

GRPC (GOOGLE REMOTE PROCEDURE CALL) ESTRUTURA UNIVERSAL DE CÓDIGO ABERTO

GRPC (GOOGLE REMOTE PROCEDURE CALL) UNIVERSAL OPEN SOURCE FRAMEWORK

GRPC (LLAMADA A PROCEDIMIENTO REMOTO DE GOOGLE) MARCO UNIVERSAL DE CÓDIGO ABIERTO

Bruno Oliveira Bronze¹, Renata Mirella Farina², Fabiana Florian³

<https://doi.org/10.47820/recima21.v4i1.4492>

PUBLICADO: 11/2023

RESUMO

Este trabalho tem o objetivo de estudar o GRPC (*google remote procedure call*), trata-se de uma estrutura universal de código aberto muito usual para melhoria na comunicação em qualquer ambiente. Com essa ferramenta é possível conectar serviços e clientes de forma eficiente dentro e entre *datacenters* com suporte conectável para balanceamento de carga, rastreamento, verificação de integridade e autenticação, sendo aplicável também na computação distribuída para conectar dispositivos, aplicativos móveis e navegadores a serviços de *backend*. Foi realizada pesquisa bibliográfica a fim de justificar a relevância do tema, tanto para a sociedade como para profissionais do setor da engenharia da computação e programadores de computador, atualizando conhecimento da aplicação, instalação e uso desta tecnologia. Conclui-se que o uso dessa ferramenta por empresas, tanto de médio quanto grande porte, é de excelente viabilidade, pois possibilita a melhoria da comunicação entre diversos setores das empresas.

PALAVRAS-CHAVE: Código. Computador. Desempenho. Estrutura. GRPC. Programação.

ABSTRACT

This work aims to study the GRPC (google remote procedure call), it is a universal open source structure very common for improving communication in any environment. With this tool, it is possible to efficiently connect services and clients within and between datacenters with pluggable support for load balancing, tracing, health checking, and authentication, and is also applicable in distributed computing to connect devices, mobile applications, and browsers to backend services. Bibliographic research was carried out in order to justify the relevance of the theme, both for society and for professionals in the computer engineering sector and computer programmers, updating knowledge of the application, installation and use of this technology. It is concluded that the use of this tool by companies, both medium and large, is of excellent feasibility, as it enables the improvement of communication between various sectors of companies.

KEYWORDS: Code. Computer. Performance. Structure. GRPC. Schedule.

RESUMEN

Este trabajo tiene como objetivo estudiar el GRPC (Google remote procedure call), es una estructura universal de código abierto muy común para mejorar la comunicación en cualquier entorno. Con esta herramienta, es posible conectar de manera eficiente servicios y clientes dentro y entre centros de datos con soporte conectable para equilibrio de carga, rastreo, verificación de estado y autenticación, y también es aplicable en computación distribuida para conectar dispositivos, aplicaciones móviles y navegadores a servicios de backend. Se realizó una investigación bibliográfica con el fin de justificar la relevancia del tema, tanto para la sociedad como para los profesionales del sector de la ingeniería informática y programadores informáticos, actualizando los conocimientos sobre la aplicación,

¹ Graduando no Curso Bacharelado de Engenharia da Computação da Universidade de Araraquara- UNIARA. Araraquara-SP. E-mail: bobronze@uniara.edu.br

² Orientador(a) Docente do curso Engenharia da Computação da Universidade de Araraquara- UNIARA. Araraquara-SP. E-mail: rmfarina@uniara.edu.br

³ Coorientadora. Doutora em Alimentos e Nutrição. Docente do curso de Engenharia da Computação da Universidade de Araraquara - UNIARA. E-mail: fflorian_@uniara.com

instalación y uso de esta tecnología. Se concluye que el uso de esta herramienta por parte de las empresas, tanto medianas como grandes, es de excelente factibilidad, ya que permite mejorar la comunicación entre diversos sectores de las empresas.

PALABRAS CLAVE: Código. Ordenador. Rendimiento. Estructura. GRPC. Programación.

1 INTRODUÇÃO

Com o propósito de reduzir a latência e melhorar a comunicação nas aplicações atuais, o Google criou o sistema gRPC, que é uma implementação do modelo RPC.

O RPC (*Remote Procedures Call*) é a definição de um protocolo que executa procedimentos em outros computadores através da rede, isto é, que não é executado no local, mas o acesso é feito por um servidor remoto, já o gRPC é um *framework* do Google de alto desempenho para atender chamadas RPC (*Remote Procedures Call*), utiliza-se *Protobuf* (*Protobuf* ou *buffer* é um protocolo de mensagens agnóstico de linguagem criado pela *Google* para serializar dados estruturados) como mecanismo de serialização de dados para redução de tráfego na rede (Zhuanlan, 2023).

O gRPC é uma estrutura RPC (*Remote Procedure Call*) moderna de código aberto e de alto desempenho que pode ser executada em qualquer ambiente. Podendo conectar serviços de forma eficiente dentro e entre datacenters com suporte conectável para balanceamento de carga, rastreamento, verificação de integridade e autenticação, é aplicável também na última milha da computação distribuída para conectar dispositivos, aplicativos móveis e navegadores a serviços de *back-end* (GRPC, 2023).

No gRPC são usados *proto buffers* para definição das funcionalidades e a comunicação é feita usando o protocolo HTTP/2 (DEV, 2023).

O objetivo deste trabalho é estudar sobre o programa gRPC, analisando o quão útil esta ferramenta é para empresas de diversos setores desde bancos, lojas etc. Esta pesquisa se justifica devido à relevância tanto para a sociedade como para profissionais do setor a saber o quão é importante a utilização das ferramentas tecnológicas nas empresas e o quanto os programas vem se aprimorando ao longo dos tempos.

Foi realizada a pesquisa bibliográfica nas bases Scielo e Google a fim de buscar informações atualizadas sobre o tema, bem como as vantagens e as desvantagens, seus benefícios limites e como as empresas estão implementando essa ferramenta.

2 REVISÃO BIBLIOGRÁFICA

2.1 *Remote procedure call (rpc): conceitos e modos de funcionamento*

A ideia do RPC é bastante simples. Isso é baseado na observação de que as chamadas de procedimento são um mecanismo bem conhecido e bem compreendido para a transferência de controle e dados dentro de um programa em execução em um único computador. Assim, propõe-se que este mesmo mecanismo seja estendido para fornecer transferência de controle e dados através de uma comunicação rede (Birrell; Nelson, 1984, p. 39).

O WCF (*Windows Communication Foundation*) e o gRPC são implementações do padrão RPC (*Chamada de Procedimento Remoto*). Esse padrão tem como objetivo fazer com que chamadas para serviços executados em um computador diferente ou, em um processo diferente, funcionem perfeitamente, como chamadas de método no aplicativo cliente. Embora os objetivos do WCF e do gRPC sejam os mesmos, os detalhes da implementação são bem diferentes (James *et al.*, 2023).

Com relação ao modo de funcionamento, quando um procedimento remoto é invocado, o ambiente de chamada é suspenso, os parâmetros são passados pela rede para o ambiente onde o procedimento deve ser executado, e o procedimento desejado é executado lá. Quando o procedimento termina e produz seus resultados, os resultados são passados de volta para o ambiente de chamada, onde a execução é retomada como se estivesse retornando de uma simples chamada de máquina única (Birrell; Nelson, 1984, p. 39).

Para realizar uma pesquisa em um aplicativo ou mecanismo de busca, é possível utilizar uma API (*Application Programming Interface*), que receberá os dados do requisitante e retornará ao mesmo a resposta para a pesquisa. API é um meio de comunicação entre aplicativos, e existem alguns padrões de uso, dentre eles, SOAP, REST, e o tema desse trabalho o gRPC.

A proposta do uso gRPC é para que o cliente interaja com o servidor por meio de chamadas de funções simples, ou seja, de interfaces de códigos geradas automaticamente pela própria aplicação do gRPC. Isso significa que é preciso apenas implementar uma lógica de programação, o que facilita muito a adoção desse recurso. (ZUP, 2023).

Algumas vantagens na sua arquitetura de microsserviços, como:

- Facilidade do contato entre cliente e servidor;
- Melhoria do desempenho dos serviços;
- Disponibilidade e variedades de idiomas;
- Programa leve e rápido, pois utiliza algoritmos binários;
- Amplos recursos do HTTP/2, como *streaming* de dados, *load balance*, monitoramento etc.

Em resumo, é como se declarasse funções e classes em um *backend* e pudesse acessá-los no *front-end* graças ao arquivo de contrato que contém suas interfaces (serviços e DTOs) (ZUP, 2023).

Deste modo, como toda tecnologia não resolve todos os problemas, o gRPC não é perfeito, existem algumas desvantagens também, como por exemplo: (Isantos, 2023)

- O *protobuf* não possui um *package manager* para gerenciar as dependências entre arquivos de interface,
- Exige uma mudança pequena de paradigma em relação ao modelo ReST,
- A curva de aprendizado inicial é complexa,
- A especificação não é conhecida por muitos,
- A arquitetura de um sistema utilizando gRPC pode tornar-se um pouco mais complexa.

Independente dos problemas e de tudo que a tecnologia tem para oferecer, temos uma série de casos de uso bem famosos no mundo open source que utiliza o gRPC como meio de comunicação (Isantos, 2023).

Na opinião de Newman (2015), caso um sistema baseado em arquitetura de microsserviços realize a comunicação entre si de forma errada, pode-se esperar que haja lentidão e até mesmo falha nas comunicações.

De acordo com Fadilpašić (2020), a editora O'Reilly realizou uma pesquisa com 1500 engenheiros de *software*, e chegou ao resultado de que 77% dos pesquisados utilizam uma arquitetura de microsserviços. Tendo em vista que um microsserviço realiza uma tarefa específica perante o todo de um projeto de *software* e de forma autônoma (Newman, 2015), se faz necessário existir uma comunicação entre esses microsserviços.

3 DESENVOLVIMENTO

3.1 Características metodológicas

Uma pesquisa bibliográfica possibilita um amplo alcance de informações, além de permitir a utilização de dados diversos em inúmeras publicações, auxiliando também na construção, ou na melhor definição do quadro conceitual que envolve o objeto de estudo proposto (Gil, 1994). Podendo estabelecer que uma pesquisa bibliográfica e científica que possui afinidade com a definição combinada, por meio de um estudo de caso real.

A revisão bibliográfica e as informações pesquisadas foram realizadas entre o período abril/2023 a agosto/2023 e em sites como o Scielo e Google entre outros canais eletrônicos nacionais e internacionais, com período entre abril e agosto de 2023.

3.2 Procedimentos operacionais

Para a realização deste artigo seguindo as etapas de indicadores para uma pesquisa científica, optou-se por adquirir informações através de profissionais do ramo da programação, esses são profissionais com uma vasta experiência nesse setor, foi utilizado também o uso de pesquisas disponíveis em livros, teses, artigos e outros documentos relevantes ao tema proposto. Para o levantamento de dados foram utilizadas bases de conteúdo como a Scielo e Google a fim de revisar artigos e teses experimentais de estudo sobre o desenvolvimento do gRPC. Esta pesquisa justifica-se devido à relevância tanto para a sociedade como para profissionais do setor em grande desenvolvimento, a fim de trazer conhecimento e conceitos para os profissionais e a sociedade.

3.3 Funcionalidade do programa

O gRPC possui uma grande semelhança em seu funcionamento em relação ao RPC, que realiza a chamada do procedimento diretamente pela rede, sem a utilização de intermediários, conforme mostra abaixo na Figura 1.

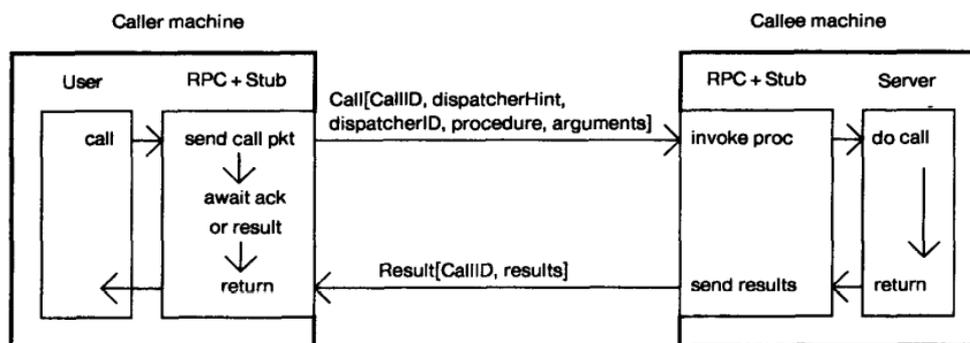


Figura 1 - Esquema exemplificando o funcionamento de uma chamada RPC
 Fonte: BIRRELL; NELSON. (1984, p. 39)

Já o gRPC possui uma sutil diferença que é a utilização de um intermediário chamado de *protobuf*, que é responsável por identificar os procedimentos disponíveis e seus respectivos parâmetros necessários para que o procedimento seja executado corretamente, como demonstra a figura 2 abaixo.

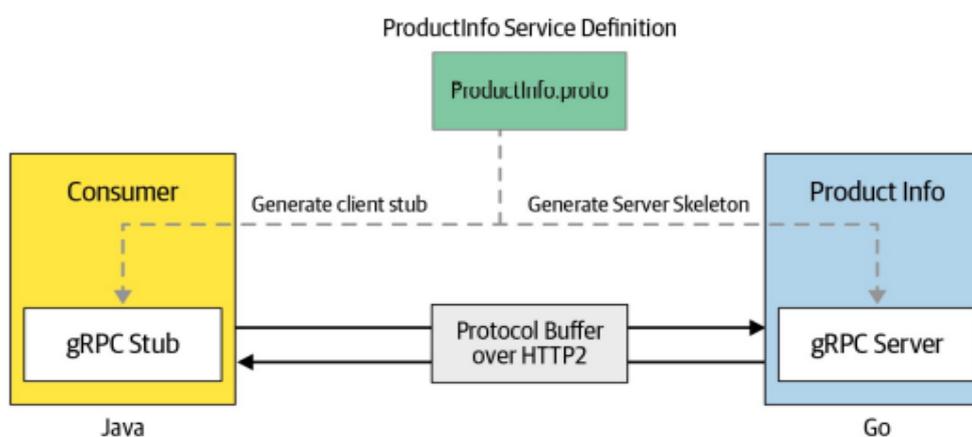


Figura 2 - Esquema exemplificando o funcionamento de uma chamada gRPC.
 Fonte: INDRASIRI; KURUPPU. (2020, p. 40)

Conforme exemplificado na figura 2 acima, o grande benefício da utilização de um intermediário é a possibilidade de realizar a comunicação de serviços em linguagens de programação diferentes, como o exemplo uma aplicação feita na linguagem JAVA realiza a chamada para uma aplicação feita na linguagem GO sem nenhum problema, pois o intermediador faz com que os dados sejam transportados da mesma forma, tanto no envio da requisição, quanto na resposta.

4 RESULTADOS

O uso de gRPC para construção de APIs vem se tornando cada vez mais popular o que torna necessário também ferramentas que deem suporte a esse tipo de tecnologia. Dito isso, neste trabalho foi apresentado o desenvolvimento de uma ferramenta de monitoramento de APIs gRPC, o HealthCheckAPI. Essa ferramenta tem como foco principal além de fazer monitoramento e

observabilidade das API, fornecer suporte e uma boa experiência para os seus usuários a fim de fomentar e tornar cada vez mais fácil criar e manter APIs usando esse tipo de tecnologia (Vertigo, 2023).

5 CONCLUSÃO

O gRPC possui uma grande semelhança em seu funcionamento em relação ao RPC, que realiza a chamada do procedimento diretamente pela rede, sem a utilização de intermediários, enquanto o gRPC possui uma sutil diferença que é a utilização de um intermediário chamado de protobuf, que é responsável por identificar os procedimentos disponíveis e seus respectivos parâmetros necessários para que o procedimento seja executado corretamente.

REFERÊNCIAS

BIRRELL, A.; NELSON, B. Implementing Remote Procedure Calls. **ACM Transactions on Computer Systems**, v. 2, n. 1, p. 39-59, fev. 1984. Disponível em: <https://dl.acm.org/doi/pdf/10.1145/2080.357392>. Acesso em: 17 jun. 2023.

DEV. **Componentes gRPC**. [S. l.]: DEV, 2023. Disponível em: <https://dev.to/expertostech/o-que-e-grpc-seus-componentes-rpc-e-http2-parte-1-nnm>. Acesso em: 18 ago. 2023.

GIL, A. C. **Métodos e técnicas de pesquisa social**. São Paulo: Atlas, 1994.

INDRASIRI, K.; KURUPPU, D. **gRPC Up & Running: Building Cloud Native Applications with Go and Java for Docker and Kubernetes**. First Edition. Sebastopol: O'Reilly, 2020.

ISANTOS. Desvantagens do gRPC. **Blog Isantos**, 2023. Disponível em: [https://blog.isantos.dev/guia-grpc1/#:~:text=O%20gRPC%20foi%20criado%20pela,RPC%20\(Remote%20Procedure%20Call\)](https://blog.isantos.dev/guia-grpc1/#:~:text=O%20gRPC%20foi%20criado%20pela,RPC%20(Remote%20Procedure%20Call)). Acesso em: 18 ago. 2023.

JAMES, N. K. **Padrão**. [S. l.: s. n.], 2023. Disponível em: <https://learn.microsoft.com/pt-br/dotnet/architecture/grpc-forwcfdevelopers/approach>. Acesso em: 18 ago. 2023.

NEWMAN, S. **Building microservices**. Sebastopol: O'Reilly, 2015.

VERTIGO. **gRPC para construção de APIs**. [S. l.]: Vertigo, 2023. Disponível em: <https://vertigo.com.br/grpc-ou-rest-qual-utilizar/>. Acesso em: 17 jun. 2023.

ZHUANLAN. **RPC**. [S. l.]: Zhuanlan, 2023. Disponível em: <https://zhuanlan.zhihu.com/p/373184014>. Acesso em: 28 ago. 2023.

ZUP. **Vantagens do gRPC**. [S. l.]: ZUP, 2023. Disponível em: <https://www.zup.com.br/blog/grpc-o-que-e-beagle>. Acesso em: 18 ago. 2023.