

gRPC

Tu framework de llamadas a procedimientos remotos favorito

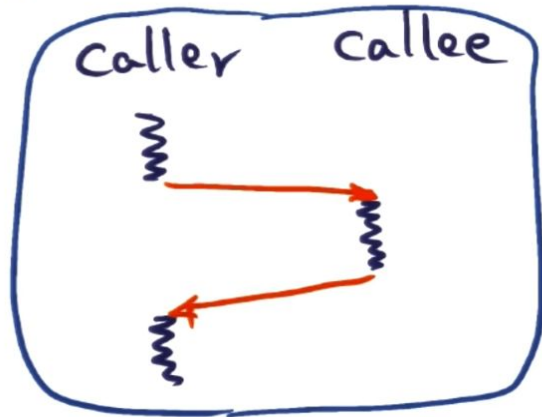
¿Qué es un RPC?

- Tipo de comunicación cliente servidor.
- Sirve para realizar llamadas en procedimientos en otros computadores diferentes al del propio cliente.

Semántica	Funcionamiento
tal-vez	<ul style="list-style-type: none">■ cliente no retransmite sus peticiones (no usa ACK)■ servidor no filtra peticiones duplicadas
al-menos-una	<ul style="list-style-type: none">■ cliente retransmite sus peticiones (usa ACK + temporizador)■ servidor no filtra peticiones duplicadas■ ante peticiones repetidas, servidor repite ejecución
como-máximo-una	<ul style="list-style-type: none">■ cliente reintenta retransmitir peticiones (usa ACK + temporizador)■ servidor filtra peticiones duplicadas■ ante peticiones repetidas, servidor retransmite las respuestas pasadas

RPC Vs. Simple Procedure Call

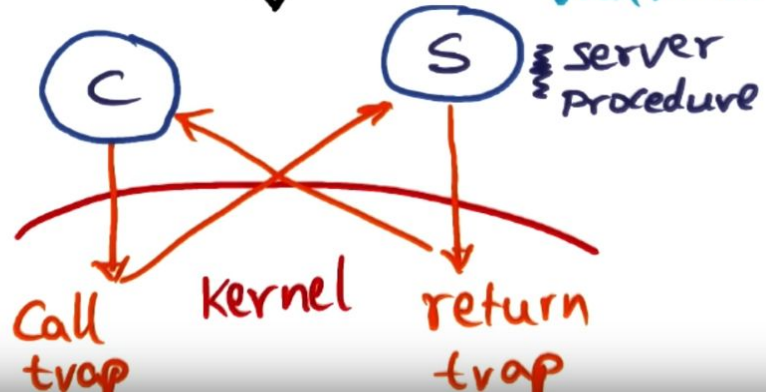
Procedure Call



Process

— All at
completetime

RPC



<https://www.youtube.com/watch?v=gr7oaiUsxSU>

Ventajas...

y desventajas de un RPC

1. Se permite una modulación coherente, aliviando la carga al repartirla entre varios ordenadores.
2. Se facilita la comunicación entre máquinas.
3. El sistema encargado de un servicio lo realiza de forma más eficiente al estar adaptado a este.

1. No existe un estándar unificado relativo a este protocolo.
2. Aun teniendo una velocidad elevada, efectuar las llamadas en modo local seguiría siendo más rápido.
3. Los sistemas suelen ser más susceptible a errores y vulnerables.

¿Qué es gRPC?

gRPC es un RPC **gratis** y de código abierto desarrollado inicialmente por Google, utilizando el protocolo HTTP/2 y los protocol Buffers.

Proporcionando multitud de características como autenticación, control de flujo, cancelaciones y tiempos de espera.

Siendo lo más destacado el soporte multiplataforma entre cliente y servidor para multitud de lenguajes.

Motivación del proyecto

Antes del desarrollo de gRPC Google usaba Stubby, un RPC de propósito general, útil con la creciente popularidad de la **arquitectura de microservicios**.

Con gRPC consiguen tener una infraestructura uniforme multiplataforma siendo fácilmente escalable

Principios y requerimientos

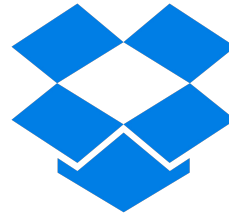
- Servicios en vez de objetos y mensajes en vez de referencias
- Gratis y código abierto
- Payload agnostic
- Control de flujo
- Extensiones como APIs
- Pluggable
- Intercambio de metadatos

Lenguajes soportados



Usuarios

NETFLIX



Características propias de gRPC

- Sincronicidad y asincronicidad
- Timeouts
- Terminación y cancelación del RPC
- Canales
- Servicios...

Protocol Buffers

Estructura de datos serializados

Desarrollada por Google, open source

XML meets JSON

```
//polyline.proto
syntax = "proto2";

message Point {
    required int32 x = 1;
    required int32 y = 2;
    optional string label = 3;
}

message Line {
    required Point start = 1;
    required Point end = 2;
    optional string label = 3;
}

message Polyline {
    repeated Point point = 1;
    optional string label = 2;
}
```

Servicios de gRPC

Definidos en Protocol Buffers

Comunicación a través de Protocol Buffers

Cuatro tipos de servicio distintos

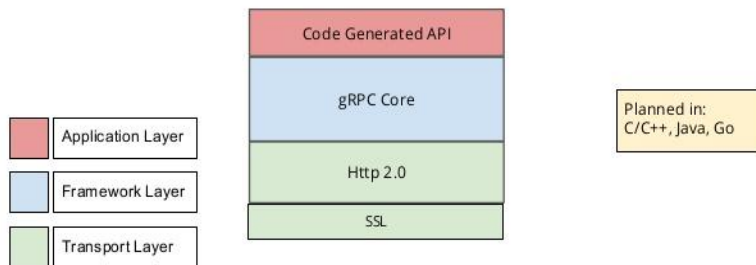
- Unarios
- Streaming servidor-cliente
- Streaming cliente-servidor
- Streaming bidireccional

```
service HelloService {  
    rpc SayHello (HelloRequest) returns (HelloResponse);  
}  
  
message HelloRequest {  
    string greeting = 1;  
}  
  
message HelloResponse {  
    string reply = 1;  
}
```

```
rpc SayHello(HelloRequest) returns (HelloResponse);  
  
rpc LotsOfReplies(HelloRequest) returns (stream HelloResponse);  
  
rpc LotsOfGreetings(stream HelloRequest) returns (HelloResponse);  
  
rpc BidiHello(stream HelloRequest) returns (stream HelloResponse);
```

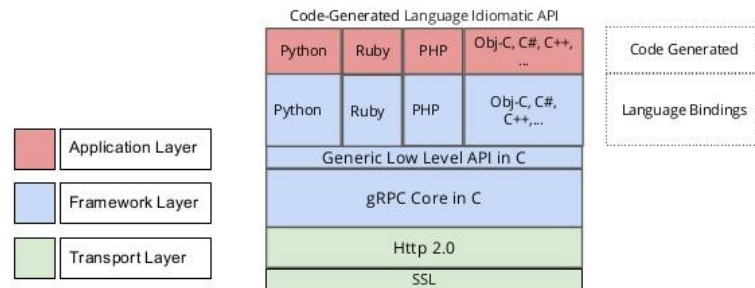
Arquitectura

Architecture: Native Implementation in Language



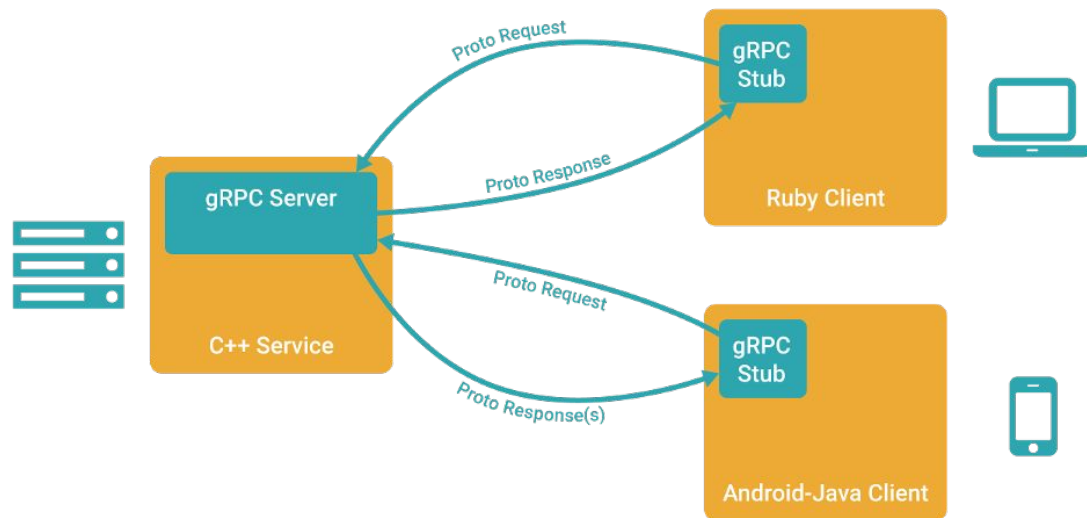
Implementado nativamente en 3 lenguajes Java, C y Go.

Architecture: Derived Stack



El soporte del resto de lenguajes está desarrollado en la implementación de C-Core.

Arquitectura cliente/servidor del proyecto



Construyendo APIs

Servicios definidos en archivos

`.proto`

- Servidor implementa la interfaz de los servicios.
- Cliente implementa en un objeto local (Stub) que contiene los mismos métodos de los servicios.

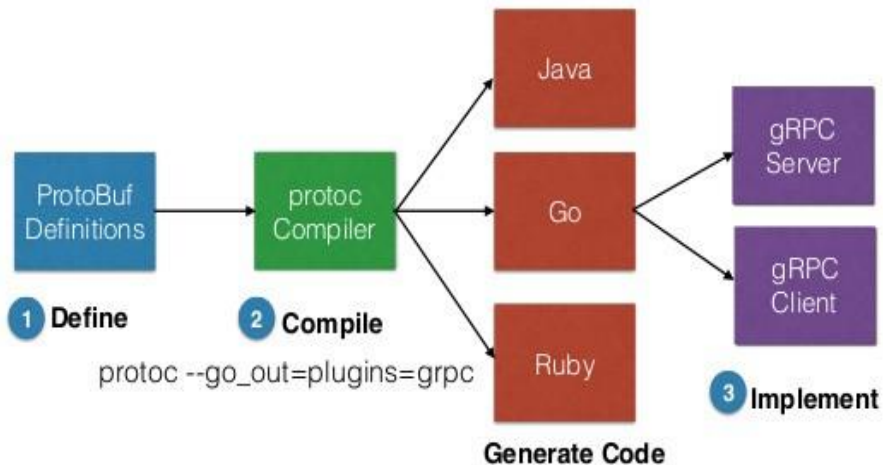
Stubs en el cliente

- Las clases que se generan en la compilación contienen los objetos locales Stubs que son usados por los clientes para llamar a los métodos definidos en los servicios.
- Cada stub envuelve un Channel que es un pipeline o conducto que es utilizado para mandar RPCs al servicio.

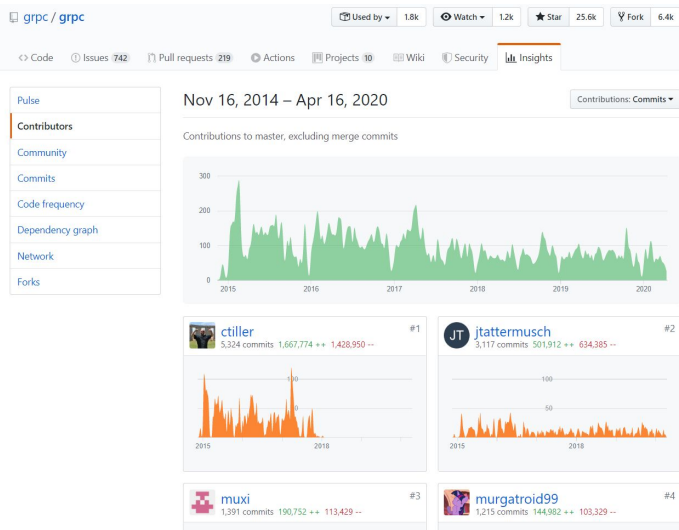
```
187
188     std::unique_ptr<grpc::testing::EchoTestService::Stub> BuildStub(
189         const std::shared_ptr<Channel>& channel) {
190         return grpc::testing::EchoTestService::NewStub(channel);
191     }
192
```

Workflow

Generación de código definido en los `.proto` en el cliente y en el servidor utilizando el compilador `protoc`.



Gestión de issues de gRPC



grpc / grpc

Used by 1.8k Watch 1.2k Star 25.6k Fork 6.4k

<> Code Issues 746 Pull requests 223 Actions Projects 10 Wiki Security Insights

Want to contribute to grpc/grpc? Dismiss

If you have a bug or an idea, read the [contributing guidelines](#) before opening an issue.
If you're ready to tackle some open issues, [we've collected some good first issues](#) for you.

Filters Labels 93 Milestones 11 New issue

746 Open ✓ 7,666 Closed Author Label Projects Milestones Assignee Sort

- Bazel build fails with incompatible_load_cc_rules_from_bzl** kind/bug priority/P2 #22653 opened 2 days ago by KeithMoyer
- Go Cloud-to-Cloud ALTS Interop Failure** kind/bug lang/go priority/P1 #22650 opened 2 days ago by gnossen 4
- 'librt.so' is in the link library list when linking static** kind/bug priority/P2 #22646 opened 4 days ago by rizzotto 1
- Can't install using pip on ubuntu 19.04** kind/bug lang/Python priority/P2 #22645 opened 4 days ago by dmars1972 1
- Python multiprocessing example shows that same process is handling the incoming requests** kind/bug lang/Python priority/P2 #22644 opened 4 days ago by srajagopalan-fliptrx 2
- Provide a way to customize generated classes in dotnet Grpc.Tools** kind/enhancement priority/P2 #22637 opened 5 days ago by HassanHashemi
- Deadlock in grpc_postfork_child** kind/bug lang/Python priority/P2 #22624 opened 7 days ago by alpire 5
- FLAKE: tests.cloud_to_prod_auth:nodetpurejs:default:jwt_token_creds:tls and tests.cloud_to_prod_auth:nodetpurejs:gateway_v4:jwt_token_creds:tls** infra/BUILDPONY kind/bug priority/P2 #22623 opened 7 days ago by markdroth

¿Cómo contribuir?

Obtaining Commit Access

We grant Commit Access to contributors based on the following criteria:

- Sustained contribution to the gRPC project.
- Deep understanding of the areas contributed to, and good consideration of various reliability, usability and performance tradeoffs.
- Contributions demonstrate that obtaining Commit Access will significantly reduce friction for the contributors or others.

In addition to submitting PRs, a Contributor with Commit Access can:

- Review PRs and merge once other checks and criteria pass.
- Triage bugs and PRs and assign appropriate labels and reviewers.

Obtaining Commit Access without Code Contributions

The [gRPC organization](#) is comprised of multiple repositories and commit access is usually restricted to one or more of these repositories. Some repositories such as the [grpc.github.io](#) do not have code, but the same principle of sustained, high quality contributions, with a good understanding of the fundamentals, apply.

Clonación y compilación del proyecto

Building with CMake

Linux/Unix, Using Make

Run from `grpc` directory after cloning the repo with `--recursive` or updating submodules.

```
$ mkdir -p cmake/build
$ cd cmake/build
$ cmake ../../
$ make
```
























If you want to build shared libraries (`.so` files), run `cmake` with `-DBUILD_SHARED_LIBS=ON` .

Windows, Using Visual Studio 2015 or 2017

When using the "Visual Studio" generator, cmake will generate a solution (`grpc.sln`) that contains a VS project for every target defined in `CMakeLists.txt` (+ few extra convenience projects added automatically by cmake). After opening the solution with Visual Studio you will be able to browse and build the code.

Clonación y compilación del proyecto

```
C:\Windows\System32\cmd.exe
-- Looking for stricmp - found
-- Looking for strncasecmp
-- Looking for strncasecmp - not found
-- Looking for strncmpi
-- Looking for strncmpi - not found
-- Looking for strnicmp
-- Looking for strnicmp - found
-- Looking for writev
-- Looking for writev - not found
-- Looking for __system_property_get
-- Looking for __system_property_get - not found
--
-- 3.11.2.0
-- Looking for a ASM_NASM compiler
-- Looking for a ASM_NASM compiler - C:/Users/Niobe/AppData/Local/bin/NASM/nasm.exe
-- The ASM_NASM compiler identification is NASM
-- Found assembler: C:/Users/Niobe/AppData/Local/bin/NASM/nasm.exe
-- Looking for stddef.h
-- Looking for stddef.h - found
-- Check size of off64_t
-- Check size of off64_t - failed
-- Looking for fseeko
-- Looking for fseeko - not found
-- Looking for unistd.h
-- Looking for unistd.h - not found
-- Renaming
--   E:/Nueva carpeta/grpc/third_party/zlib/zconf.h
-- to 'zconf.h.included' because this file is included with zlib
-- but CMake generates it automatically in the build directory.
-- Configuring done
-- Generating done
-- Build files have been written to: E:/Nueva carpeta/grpc/.build
E:/Nueva carpeta/grpc/.build>
```

Nombre	Fecha de modificación
 grpc_csharp_ext.vcxproj.filters	12/04/2020 12:11
 grpc_csharp_plugin.vcxproj.filters	12/04/2020 12:11
 grpc_node_plugin.vcxproj.filters	12/04/2020 12:11
 grpc_objective_c_plugin.vcxproj.filters	12/04/2020 12:11
 grpc_php_plugin.vcxproj.filters	12/04/2020 12:11
 grpc_plugin_support.vcxproj.filters	12/04/2020 12:11
 grpc_python_plugin.vcxproj.filters	12/04/2020 12:11
 grpc_ruby_plugin.vcxproj.filters	12/04/2020 12:11
 grpc_unsecure.vcxproj.filters	12/04/2020 12:11
 grpc++.vcxproj.filters	12/04/2020 12:11
 grpc++_alts.vcxproj.filters	12/04/2020 12:11
 grpc++_error_details.vcxproj.filters	12/04/2020 12:11
 grpc++_reflection.vcxproj.filters	12/04/2020 12:11
 grpc++_unsecure.vcxproj.filters	12/04/2020 12:11
 grpcpp_channelz.vcxproj.filters	12/04/2020 12:11
 INSTALL.vcxproj.filters	12/04/2020 12:11
 plugins.vcxproj.filters	12/04/2020 12:11
 tools.vcxproj.filters	12/04/2020 12:11
 tools_c.vcxproj.filters	12/04/2020 12:11
 tools_cxx.vcxproj.filters	12/04/2020 12:11
 upb.vcxproj.filters	12/04/2020 12:11
 ZERO_CHECK.vcxproj.filters	12/04/2020 12:11
 <u>grpc.sln</u>	12/04/2020 12:11

Vista del proyecto

Lista de errores

Toda la solución 2 Errores 24 Advertencias 0 Mensajes Compilación + IntelliSense

Cód...	Descripción	Proyecto	Archivo	Lí...
C4146	operador unario menos aplicado a un tipo unsigned; el resultado aún no tiene signo	upb	msg.c	36
C4146	operador unario menos aplicado a un tipo unsigned; el resultado aún no tiene signo	upb	msg.c	40
C4146	operador unario menos aplicado a un tipo unsigned; el resultado aún no tiene signo	upb	msg.c	46
C4116	definición de tipo sin nombre en paréntesis	upb	upb.c	192
C4715	'upb_encode_scalarfield': no todas las rutas de acceso de control devuelven un valor	upb	encode.c	310
C4715	'upb_decode_toarray': no todas las rutas de acceso de control devuelven un valor	upb	decode.c	488
C4018	'>=': no coinciden signed/unsigned	absl_strings	charconv_parse.cc	379
C4018	'>=': no coinciden signed/unsigned	absl_strings	charconv_parse.cc	382
C4018	'>=': no coinciden signed/unsigned	absl_strings	charconv_parse.cc	402
C4018	'>=': no coinciden signed/unsigned	absl_strings	charconv_parse.cc	416
C4018	'>=': no coinciden signed/unsigned	absl_strings	charconv_parse.cc	419
C4018	'<': no coinciden signed/unsigned	absl_str_format_internal	bind.cc	224
C4018	'<': no coinciden signed/unsigned	absl_status	status.cc	83
C4018	'<': no coinciden signed/unsigned	absl_status	status.cc	172
C4018	'>=': no coinciden signed/unsigned	absl_flags_usage_internal	usage.cc	166
C4018	'<': no coinciden signed/unsigned	grpc_unsecure	grpc_ares_ev_driver_wind...	324
C4146	operador unario menos aplicado a un tipo unsigned; el resultado aún no tiene signo	grpc	grpc_ares_ev_driver_wind...	324
C4146	operador unario menos aplicado a un tipo unsigned; el resultado aún no tiene signo	grpc_unsecure	resource_quota.cc	948
C4700	se utilizó la variable local 'slice' sin inicializar	grpc	slice.cc	220
C4700	se utilizó la variable local 'slice' sin inicializar	grpc_unsecure	slice.cc	220
C1001	Error interno en el compilador.	grpc	validate_service_config.cc	41
LNK4221	Este archivo de objeto no define ningún símbolo público que no se haya definido previamente; por lo que no se usará en ninguna vinculación que utilice esta biblioteca	grpc++_unsecure	server_posix.obj	1
LNK4221	Este archivo de objeto no define ningún símbolo público que no se haya definido previamente; por lo que no se usará en ninguna vinculación que utilice esta biblioteca	grpc++	rpc_method.obj	1
LNK4221	Este archivo de objeto no define ningún símbolo público que no se haya definido previamente; por lo que no se usará en ninguna vinculación que utilice esta biblioteca	grpc++	create_channel_posix.obj	1
C1001	Error interno en el compilador.	grpc++_error_details	map	597

Solución 'grpc' (101 proyectos)

- absl_bad_any_cast_impl
- absl_bad_optional_access
- absl_bad_variant_access
- absl_base
- absl_city
- absl_civil_time
- absl_cord
- absl_debugging_internal
- absl_demangle_internal
- absl_dynamic_annotations
- absl_examine_stack
- absl_exponential_biased
- absl_failure_signal_handler
- absl_flags
- absl_flags_config
- absl_flags_internal
- absl_flags_marshallling
- absl_flags_parse
- absl_flags_program_name
- absl_flags_registry
- absl_flags_usage
- absl_flags_usage_internal
- absl_graphcycles_internal
- absl_hash
- absl_hashtablez_sampler
- absl_int128
- absl_leak_check
- absl_leak_check_disable
- absl_log_severity
- absl_malloc_internal
- absl_periodic_sampler
- absl_random_distributions
- absl_random_internal_distribution_test_util
- absl_random_internal_pool_urbg
- absl_random_internal_randen
- absl_random_internal_randen_hwaes
- absl_random_internal_randen_hwaes_impl
- absl_random_internal_randen_slow
- absl_random_internal_seed_material
- absl_random_seed_gen_exception
- absl_random_seed_sequences
- absl_raw_hash_set
- absl_raw_logging_internal
- absl_scoped_set_env
- absl_spinlock_wait

Explorador de soluciones Team Explorer