

# 微服务框架开发一

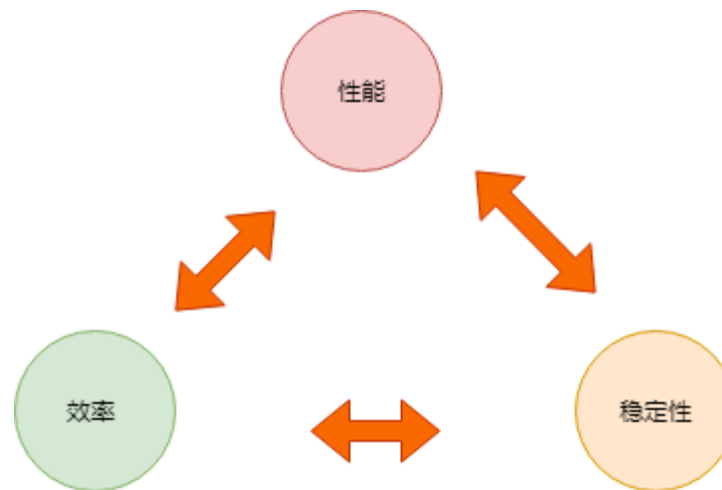
联系QQ: 2816010068, 加入会员群

# 目录

- 微服务框架整体架构
- http/2介绍
- grpc介绍

# 微服务框架整体架构

- 微服务框架的意义
  - 性能
    - 专业的人做专业的事情
    - 专注框架层面的优化
  - 稳定性
    - 专门的团队进行维护
    - Bug快速fix
  - 效率
    - 解决共性问题
    - 提高代码复用粒度



# 微服务框架整体架构

- 开发过程中的共性问题

- 客户端调用（rpc：Remote Procedure Call）

RPC（Remote Procedure Call）—远程过程调用，它是一种通过网络从远程计算机程序上请求服务，而不需要了解底层网络技术的协议。RPC协议假定某些传输协议的存在，如TCP或UDP，为通信程序之间携带信息数据。在OSI网络通信模型中，RPC跨越了传输层和应用层。RPC使得开发包括网络分布式多程序在内的应用程序更加容易。

- 网络通信

- 连接管理

- 建立连接
      - 发送数据包
      - 接收数据包
      - 超时控制
      - TCP粘包问题解决

- 连接池

- 数据传输问题

- 序列化：本地数据结构转成网络数据[字节流]
    - 反序列化：网络数据[字节流]转成本地数据结构

# 微服务框架整体架构

- 开发过程中的共性问题
  - 客户端调用（rpc: Remote Procedure Call）
    - 负载均衡
      - 随机算法
      - 轮询算法
      - 加权随机算法
      - 加权轮询算法
      - 一致性哈希算法
    - 服务发现
      - 通过服务名发现服务的位置（ip列表）
  - 服务框架
    - 网络通信
      - 连接管理
        - 超时控制
    - 数据传输
      - 序列化
      - 反序列化

# 微服务框架整体架构

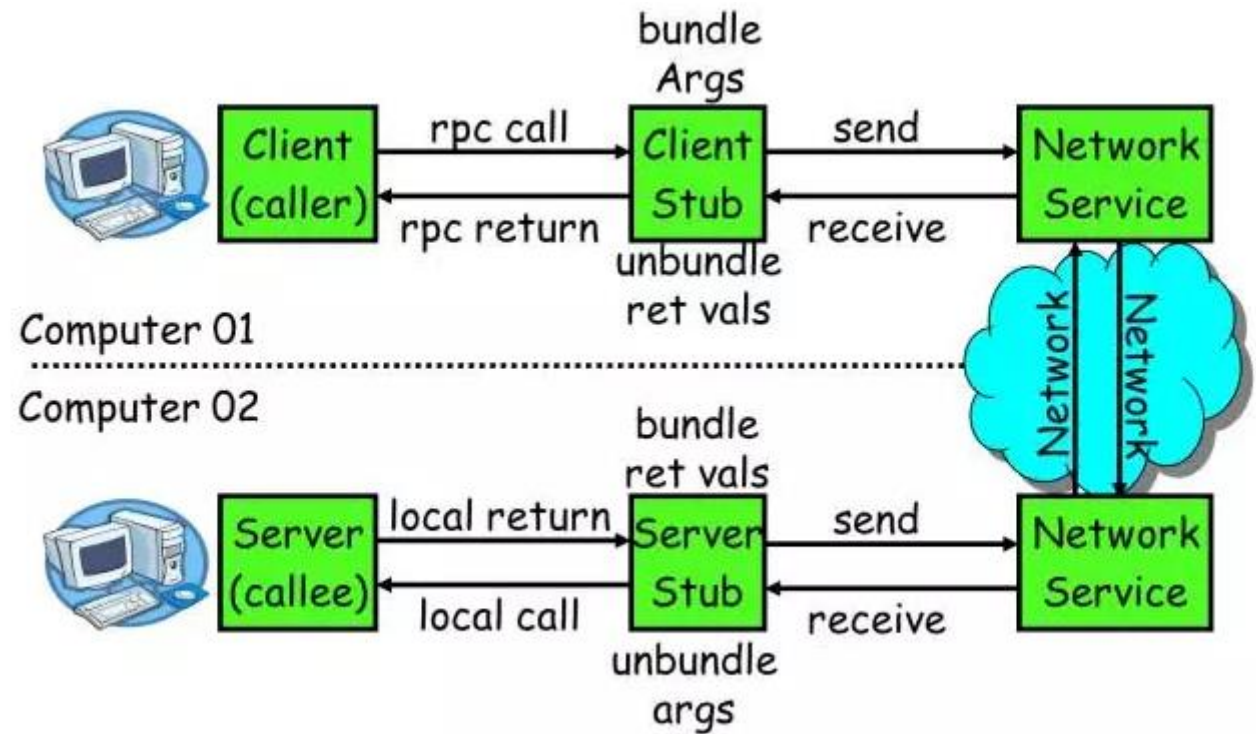
- 开发过程中的共性问题
  - 服务框架
    - 请求路由
      - 通过客户端调用的方法，路由到业务对应的处理函数
    - 日志组件
      - 服务访问记录
  - 异常处理
    - 限流和熔断
      - 限流策略
        - 令牌桶算法
      - 熔断策略
        - 电路熔断策略
    - 自动摘除
      - 服务注册、发现=》自动摘除

# 微服务框架整体架构

- 开发过程中的共性问题
  - 服务治理
    - 分布式追踪系统
      - Dapper论文
      - Zipkin: <https://zipkin.apache.org/>
      - Jaeger: <https://www.jaegertracing.io/>
      - Opentrace: <https://opentracing.io/>
    - 服务监控
      - Metrics采样打点
        - Prometheus
        - Grafana 展示

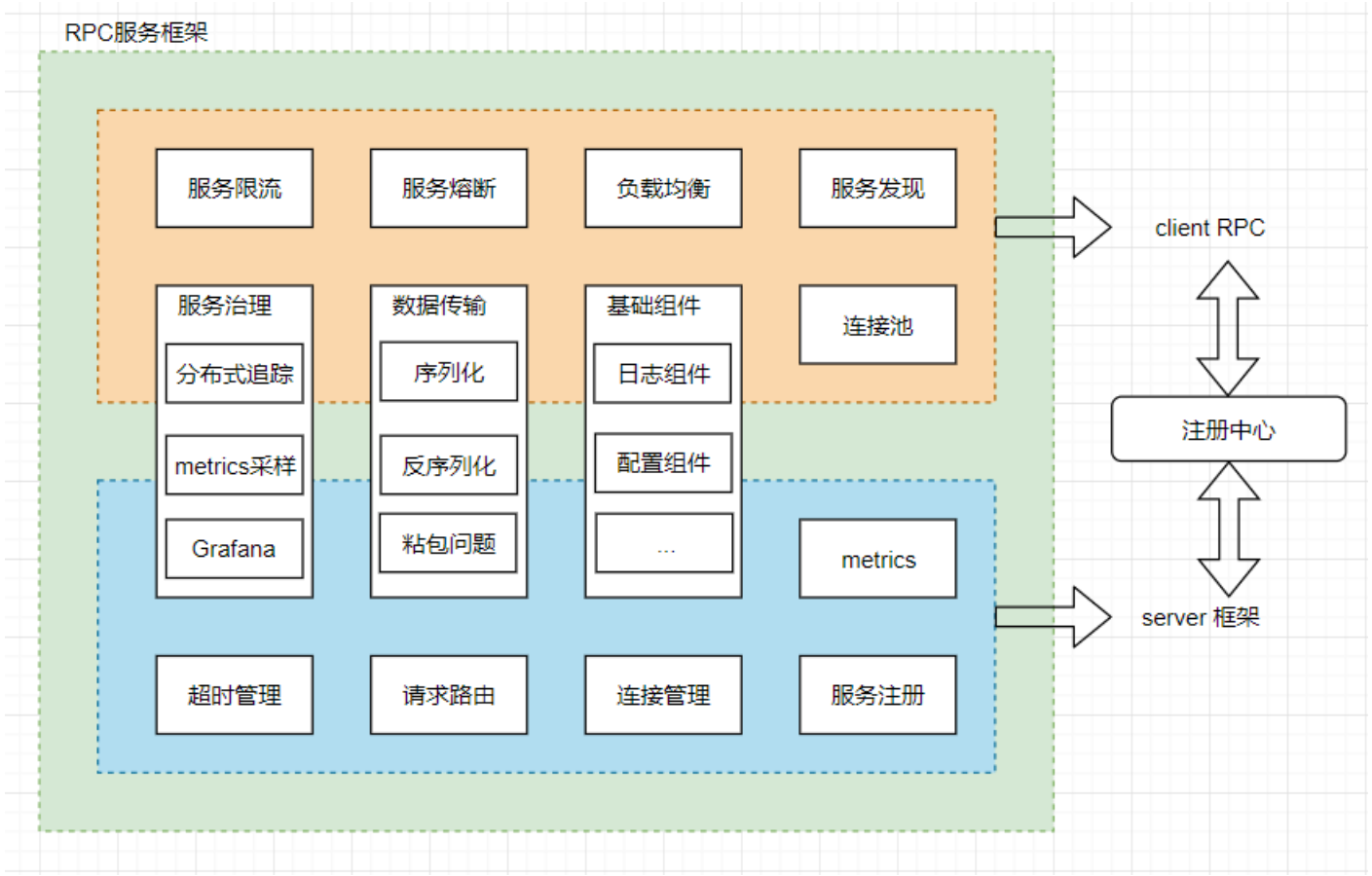
# 微服务框架整体架构

- 微服务框架交互流程





# 微服务框架整体架构

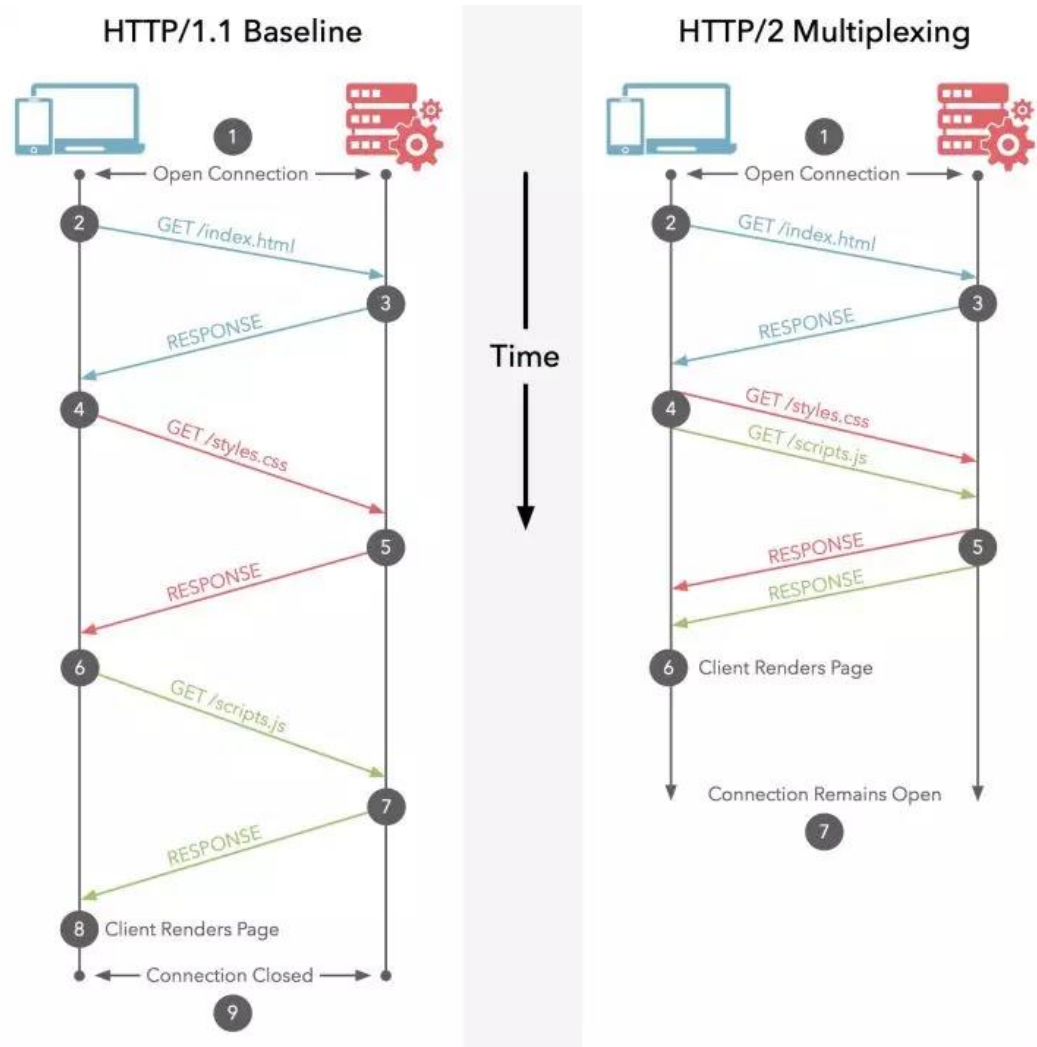


# Http2介绍

- 在http1.1基础上做了大量的改进
  - 多路复用
  - 二进制分帧
  - 首部压缩
  - 服务器推送

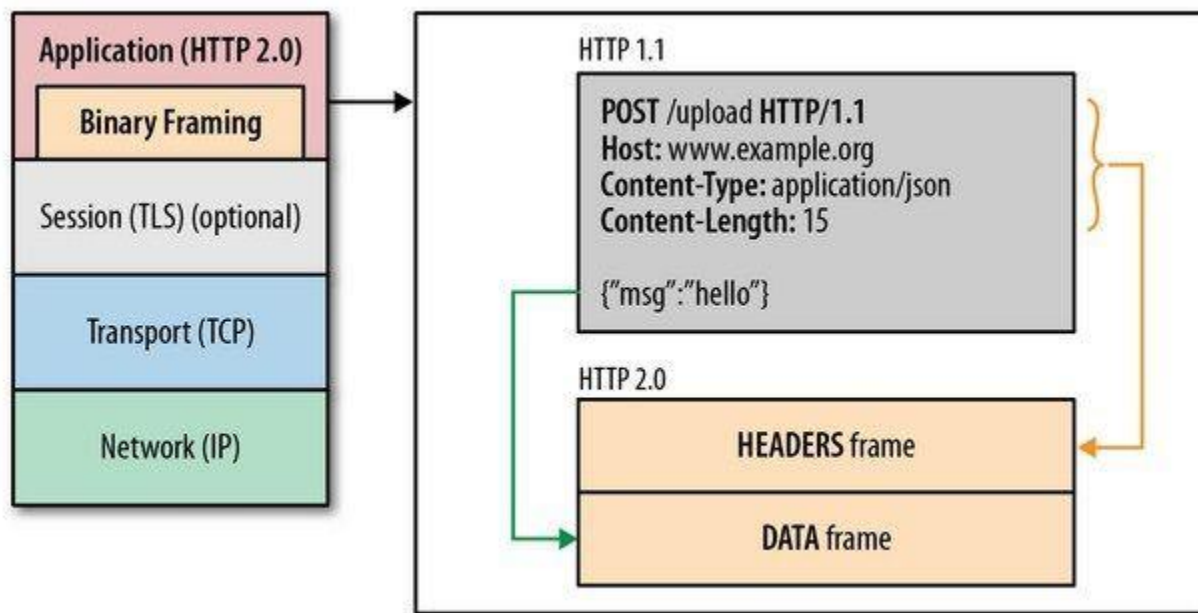
# Http2介绍

- 多路复用



# Http2介绍

- 二进制分帧



# Http2介绍

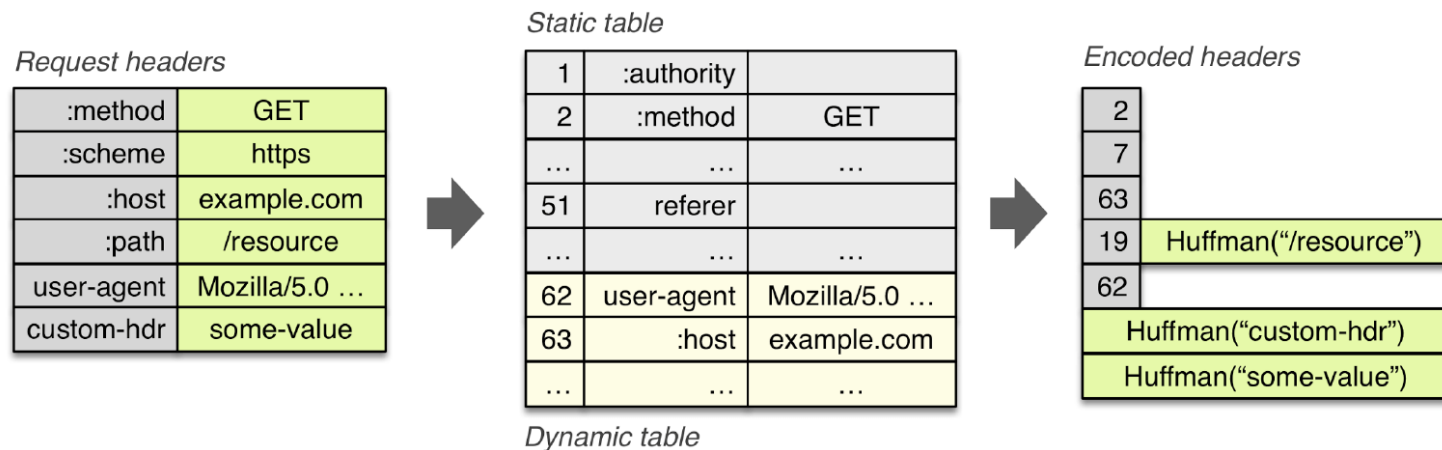
- 头部压缩
  - 全部头部定义

Index	Header Name	Header Value
1	:authority	
2	:method	GET
3	:method	POST
4	:path	/
5	:path	/index.html
6	:scheme	http
7	:scheme	https
8	:status	200
...	...	...
32	cookie	
...	...	...
60	via	
61	www-authenticate	

# Http2介绍

- 头部压缩

## HPACK header compression

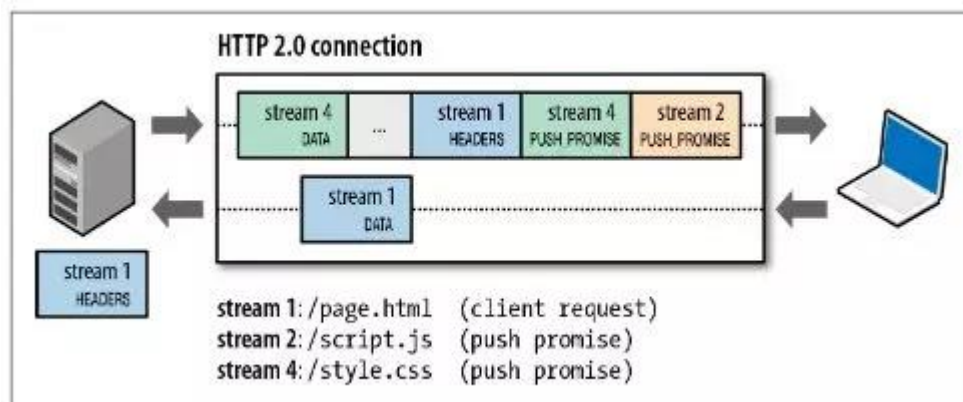


- Literal values are (optionally) encoded with a static Huffman code
- Previously sent values are (optionally) indexed
  - e.g. "2" in above example expands to "method: GET"



# Http2介绍

- 服务器推送



# Http2介绍

- Go只要使用https, 默认支持http2
  - 如何兼容
    - 通过客户端协商解决, 协商算法: ALPN

No.	Time	Source	Destination	Protocol	Length	Info
93	14.940515	10.18.60.26	114.215.116...	TLSv1...	254	Client Hello
94	14.942239	123.151.10...	10.18.60.26	SSL	349	Continuation Data
97	14.970500	114.215.11...	10.18.60.26	TLSv1...	1514	Server Hello
99	14.970505	114.215.11...	10.18.60.26	TLSv1...	1347	Certificate
102	14.971300	10.18.60.26	114.215.116...	TLSv1...	180	Client Key Exchange, Change Cipher Spe...
103	14.987435	114.215.11...	10.18.60.26	TLSv1...	312	New Session Ticket, Change Cipher Spec...
104	14.987437	114.215.11...	10.18.60.26	TLSv1...	123	Application Data
111	15.135321	10.18.60.26	114.215.116...	TLSv1...	107	Application Data
112	15.135370	10.18.60.26	114.215.116...	TLSv1...	104	Application Data
▶ Extension: next_protocol_negotiation						
▶ Extension: signed_certificate_timestamp						
▼ Extension: Application Layer Protocol Negotiation						
Type: Application Layer Protocol Negotiation (0x0010)						
Length: 23						
ALPN Extension Length: 21						
▼ ALPN Protocol						
ALPN string length: 2						
ALPN Next Protocol: h2						
ALPN string length: 8						
ALPN Next Protocol: spdy/3.1						
ALPN string length: 8						
ALPN Next Protocol: http/1.1						
▶ Extension: channel_id						
00c0	00 05 01 00 00 00 33	74 00 00 00 12 00 00 00	.....3 t.....			
00d0	10 00 17 00 15 02 68 32	08 73 70 64 79 2f 33 2e	.....h2 .spdy/3.			
00e0	31 08 68 74 74 70 2f 31	2e 31 75 50 00 00 00 0b	1.http/1 .luP....			
00f0	00 02 01 00 00 0a 00 06	00 04 00 17 00 18	.....			

<https://blog.csdn.net/nimasike>



# Http2介绍

- Go只要使用https, 默认支持http2
  - 如何兼容
    - 通过客户端协商解决, 协商算法: ALPN

No.	Time	Source	Destination	Protocol	Length	Info
93	14.940515	10.18.60.26	114.215.116...	TLSv1...	254	Client Hello
94	14.942239	123.151.10...	10.18.60.26	SSL	349	Continuation Data
97	14.970500	114.215.11...	10.18.60.26	TLSv1...	1514	Server Hello
99	14.970505	114.215.11...	10.18.60.26	TLSv1...	1347	Certificate
102	14.971300	10.18.60.26	114.215.116...	TLSv1...	180	Client Key Exchange, Change Cipher Spe...
103	14.987435	114.215.11...	10.18.60.26	TLSv1...	312	New Session Ticket, Change Cipher Spec...
104	14.987437	114.215.11...	10.18.60.26	TLSv1...	123	Application Data
111	15.135321	10.18.60.26	114.215.116...	TLSv1...	107	Application Data

▶ Extension: renegotiation\_info

▶ Extension: ec\_point\_formats

▶ Extension: SessionTicket TLS

▶ Extension: status\_request

▶ Extension: signed\_certificate\_timestamp

▼ Extension: Application Layer Protocol Negotiation

Type: Application Layer Protocol Negotiation (0x0010)

Length: 5

ALPN Extension Length: 3

▼ ALPN Protocol

ALPN string length: 2

ALPN Next Protocol: h2

0170	06 aa 00 10 00 05 00 03	02 68 32 16 03 03 0a af	..... .h2....
0180	0b 00 0a ab 00 0a a8 00	06 0c 30 82 06 08 30 82	..... ..0...0.
0190	04 f0 a0 03 02 01 02 02	12 03 0b a8 05 c0 57 29	..... .....W)
01a0	8e a3 e2 86 d4 d3 5e ff	9a 1f 08 30 0d 06 09 2a	.....^...0...*
01b0	86 48 86 f7 0d 01 01 0b	05 00 30 4a 31 0b 30 09	.H..... ..0J1.0.
01c0	06 03 55 04 06 13 02 55	53 31 16 30 14 06 03 55	..U....U S1.0...U
01d0	04 0a 13 0d 4c 65 74 27	73 20 45 6e 63 72 79 70	....Let' s Encryp

<https://blog.csdn.net/nimasike>

# grpc介绍

- 基于http2的rpc框架
- IDL使用protobuf
  - 先使用idl描述服务接口
  - 通过工具自动生成代码框架
  - 业务使用接口
- 多语言支持
  - Go
  - Java
  - C++

# grpc介绍

```
1  syntax = "proto3";
2
3  package pb;
4
5  // The Add service definition.
6  service Add {
7      // Sums two integers.
8      rpc Sum (SumRequest) returns (SumReply) {}
9
10     // Concatenates two strings
11     rpc Concat (ConcatRequest) returns (ConcatReply) {}
12 }
13
14 // The sum request contains two parameters.
15 message SumRequest {
16     int64 a = 1;
17     int64 b = 2;
18 }
19
20 // The sum response contains the result of the calculation.
21 message SumReply {
22     int64 v = 1;
23     string err = 2;
24 }
25
```