em Java, usando *threads* e *sockets* TCP, mantendo em memória a informação relevante para suportar as funcionalidades acima descritas, receber conexões e input dos clientes, bem como fazer chegar a estas a informação pretendida.

Biblioteca do cliente

Deverá ser disponibilizada uma biblioteca (conjunto de classes e interfaces) que proporcione o acesso à funcionalidade do serviço descrita acima. Esta biblioteca deve ser independente da interface com o utilizador e deverá ser escrita em Java usando *threads* e *sockets* TCP.

Interface do utilizador

Finalmente, deverá ser disponibilizada uma interface com o utilizador que permita interagir com o serviço através da biblioteca cliente. Esta interface deverá também ser escrita em Java e tem como único objetivo a interação com o serviço para testes e durante a apresentação do trabalho. Esta interface deve ser capaz de ler as tarefas de ficheiros e depositar os resultados também em ficheiros.

Requisitos

Na implementação devem ser satisfeitos os seguintes requisitos:

- Deve haver apenas um única conexão entre cada duas máquinas envolvidas, por onde pedidos, respostas e notificações devem passar.
- O protocolo de comunicação deverá ser num formato binário, através de código desenvolvido no trabalho, podendo recorrer apenas a `Data[Input|Output]Stream`.
- Para o serviço não ficar vulnerável a clientes lentos, não deverá ter *threads* do servidor a escrever em mais do que um socket, devendo as escritas ser feitas por threads associadas a esse socket.

Valorizam-se estratégias que diminuam a contenção e minimizem o número de threads acordadas.