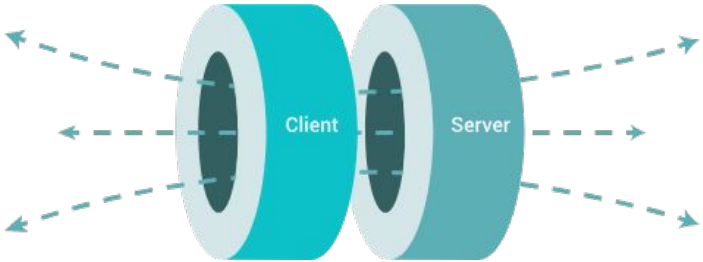
Raul Mateo, Sandra Flores, Agustín D’Acunto, Adam Scher

Introducción

- gRPC es un framework open source que se puede ejecutar en cualquier entorno
  - Framework
  - Open source
- Permite la conexión entre aplicaciones cliente y servicios ubicados en diferentes servidores sin la necesidad de que los servicios conozcan los detalles de los servidores.
- Usa HTTP/2 y buffers de protocolo para transportar información.



- Se usa para conectar dispositivos, aplicaciones móviles y navegadores a servicios backend.
- Es compatible con C/C++, C#, Java, Node.js, Ruby y Python, entre otros.
- Altamente escalable, por lo que se puede utilizar desde pequeños proyectos caseros hasta grandes compañías multinacionales.
    - Se usa en Google, Dropbox, Netflix, Cisco, etc
- Capaz de generar librerías para Android e iOS.
- Dispone de loadbalacing, autenticación, etc.

¿Dónde se utiliza?

Un sitio donde se implementa gRPC es google, esto dicen de ello:

Google ha estado utilizando una única infraestructura RPC de propósito general llamada Stubby para conectar la gran cantidad de microservicios que se ejecutan dentro y en todos los centros de datos durante más de una década. Los sistemas internos han abrazado por mucho tiempo la arquitectura de microservicio que está ganando popularidad hoy en día. Stubby ha impulsado todas las interconexiones de microservicios de Google durante más de una década y es la columna vertebral de RPC detrás de cada servicio de Google que utiliza hoy en día. En marzo de 2015, se decidió construir la próxima versión de Stubby.

Historia

- La primera pre-release que lanzaron de gRPC fue la versión 0.5.0, el 26 de febrero del 2015, y fue su primer lanzamiento público. Daba soporte a: c-core, C++, Ruby, Node. Las próximas plataformas a las que darían soporte serian: Python, PHP, C# y Objective-C.
- Desde entonces han ido actualizándose y mejorando, actualmente suben versiones cada 6 semanas, en las cuales hacen cambios funcionales, no necesariamente en todas las plataformas que soportan.
- Actualmente está disponible la versión 1.19.0, subida el 27 de febrero de 2019.


Características relevantes

- Propósito general y rendimiento: aplicable a una larga variedad de casos de uso a la vez sin sacrificar rendimiento
- Gratis y *open-source*
- Cobertura y simplicidad: disponible en una gran variedad de plataformas de desarrollo y capaz de correr en dispositivos de gama baja
- *Async/sync*: permite intercambiar mensajes con el servidor de forma síncrona y asíncrona
- Cancelables y *timeouts*: las operaciones son cancelables y el cliente puede determinar un tiempo límite para completar una operación

Ventajas

gRPC utilitza la semántica de HTTP, usando HTTP/2, y permite explícitamente full-duplex streaming, y normalmente opera muy rápidamente.

gRPC usa buffers de protocolo (Protobuf) para los mensajes, que es un formato muy eficiente y bien encapsulado, en vez de JSON, que es solo texto.

Acepta más de 10 lenguajes: C/C++, Java (incl. support for Android), Objective-C (for iOS), Python, Ruby, Go, C#, Node.js.

Los servicios RPC son más simples y funcionan específicamente bien cuando el cliente y el servidor están muy acoplados y siguen el mismo ciclo de desarrollo.

El modelo conceptual usado por gRPC da servicios con interfaces claras y mensajes estructurados (por Protobuf). Traduce los conceptos (interfaces, funciones, estructuras de datos) directamente del lenguaje de programación, lo que permite generar librerías del cliente automáticamente para el usuario.

Desventajas

El soporte para gRPC en el navegador no está bien acabado aún, por eso se usa principalmente para servicios internos, no expuestos directamente al mundo.

Los buffers de protocolo son eficientes y compactos pero son difíciles de entender, porque no son tan simples como JSON.

La documentación proporcionada por gRPC no está bien organizada y es muy difícil para el usuario entender cómo usar el framework en función de ésta.

El cliente y el servidor tienen que estar acoplados fuertemente para que funcione correctamente.

Conclusión y valoración

- gRPC es un framework que da soporte a muchos lenguajes y entornos, tanto lenguajes de front-end, back-end y a protocolos de transmisión de información.
- Es ideal para microservicios.
- Se actualiza y saca versiones cada muy poco tiempo, de manera que se corrigen los errores rápidamente, dando un soporte ágil a las plataformas.
- Es open source

Referencias

https://code.tutsplus.com/tutorials/rest-vs-grpc-battle-of-the-apis--cms-30711

https://www.reddit.com/r/golang/comments/7lhfau/grpc_pros_and_cons/

https://github.com/grpc/grpc/releases

https://grpc.io/blog/gablogpost

https://grpc.io/about/

Reparto

- Adam Scher > Introducción
- Raul Mateo > Características relevantes
- Agustín D’Acunto > Ventajas y desvetajas
- Sandra Flores > Historia y donde se utiliza