gRPC简介

杜远超

20190103

背景介绍

- 当前瓶颈
- 微服务架构
  - 主要目标
    - 性能提升
    - 模块剥离
  - 主流实现方式：不同模块之间的结合（调用，以及数据传输）
    - WebService / RESTful API
    - RPC

What is gRPC

- gRPC(goole Remote Procedure Call)

  - 跨语言，跨平台，
  - 接入多种设备
  - 双向流 & http/2支持
  - 可扩展：授权，追踪，负载均衡，健康检查

上下游关系

微服务

protocol buffer

HTTP/2

gRPC

服务发现

负载均衡

如何使用gRPC

protocol buffer

a language-neutral, platform-neutral, extensible way of serializing structured data for use in communications protocols, data storage, and more.

- 序列化
  - JSON `json.dump(), json.load()`
  - pickle
  - npz

```
syntax = "proto2";

package tutorial;

message Person {
  required string name = 1;
  required int32 id = 2;
  optional string email = 3;

  enum PhoneType {
    MOBILE = 0;
    HOME = 1;
    WORK = 2;
  }

  message PhoneNumber {
    required string number = 1;
    optional PhoneType type = 2 [default = HOME];
  }

  repeated PhoneNumber phones = 4;
}

message AddressBook {
  repeated Person people = 1;
}
```

```
protoc -I=$SRC_DIR --python_out=$DST_DIR $SRC_DIR/addressbook.proto
```
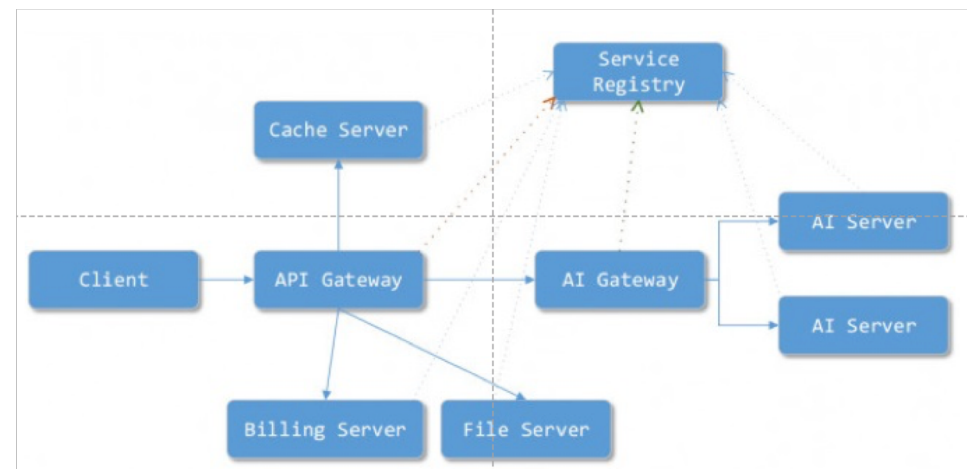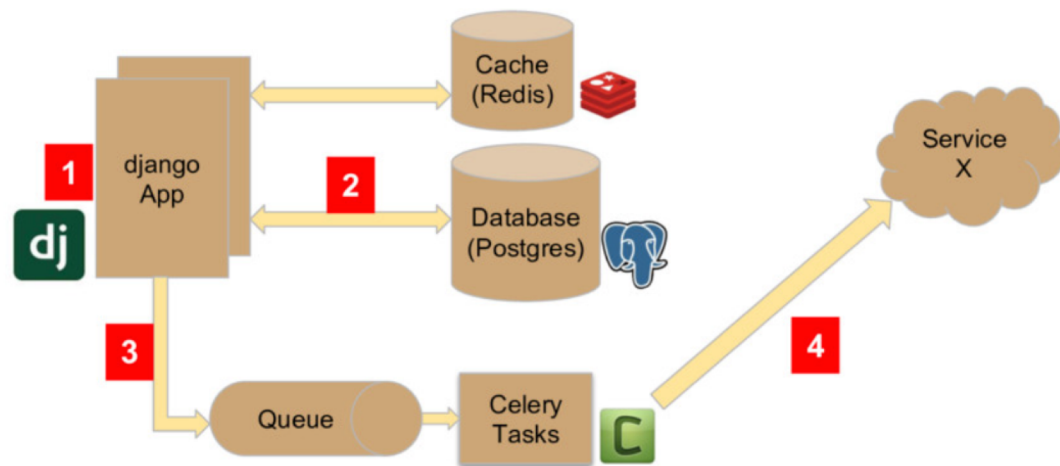
1. 定义格式
2. 生成语言代码
3. 使用pb

HTTP2

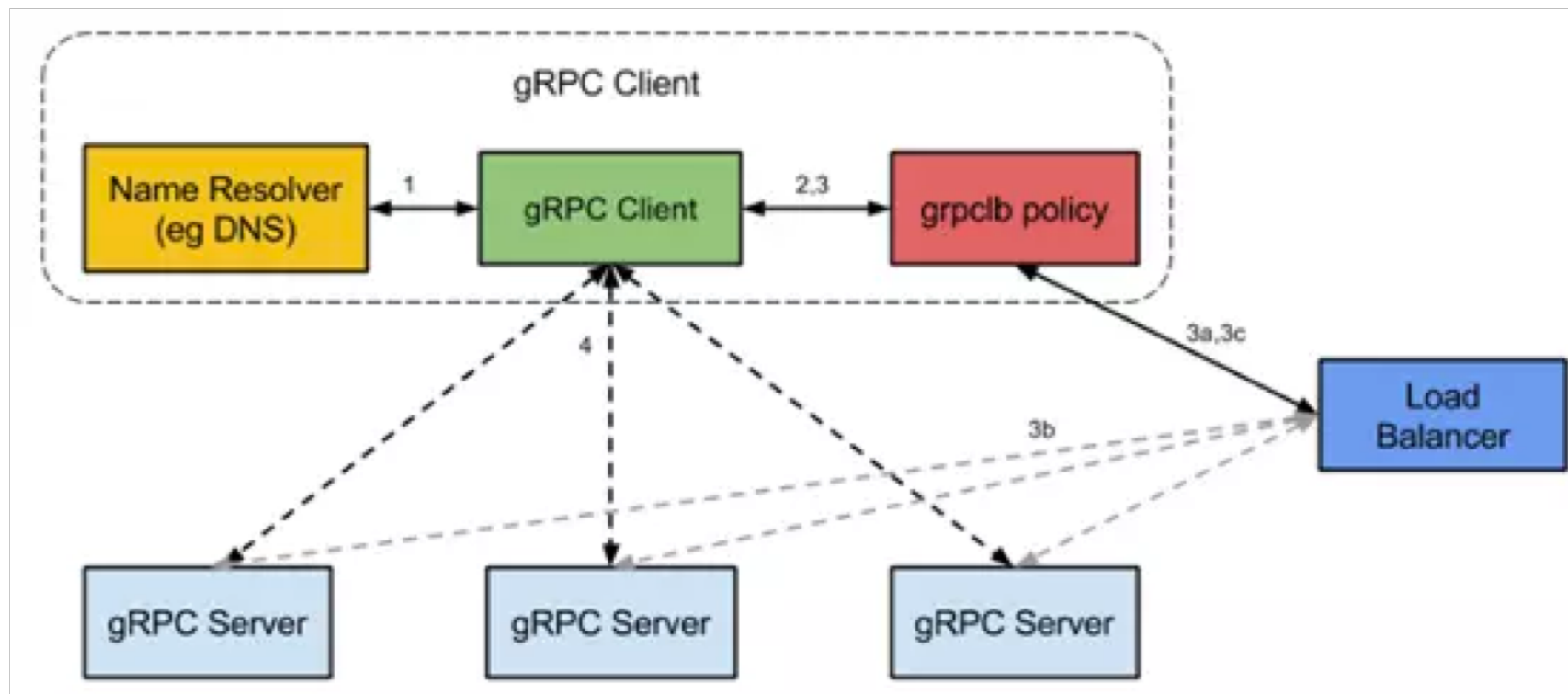| HTTP1.x	SPDY	HTTP2
SSL not required but recommended.	SSL required.	SSL not required but recommended.
Slow encryption.	Fast encryption.	Even faster encryption.
One client-server request per TCP connection.	Multiple client-server request per TCP connection. Occurs on a single host at a time.	Multi-host multiplexing. Occurs on multiple hosts at a single instant.
No header compression.	Header compression introduced.	Header compression using improved algorithms that improve performance as well as security.
No stream prioritization.	Stream prioritization introduced.	Improved stream prioritization mechanisms used.

微服务

服务发现与负载均衡

gRPC LB策略

gRPC Demo

1. 环境安装
2. 定义服务
    1. 编写.proto文件
    2. 语言编译
3. 编写server，client
4. 运行

- HelloWorld

- video_stream

总结

gRPC	web-service/ Restful API	结果
pb(强类型)	json（序列化）	JSON更灵活
HTTP/2	HTTP1.1	HTTP2
message	Resources Verbes	平手
Streaming	Request-Response	Streaming