UNIVERSIDAD POLITÉCNICA SALESIANA ECUADOR

POSGRADOS

Maestría en

Ingeniería de software

Asignatura:

Patrones de Integración Empresarial

Tarea Nro. 06:

Integración con GRPC

Autor(es):

Cabascango Garcia Amanda Elizabeth
Calo Catota Carlos Edison
Fuentes Espinoza Pablo Gustavo
Guaman Guaman Saul German
Guerra Campuzano Cesar Hugo
Rengel Rivera Mateo Santiago
Vela Moya Christian Eduardo

Remote Procedure Invocation (RPC)

Remote Procedure Invocation (RPC) es un modelo de comunicación que permite a un programa ejecutar procedimientos en un servidor remoto como si fueran llamadas locales. Esto facilita la integración entre sistemas distribuidos al abstraer la complejidad de la comunicación en red.

Comparación entre RPC, REST y gRPC

Característica	RPC	REST	gRPC
Basado en	Llamadas de procedimiento	HTTP y recursos	Protocol Buffers
Eficiencia	Alta (Binario)	Media (Texto)	Muy alta (Binario y HTTP/2)
Complejidad	Media	Baja	Alta
Compatibilidad	Limitada	Amplia	Necesita cliente gRPC
Seguridad	Depende de implementación	Basado en HTTPS	Soporta autenticación TLS

Ventajas y Desventajas del Uso de RPC en Sistemas Distribuidos

Ventajas:

- Mayor eficiencia en la comunicación que REST, especialmente en entornos de alto rendimiento.
- Protocolo flexible que permite el uso de diferentes implementaciones.
- Soporte para múltiples lenguajes de programación.

Desventajas:

- Mayor complejidad de implementación en comparación con REST.
- Requiere herramientas específicas para la generación de código y la comunicación (ej., gRPC, Apache Thrift).
- Puede presentar problemas de interoperabilidad entre diferentes tecnología

Ejemplos de Aplicaciones Empresariales

El uso de RPC es recomendado en diversos sectores y casos de uso, donde la eficiencia y la comunicación rápida entre servicios son clave. A continuación, se presentan algunos ejemplos de aplicación:

Sector	Aplicación	Beneficio
Finanzas	Procesamiento de transacciones bancarias	Baja latencia y alta seguridad en la comunicación
E-Commerce	Gestión de inventarios y pagos en tiempo real	Sincronización eficiente entre múltiples servicios
Telecomunicaciones	VoIP y servicios de mensajería instantánea	Comunicación en tiempo real con menor latencia
Juegos en línea	Sincronización de estados entre jugadores	Experiencia fluida sin retrasos

IoT (Internet de las Cosas)	Comunicación entre dispositivos inteligentes	Optimización en el uso del ancho de banda
Salud	Registros electrónicos de pacientes en la nube	Acceso rápido y seguro a los datos médicos

Repositorio con Ejemplo Práctico

El repositorio incluye una implementación de una comunicación RPC utilizando **gRPC**. La implementación se encuentra en el repositorio del **proyecto del grupo 1** en GitHub: proyectoFinal-g1.

Implementación Técnica

En el repositorio del proyecto del grupo 1, específicamente en la ruta postservice, se ha implementado un servicio utilizando gRPC para gestionar las operaciones relacionadas con publicaciones. A continuación, se detalla la implementación técnica realizada:

1. Definición del Servicio gRPC

Se creó un archivo de definición .proto que describe los servicios y mensajes utilizados en la comunicación entre el cliente y el servidor.

2. Generación de Código a partir del Archivo .proto

Utilizando las herramientas de gRPC, se generaron las clases base tanto para el servidor como para el cliente a partir del archivo de definición.

3. Implementación del Servidor gRPC

En el lado del servidor, se desarrolló una clase que extiende la interfaz generada a partir del archivo .proto. Esta clase implementa los métodos definidos, manejando la lógica de negocio correspondiente a cada operación sobre las publicaciones.

4. Implementación del Cliente gRPC

Para el cliente, se creó una clase que utiliza el stub generado para comunicarse con el servidor gRPC.

Conclusión

RPC es una técnica eficiente para la comunicación entre sistemas distribuidos, especialmente cuando se busca alto rendimiento. Sin embargo, su implementación requiere herramientas específicas y un mayor esfuerzo en la configuración. Comparado con REST, RPC es más rápido y eficiente, aunque con menor compatibilidad general. Implementaciones modernas como gRPC mejoran sus capacidades, haciendo de RPC una opción viable para microservicios y aplicaciones en tiempo real.

Referencias

- Birrell, A. D., & Nelson, B. J. (1984). Implementing remote procedure calls. *ACM Transactions on Computer Systems*, *2*(1), 39-59.
- Tanenbaum, A. S., & Van Steen, M. (2007). *Distributed Systems: Principles and Paradigms*. Prentice Hall.

- Google Developers. (2024). gRPC: A high-performance, open-source universal RPC framework. Disponible en: https://grpc.io
- Apache Thrift. (2024). The Apache Thrift framework. Disponible en: https://thrift.apache.org
- Fielding, R. (2000). Architectural styles and the design of network-based software architectures. *Doctoral dissertation, University of California, Irvine*.
- Repositorio del Proyecto: https://github.com/UpsIE2025/proyectoFinal-g1