

gRPC

RPC(remote procedure call) 가 뭘까..?

- 프로세스간 통신을 위한 IPC의 일종(Inter Process Communication)
 - IPC 에는 공유 메모리, Pipe, 메시지큐, 소켓 등 이 있습니다.
- RPC : 네트워크로 연결된 서버 상의 프로시저를 원격으로 호출
 - 원격지에 있는 함수를 호출한다고 생각하시면 됩니다!
- RPC -> 소켓의 한계점 (서비스가 커져감에 따라 개발자가 장애 처리해야하는 것과 data formating할게 많아짐)에 따라 RPC가 주목받음
- 하지만 현재 많이 안쓰이는 이유 :
 - 구현의 어려움과 지원의 한계로 인해 REST가 인기를 얻게 됨

gRPC

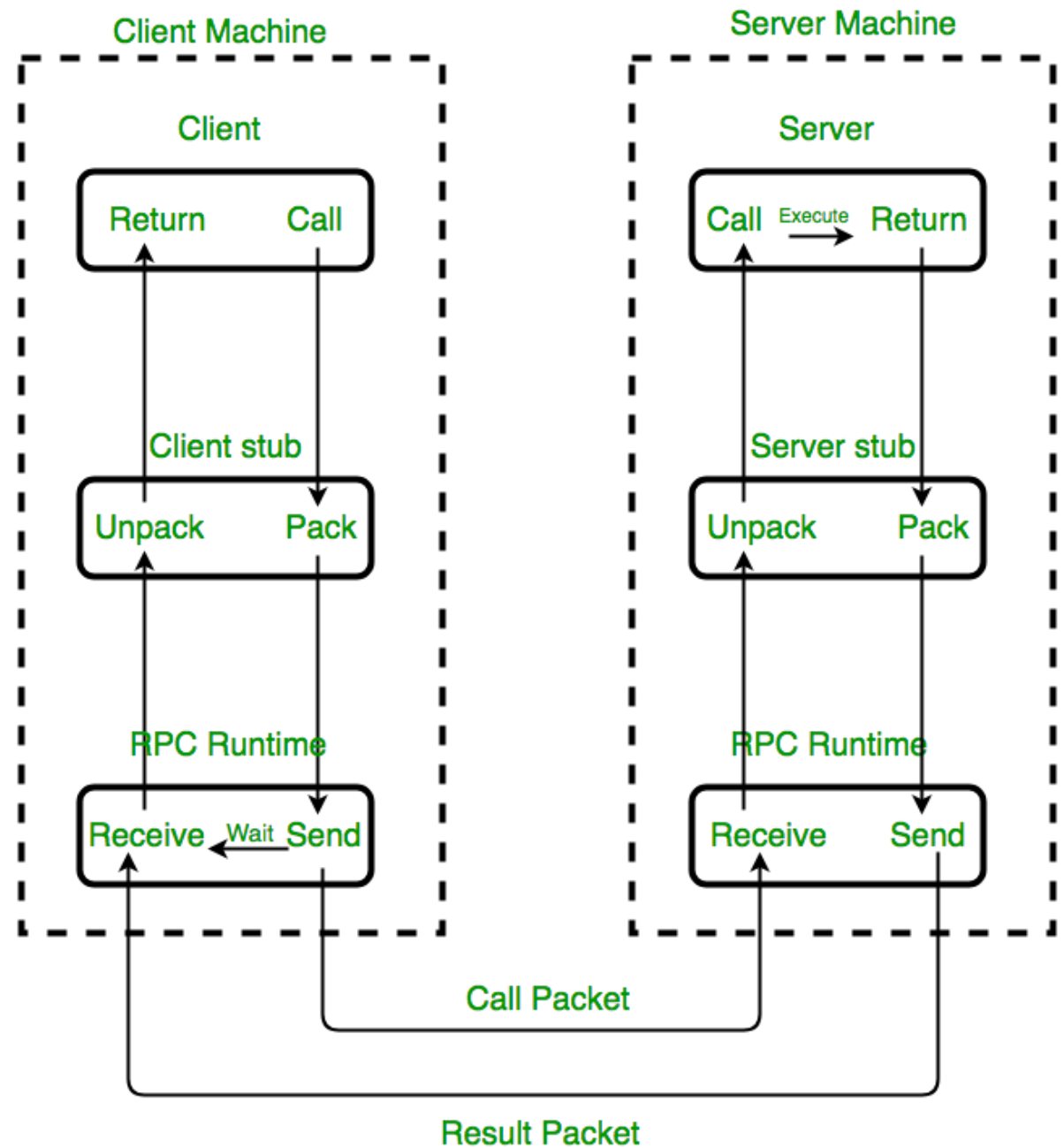
- gRPC는 google사에서 개발한 오픈소스 RPC프레임워크
- PB(protocol buffer) + HTTP/2 를 결합한 형태
- REST와의 차이점 :
 - 기반 기술이 다르다
 - HTTP/2를 사용한다
 - Protocol buffer를 데이터로 전달한다.

gRPC 서비스를 만드는 방법

1. Protocol buffer(.proto)파일에 서비스와 전달한 payload 구조를 정의합니다.
2. gRPC코드를 .proto파일에 생성합니다.
3. 서버에 구현합니다.
4. 클라이언트에 service를 실행할 수 있는 stub을 만듭니다.
5. 클라이언트와 서버를 실행합니다.

Stub? Protocol Buffer?

- Stub : 함수 호출에 사용되는 매개 변수를 변환함
 - Server stub : 클라이언트가 전달한 매개 변수의 역변환(언마샬링) 및 함수 실행 결과 변환을 담당
 - Client stub : 함수 호출에 사용된 파라미터의 변환(마샬링) 및 함수 실행 후 서버에서 전달된 결과를 변환



Implementation of RPC mechanism

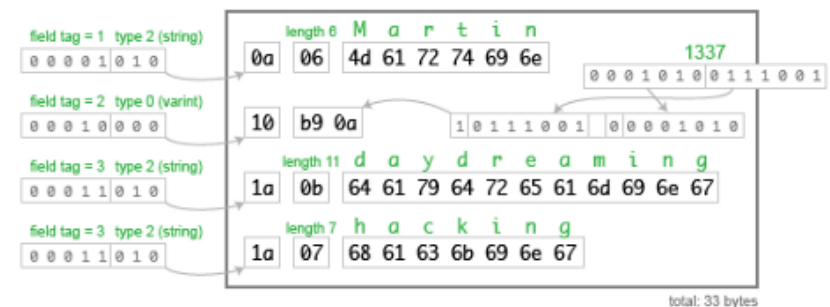
Stub? Protocol Buffer?

- Json 이나 xml과 같은 직렬화 데이터 구조
- 다른 구조에 비해 데이터를 크기가 작다.
- 5bit 로 field tag 숫자로 표현
- 3bit로 type 명시
- 그다음 데이터의 길이 나타냄

```
{  
  "userName": "Martin",  
  "favouriteNumber": 1337,  
  "interests": ["daydreaming", "hacking"]  
}
```

```
message Person {  
  required string user_name      = 1;  
  optional int64  favourite_number = 2;  
  repeated string interests      = 3;  
}
```

Protocol Buffers



장점?단점

- 장점
 - 빠름
 - 언어의 구매를 안받는다
- 단점
 - 인간이 읽기 부적합함 (프로토 파일 없으면)
 - 러닝커브
 - 브라우저 사용이 어려움

Quiz

- gRPC와 HTTP API(REST)와의 차이점 두가지를 말해주세요