列举一些大厂使用的微服务框架，基本上都有工具链，生成代码、docker、k8s部署文件等：

1.字节跳动：kitex，开源

2.滴滴：有自己的一套微服务框架，未开源

3.腾讯：TarsGo，开源，并且是Linux基金会项目

4.B站：kratos，开源

5.斗鱼：Jupiter，开源

6.好未来：go-zero，开源

7.阿里：dubbo-go，开源

8.华为：Go Chassis，开源

9.知乎：内部改用go重构后端..

作者：相守之路  
链接：https://www.zhihu.com/question/386183447/answer/1578400419  
来源：知乎  
著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

建议放弃一代微服务框架如dubbo、go-micro之类的玩意儿，直接上service mesh，或者不上mesh干用k8s一时半会也没啥大问题。

<https://github.com/bytedance/gopkg>

# Thrift协议

Thrift是一种接口描述语言和二进制通讯协议。原由Facebook于2007年开发，2008年正式提交Apache基金会托管，成为Apache下的开源项目。

​Thrift是一个RPC通讯框架，采用自定义的二进制通讯协议设计。相比于传统的HTTP协议，效率更高，传输占用带宽更小。另外，Thrift是跨语言的。Thrift的接口描述文件，通过其编译器可以生成不同开发语言的通讯框架。

原文链接：<https://blog.csdn.net/Lambert_Wang/article/details/113627067>

# CloudweGo

<https://github.com/cloudwego/kitex/blob/develop/README_cn.md>

联机事务处理OLTP（on-line transaction processing）

OLTP是传统的关系型数据库的主要应用，主要是基本的、日常的事务处理，例如银行交易。

联机分析处理OLAP（On-Line Analytical Processing）

OLAP是数据仓库系统的主要应用，支持复杂的分析操作，侧重决策支持，并且提供直观易懂的查询结果。

下表列出了OLTP与OLAP之间的比较：

|  |  |  |
| --- | --- | --- |
|  | OLTP | OLAP |
| 用户 | 操作人员，低层管理人员 | 决策人员，高级管理人员 |
| 功能 | 日常操作处理 | 分析决策 |
| DB 设计 | 面向应用 | 面向主题 |
| 数据 | 当前的， 最新的细节的， 二维的分立的 | 历史的， 聚集的， 多维的  集成的， 统一的 |
| 存取 | 读/写数十条记录 | 读上百万条记录 |
| 工作单位 | 简单的事务 | 复杂的查询 |
| DB 大小 | 100MB-GB | 100GB-TB |

总的来说，OLTP就是面向我们的应用系统数据库的，OLAP是面向数据仓库的。

<https://zhuanlan.zhihu.com/p/408262736>

<https://www.cloudwego.io/docs/getting-started/>

<https://github.com/bytedance/gopkg>

# Go-micro

项目简介：Micro是一个专注于简化分布式系统开发的微服务生态系统。可插拔的插件化设计，提供强大的可插拔的架构来保证基础组件可以被灵活替换。

仓库地址：https://github.com/micro/micro（10.7K）

官方文档地址：https://micro.mu/

原文链接：https://blog.csdn.net/qq2942713658/article/details/112721577

我认为最流行的框架之一。有很多博客文章和简单的例子。您可以在 medium 上关注 microhq 或 @MicroHQ 以获取 Go-Micro 中的最新更新。

好吧，什么是 Go Micro？它是一个可插入的 RPC 框架，用于在 Go 中编写微服务。开箱即用，您将收到：

服务发现 - 应用程序自动注册到服务发现系统。

负载平衡 - 客户端负载平衡，用于平衡服务实例之间的请求。

同步通信 - 提供请求 / 响应传输层。

异步通信 - 内置发布 / 订阅功能。

消息编码 - 基于消息的内容类型头的编码 / 解码。

RPC 客户机 / 服务器包 - 利用上述功能并公开接口来构建微服务。

Go 微体系结构可以描述为三层堆栈。

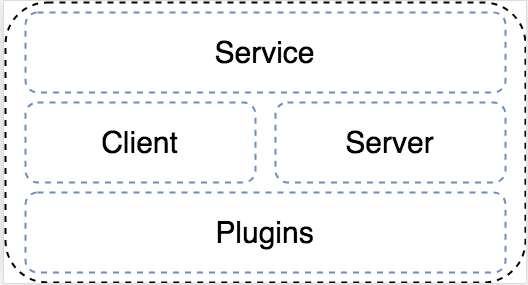


图 1. Go Micro 架构

顶层由客户端 - 服务器模型和服务抽象组成。服务器是用于编写服务的构建块。客户端提供了向服务请求的接口。

底层由以下类型的插件组成：

代理 - 为异步发布 / 订阅通信提供消息代理的接口。

编解码器 - 用于编码 / 解码消息。支持的格式包括 json，bson，protobuf，msgpack 等。

注册表 - 提供服务发现机制（默认为 Consul）。

选择器 - 建立在注册表上的负载平衡抽象。它允许使用诸如随机，轮循，最小康等算法来 “选择” 服务。

传输 - 服务之间同步请求 / 响应通信的接口。

Go Micro 还提供了 Sidecar 等功能。这使您可以使用以 Go 以外的语言编写的服务。 Sidecar 提供服务注册，gRPC 编码 / 解码和 HTTP 处理程序。它支持多种语言。

————————————————

原文作者：Summer

转自链接：https://learnku.com/go/t/36973

# istio

项目简介：Istio是由Google、IBM和Lyft开源的微服务管理、保护和监控框架。使用istio可以很简单的创建具有负载均衡、服务间认证、监控等功能的服务网络，而不需要对服务的代码进行任何修改。

仓库地址：https://github.com/istio/istio（28.8K）

官方文档地址：https://istio.io/latest/docs/

————————————————

版权声明：本文为CSDN博主「深漂小码哥」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/qq2942713658/article/details/112721577>

# MOSN

<https://github.com/mosn/mosn>

<https://skyao.io/post/202004-servicemesh-and-api-gateway/>

https://skyao.io/post/201804-servicemesh-architecture-introspection/

https://mp.weixin.qq.com/s/dm60z\_7soeIM-Wc7fU06OQ