

gRPC 在 Spring Cloud 中的应用

张艺辰 (Michael Zhang)

About Me

张艺辰

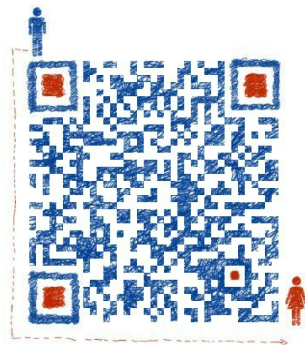
目前就职于 腾讯

曾先后就职于 开源中国、Coding.net、小牛在线

关注 微服务架构、高可用架构

GitHub: <https://github.com/yidongnan>

P.S. 今天分享并不代表腾讯对外的一个技术输出



微信

大纲

- Spring Cloud 简介和缺陷
 - gRPC 的介绍
 - gRPC Spring Boot Starter 项目
 - gRPC 周边生态
- 



Spring Cloud

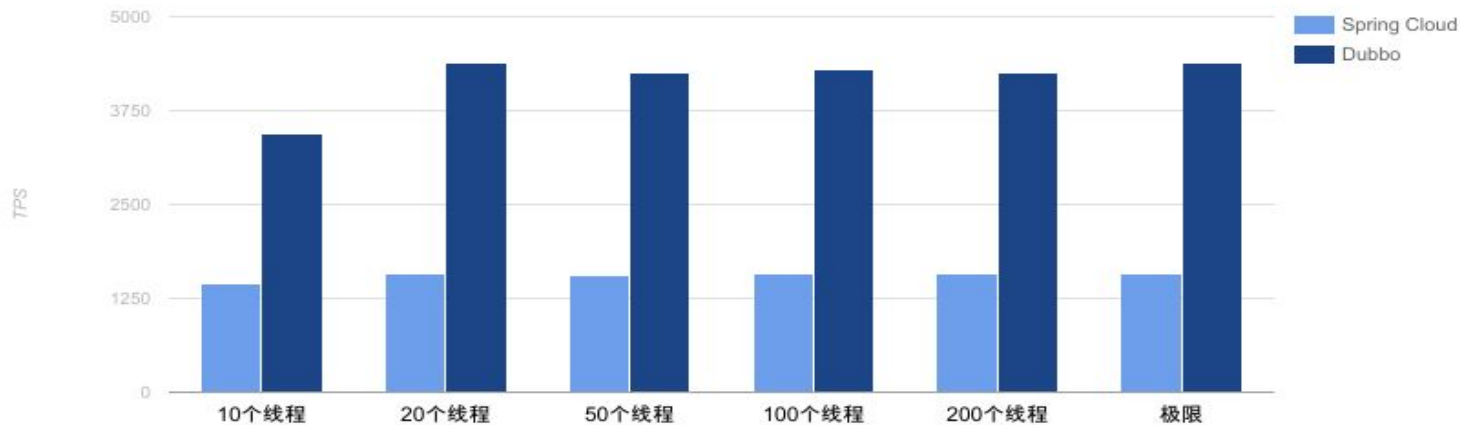
基于Spring Boot的一整套实现微服务的框架。

提供了微服务开发所需的服务发现、配置管理、断路器、智能路由、微代理、控制总线、分布式应用跟踪、分布式会话和集群状态管理等组件

微服务需要功能	Dubbo	Motan	Spring Cloud
服务注册和发现	Zookeeper	Zookeeper/ Consul	Eureka、Consul
服务调用方式	RPC	RPC	RESTful API
断路器	无	无	Hystrix
负载均衡	有	有	Ribbon
服务路由和过滤	有	无	Zuul
分布式配置	无	有	Spring Cloud Config
分布式链路追踪	无	无	Spring Cloud Sleuth
分布式消息	无	无	Spring Cloud Bus

Spring Cloud VS Dubbo

发送10W次请求，传输10个POJO对象





提高 SpringCloud RPC 性能？



gRPC 是 Google 开发的基于
HTTP/2 和 Protocol Buffer 3 的 RPC
框架

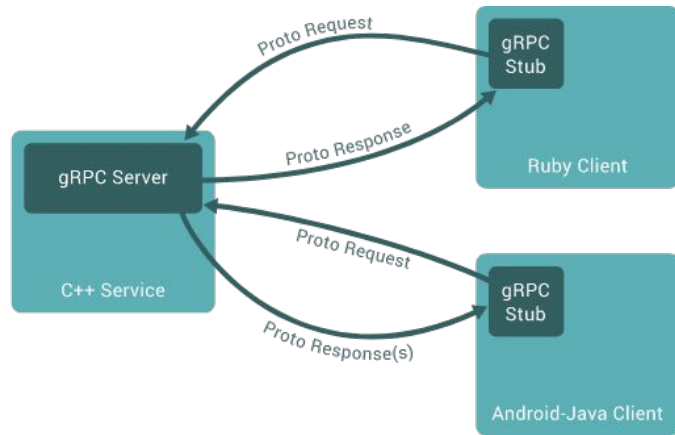
gRPC 特点

HTTP/2: 单个TCP连接、多路复用、报头压缩

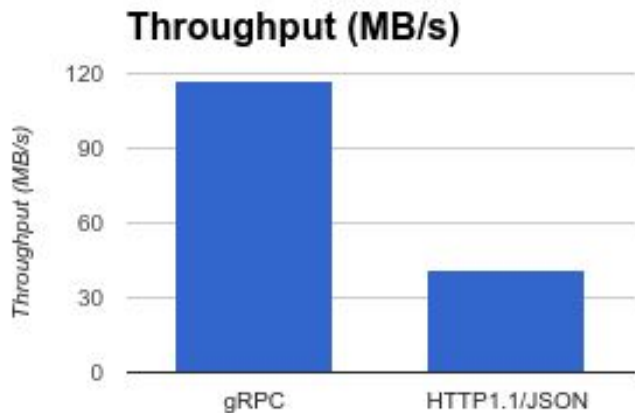
ProtocolBuffer 3.0: 支持Android Java、Objective C

平台: Linux、Android、iOS、MacOS、Windows

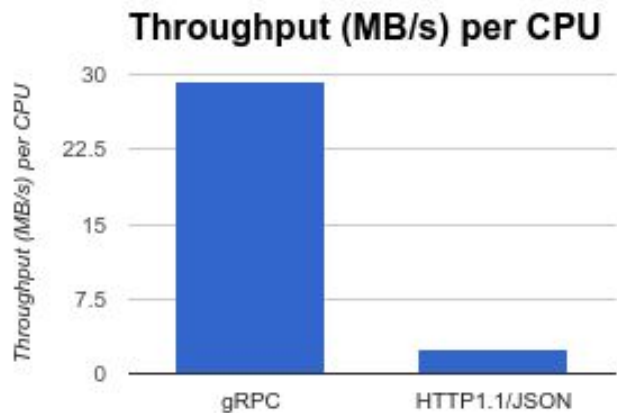
灵活: 支持同步、异步、Stream



gRPC VS JSON \ HTTP



使用 9 个 gRPC Channel。发布50KB消息 最大吞吐量



CPU资源占用是 HTTP1.1 / JSON 的 1/4, 每个CPU的TPS是HTTP1.1/JSON的11倍

<https://cloud.google.com/blog/big-data/2016/03/announcing-grpc-alpha-for-google-cloud-pubsub>

有哪些厂商在用？

Google

NETFLIX



Core OS



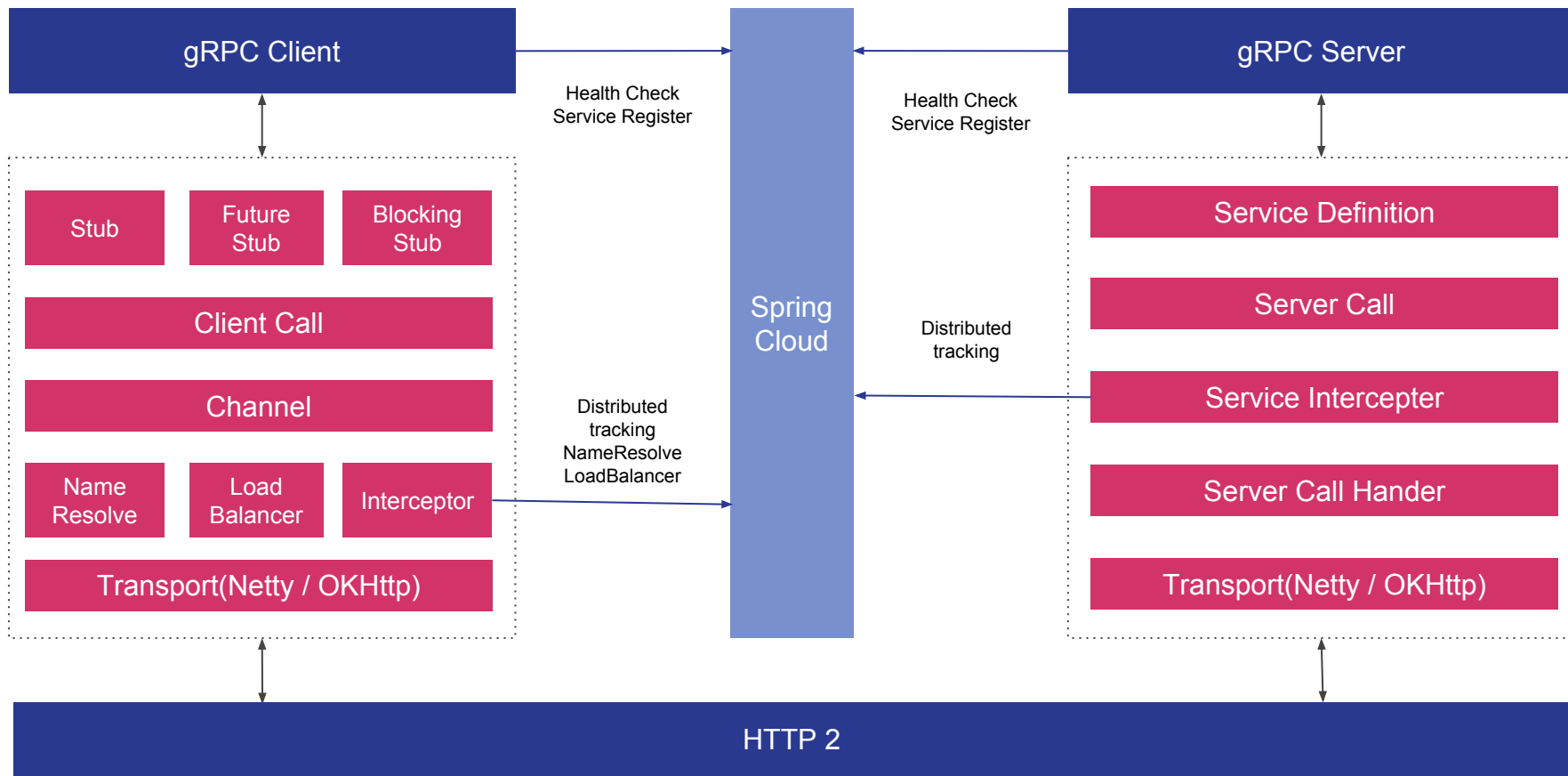
JD.COM 京东
多·快·好·省



gRPC Spring Boot Starter

- 使用 Spring Boot 的应用进行自动配置, 内嵌 gRPC server
- 支持 Spring Cloud Netflix、Spring Cloud Consul 进行服务发现和注册
- 支持 Spring Cloud Sleuth 进行分布式应用跟踪
- 支持对 gRPC Server 的健康检查

<https://github.com/yidongnan/grpc-spring-boot-starter>



定义IDL、生成gRPC代码

```
syntax = "proto3";  
package helloworld;
```

```
service Simple {  
  rpc SayHello (HelloRequest) returns (HelloReply) {  
  }  
}
```

```
message HelloRequest {  
  string name = 1;  
}
```

```
message HelloReply {  
  string message = 1;  
}
```

```
protobuf {  
  protoc {  
    artifact = "com.google.protobuf:protoc:${protobufVersion}"  
  }  
  generatedFilesBaseDir = "$projectDir/src/generated"  
  clean {  
    delete generatedFilesBaseDir  
  }  
  plugins {  
    grpc {  
      artifact = "io.grpc:protoc-gen-grpc-java:${grpcVersion}"  
    }  
  }  
  generateProtoTasks {  
    all().*.plugins {  
      grpc {}  
    }  
  }  
}
```

gRPC Server

```
dependencies {  
    compile 'net.devh:grpc-server-spring-boot-starter:1.1.1.RELEASE'  
}
```

```
@GrpcService(GreeterGrpc.class)
```

```
public class GrpcServerService extends GreeterGrpc.GreeterImplBase {
```

```
    @Override
```

```
    public void sayHello(HelloRequest req, StreamObserver<HelloReply> responseObserver) {
```

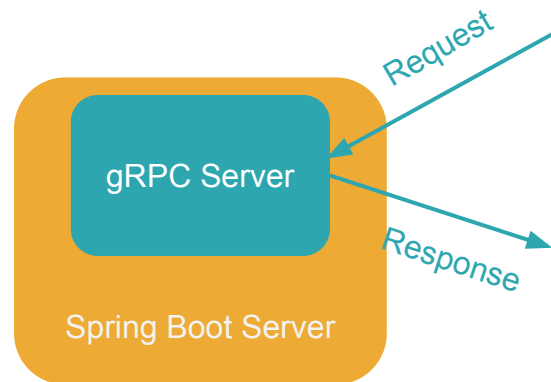
```
        HelloReply reply = HelloReply.newBuilder().setMessage("Hello " + req.getName()).build();
```

```
        responseObserver.onNext(reply);
```

```
        responseObserver.onCompleted();
```

```
    }
```

```
}
```



gRPC Client

```
dependencies {  
    compile 'net.devh:grpc-client-spring-boot-starter:1.1.1.RELEASE'  
}
```

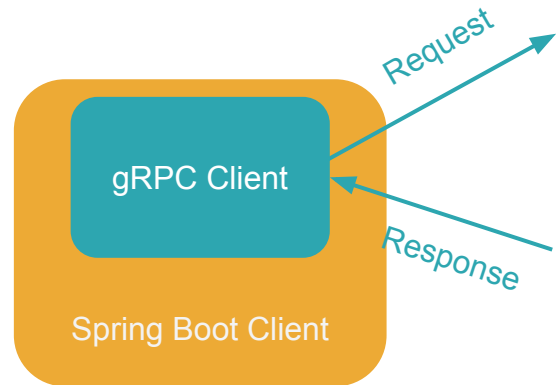
```
@GrpcClient("gRPC server name")
```

```
private Channel serverChannel;
```

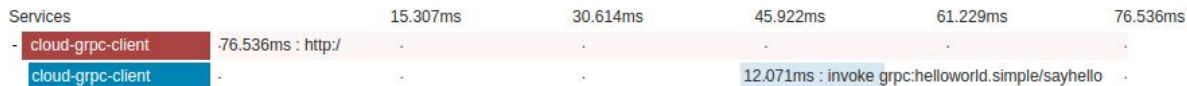
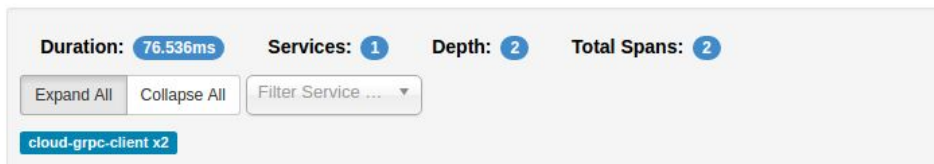
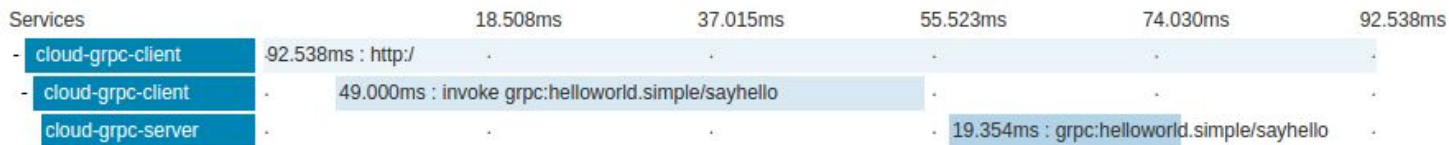
```
.....
```

```
GreeterGrpc.GreeterBlockingStub stub = GreeterGrpc.newBlockingStub(serverChannel);  
HelloReply response = stub.sayHello(HelloRequest.newBuilder().setName("Michael").build());
```

<https://github.com/yidongnan/grpc-spring-boot-starter/tree/master/examples>



分布式跟踪



gRPC Ecosystem

grpc-gateway

gRPC 转 JSON 的代理生成器

go-grpc-middleware

Golang gRPC 中间件: interceptor 链、认证、登录、重试等等.

java-grpc-prometheus

gRPC Java 集成 Prometheus 监控

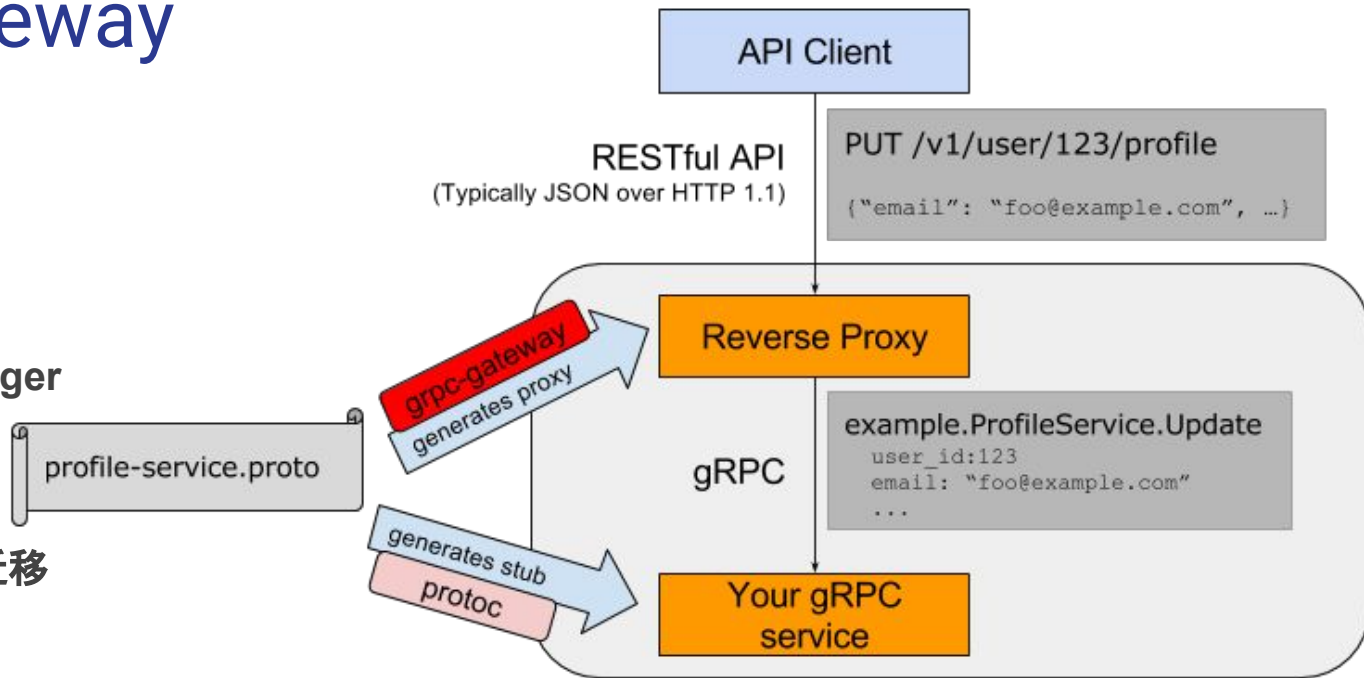
<https://github.com/grpc-ecosystem>

gRPC gateway

方便做测试

OpenAPI 、 Swagger

方便内部系统的迁移



Questions ?





WE WANT YOU



NOW





Thank You For Listening.