

# Protobuf分享

By 袁杰威

# protobuf

- Protocol buffers 是一个语言中立，平台无关的，可拓展的序列化数据格式
  - 传输协议
  - IDL
  - 存储格式
- 优势
  - 小&快
  - 多语言自动生成代码

# 关键技术-varints

- 每个字节使用其中7位保存数字，最高位表示后面是否还有内容
- 低位在前，高位在后
- 保留fixed32和fixed64，用于传递大整数
- Int32,int64,uint32,uint64,bool序列化结果相互兼容，可以修改

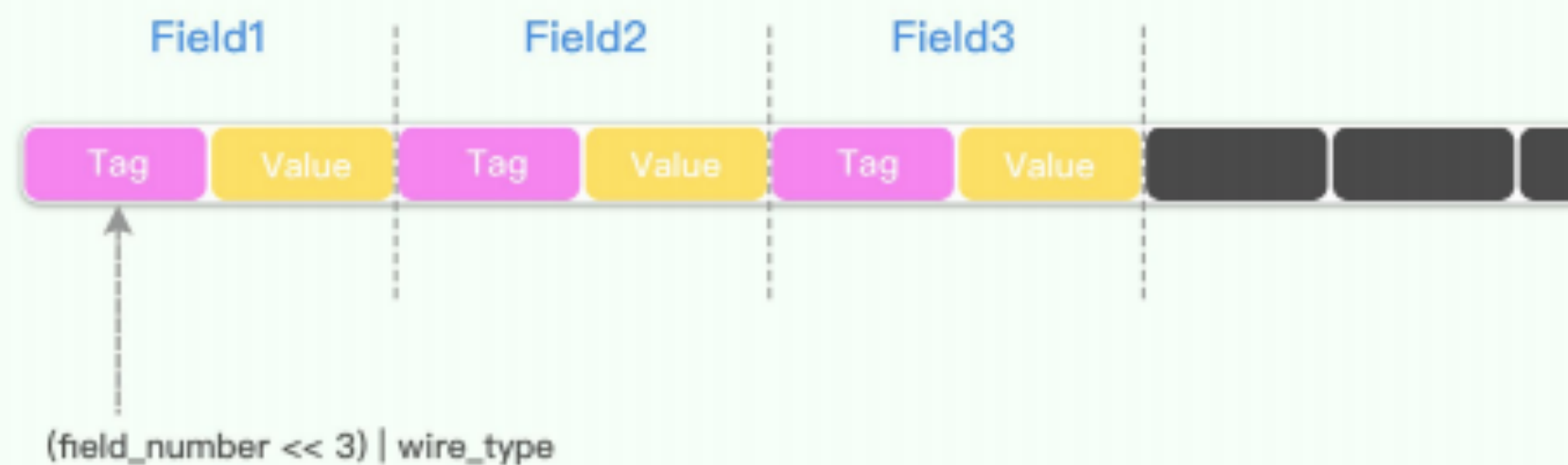
# 关键技术-zigzag编码

- 传统上，负数最高位为1，小负数会浪费编码长度
- $(n \ll 1) \wedge (n \gg 31)$
- -1 会被变成编码1， 1会被编码成2，绝对值小的负数编码后还是较小的整数

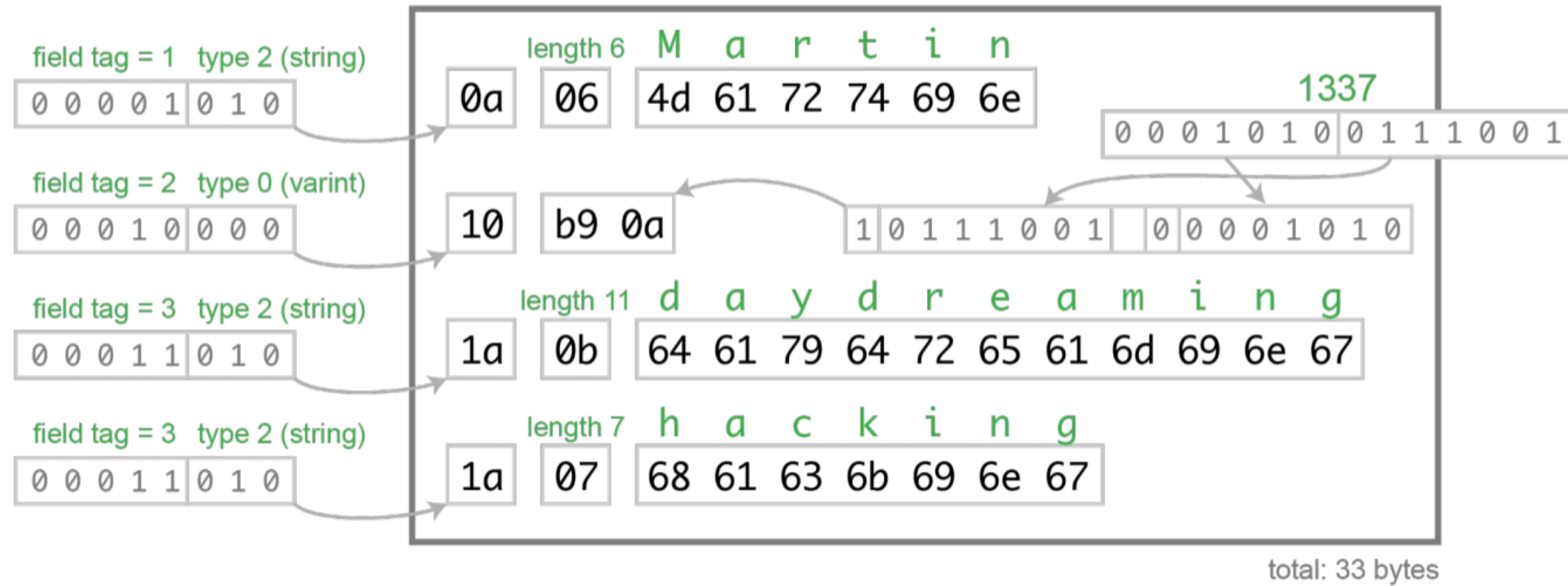
# 关键技术-message structure编码

- Tag-Value编码
- $\text{Tag} = (\text{field\_number} \ll 3) \mid \text{wire\_type} \rightarrow \text{varints}$
- wire\_type: 0-varints , 1-fixed64, 5-fixed32
- wire\_type: 2-TLV(Tag-Length[varints]-Value编码)
- string bytes, message 嵌套, 都是用TLV编码
-

## Message Structure



## Protocol Buffers



# 关键技术-repeated编码

- 第一种方式: 重复出现的相同tag
- 第二种编码方式(packedk=true), TLVV...编码
- 只有数字类型才可以使用第二种方法, pb3默认使用第二种, pb2需要显示指定, 第一种任何情况都会被支持
- 非repeated情况出现重复tag 后面的覆盖前面的因此 optional和repeated相互兼容



# 关键技术Map编码

- `map<key_type,value_type> map_field = N;`
- 序列化等价于：

```
message MapFieldEntry{  
    key_type key = 1;  
    value_type value = 2;  
}  
repeated MapFieldEntry map_field = N ;
```

# protoc编译器

- C++编写的proto文件编译器
- 支持各种语言的插件

# 编码规范

- 命名: 自己去看文档呗

# RPC框架

- gRPC
- Thrift
- ...