Atividade extraclasse

A) Objetivo: O desenvolvimento e disponibilização de aplicações distribuídas com uso de computação em nuvem e de contêiners tem ganhado destaque, motivando o uso de plataformas como o Kubernetes (https://kubernetes.io/), que tem sido visto como uma espécie popular de sistema operacional no mundo Cloud Native (https://www.cncf.io/). O objetivo dessa especificação é promover um contexto onde os alunos possam *(i)* experimentar a construção de aplicações baseadas em microserviços baseados no *framework* gRPC (https://grpc.io/), e, ao mesmo tempo, *(ii)* explorar tecnologia para *deploy* de aplicações baseadas em contêiners kubernetes.

B) Metodologia, especificação e detalhamento de atividades

Para o alcance do objetivo desejado, a metodologia envolve *(i)* estudo do gRPC e criação de uma aplicação distribuída, e *(ii)* estudo e montagem do Kubernetes (também grafado como K8S) usando uma versão simples não-clusterizada. As subseções a seguir detalham cada um desses itens.

B.1) Conhecendo o gRPC

Inicialmente os alunos devem estudar e documentar sobre os componentes do *framework* gRPC (protobuf, http/2), usando como base o site https://grpc.io/ e demonstrar, com exemplos de aplicações, os tipos de comunicação suportados pelo gRPC (*unary calls*, *server-streaming calls*, *client-streaming calls* e *bidirecional streaming calls*). Na entrega, apresentar os códigos dos testes realizados, o arquivo protobuf usado e uma conclusão de situações em que cada um desses tipos de comunicação pode ser utilizado.

B.2) Construindo uma aplicação cliente/servidor com gRPC

Em seguida, devem construir uma aplicação distribuída, composta por três módulos (P, A e B), conforme ilustrado na Figura 1. Nessa figura, o módulo (P) deve conter um *gRPC Stub* integrado a um *web server* para receber requisições remotas dos clientes, preferencialmente por uma interface Rest-API. Neste contexto, *gRPC Server (A)* e *gRPC Server (B)* são os componentes distribuídos que colaboram com *gRPC Stub (P)* de modo a atender as demandas dos clientes remotos (vide módulo *web client* da Figura 1). Os diálogos e a arquitetura da aplicação distribuída a ser entregue está apresentada na parte direita da Figura 1, como um *backend gRPC*.

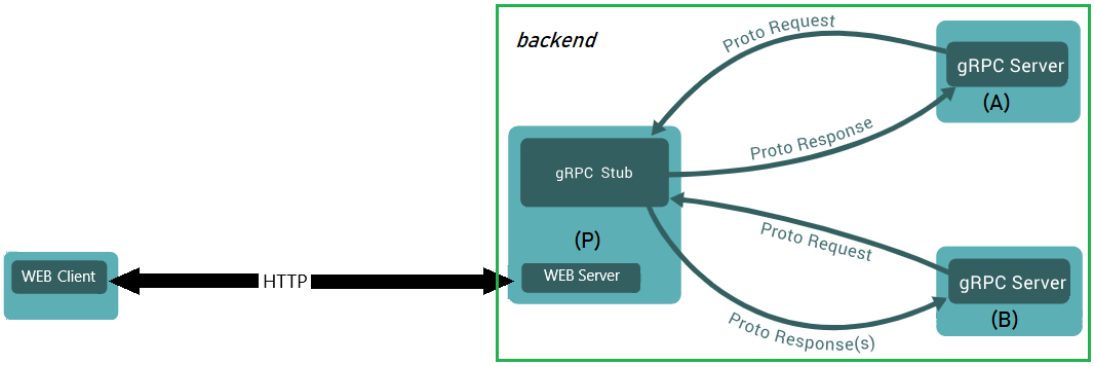

Figura 1 – Arquitetura da aplicação distribuída – baseada em microserviços

A aplicação distribuída é de livre escolha de cada grupo, mas a arquitetura proposta na Figura 1 deve ser mantida (um *frontend* para requisições HTTP e um *backend* com os microserviços ilustrados). Detalhes adicionais sobre os requisitos:

- Os microserviços gRPC (A) e (B) devem ser implementados em linguagem de programação distinta da que for utilizada para construção do *gRPC Stub* (P).
- Os microserviços (A) e (B) devem ser diferentes entre si e serem consultados por (P). Obs.: Opcionalmente, é aceitável que além de atender as requisições de (P), o gRPC Server (A) possa consultar o gRPC Server (B) e vice-versa.
- Os módulos que compõem o *backend* (P, A e B) devem prover um serviço colaborativo, onde cada módulo seja diferente do outro, mas, em conjunto todos consigam atender as demandas do usuário. Por exemplo, uma pequena aplicação que mantenha uma base de informações cujas partes complementares estão distribuídas entre os módulos gRPC Server (A) e (B).
- Em (P) a aplicação deve conter uma interface web que permita a oferta dos serviços da aplicação escolhida para usuários da Internet. Para isso, deve-se instalar um *web server* e deve ser montada uma API que possa traduzir as requisições e respostas que transitam entre o cliente web (*browser*) e os microserviços gRPC.

B.3) Preparando a infraestrutura para disponibilização do microserviço

Inicialmente os alunos devem realizar um estudo sobre Cloud Native e kubernetes para compreender a arquitetura do Kubernetes e o desenvolvimento de aplicações distribuídas utilizando contêiners (vide livro sobre Kubernetes/Cloud Native no site Moodle da disciplina).

Em seguida, deve-se montar uma infraestrutura composta por containers usando uma versão simples do kubernetes que funciona em *host* único conhecida como minikube (https://minikube.sigs.k8s.io/docs/) para comportar a aplicação distribuída.

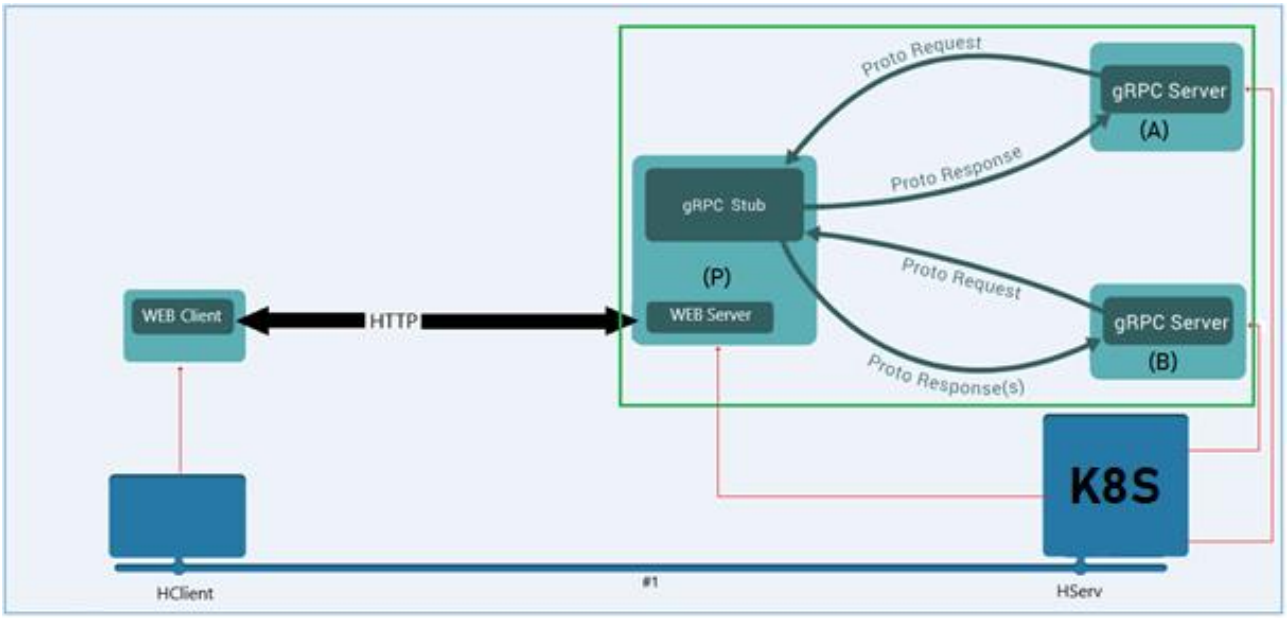

Figura 2 – Distribuição dos módulos da aplicação distribuída no Kubernetes

Alguns detalhes importantes:

- A aplicação distribuída descrita em B.2 deve ser instanciada em um *host* Linux, denominado de HServ com kubernetes, como apresentado na Figura 2. Por sua vez, HClient é um desktop com um *browser* (*web client*) a partir de onde a aplicação distribuída é acessada via interface HTTP tradicional.
- O container P instanciado em HServ deve comportar a API-Gateway da aplicação (cliente gRPC, web server e demais funcionalidades necessárias), de modo a repassar as requisições HTTP para o *backend* e vice-versa. Deve-se organizar esse contêiner de modo que o mesmo possa participar da rede externa (que processa requisições HTTP tradicionais) e da rede interna (que processa requisições HTTP/2) onde os microserviços estão instanciados.
- As demais partes da aplicação devem ser instanciadas em dois outros contêiners A e B em HServ de modo a proverem o serviço gRPC e atenderem requisições solicitadas por P.

C) Questões de ordem

- O experimento pode ser feito por grupos de 4 ou 5 alunos. Nesse caso, basta que um dos alunos faça a postagem das entregas no Moodle da disciplina.
- A entrega é composta por *(i)* um relatório, cuja estrutura e conteúdo está descrito a seguir, *(ii)* informações (arquivos) com a configuração da aplicação distribuída e da versão Kubernetes instalada, incluindo informações necessárias para replicação do laboratório pelo professor, *(iii)* um vídeo gravado pelos membros participantes, com apresentação do projeto. Nesse caso, considerar uma média de 4 minutos por aluno para que possam demonstrar como participaram e conhecimentos adquiridos *(iv)* uma documentação sobre os códigos e configurações entregues (se houver alguma configuração extra, incluir à parte) – essa documentação deve fazer parte do relatório a ser entregue.
- O relatório a ser entregue deve conter os seguintes pontos:
    - Título da atividade extraclasse, dados do curso, da disciplina/turma e identificação dos alunos participantes.
    - Introdução – pequena descrição da solicitação feita e uma visão geral sobre o conteúdo do relatório.
    - Uma seção sobre o *framework* gRPC – apresentando os elementos constituintes, especialmente protbuf e HTTP/2 e uma discussão sobre os tipos de comunicação providos pelo gRPC (com apresentação de testes sobre cada um desses tipos de comunicação).
    - Uma seção sobre a aplicação descrita em B.2 contendo *(i)* detalhes da aplicação escolhida, e *(ii)* um comparativo com aplicações tradicionais. Mais especificamente:
        - Sobre a aplicação gRPC, descrever a aplicação/funcionalidades planejadas e comentar sobre os passos para instanciação do serviço; descrever as dificuldades encontradas e a metodologia usada para finalização da aplicação. Inserir todas as informações relevantes para testes e avaliação do que foi feito.

- Sobre o comparativo solicitado, os alunos devem montar uma versão alternativa da aplicação descrita em B.2 baseada em Rest-API/JSON para os diálogos entre P, A e B. Com essas duas versões prontas, montar cenários de teste e medir *performance* (tempo de resposta) das duas versões sob as mesmas condições. Os resultados devem ser apresentados em tabela e deve ter texto conclusivo associado.

  Dica: escolher uma aplicação que envolva alguma demora no processamento (envolvendo requisições, fluxo de comunicação entre as partes, etc.) para facilitar as comparações.

- Uma seção sobre o kubernetes, com informações sobre arquitetura e funcionalidades do ambiente testado. Incluir informações sobre os arquivos de configuração utilizados nessa instalação básica e os passos para se chegar à configuração entregue. Incluir um descritivo de todos os comandos que foram utilizados, dificuldades encontradas e resultados alcançados para formação da estrutura de contêiners proposta no experimento.
- Conclusão – iniciar com um texto conclusivo sobre o experimento e subseções para que cada aluno possa manifestar sua opinião e aprendizados específicos sobre o que foi feito, além de uma nota de autoavaliação, em função do grau de envolvimento com essa atividade extraclasse.
- Apêndice/Anexo (seção opcional) – com eventuais informações não apresentadas anteriormente, tais como arquivos de definição de interface, comentários sobre os códigos construídos, instruções de execução, dentre outros.

- Além das entregas, os alunos devem estar preparados para uma apresentação em sala, conforme definido pelo professor em data oportuna (nesse caso, trazer slides para facilitar a palestra).