

个人简介

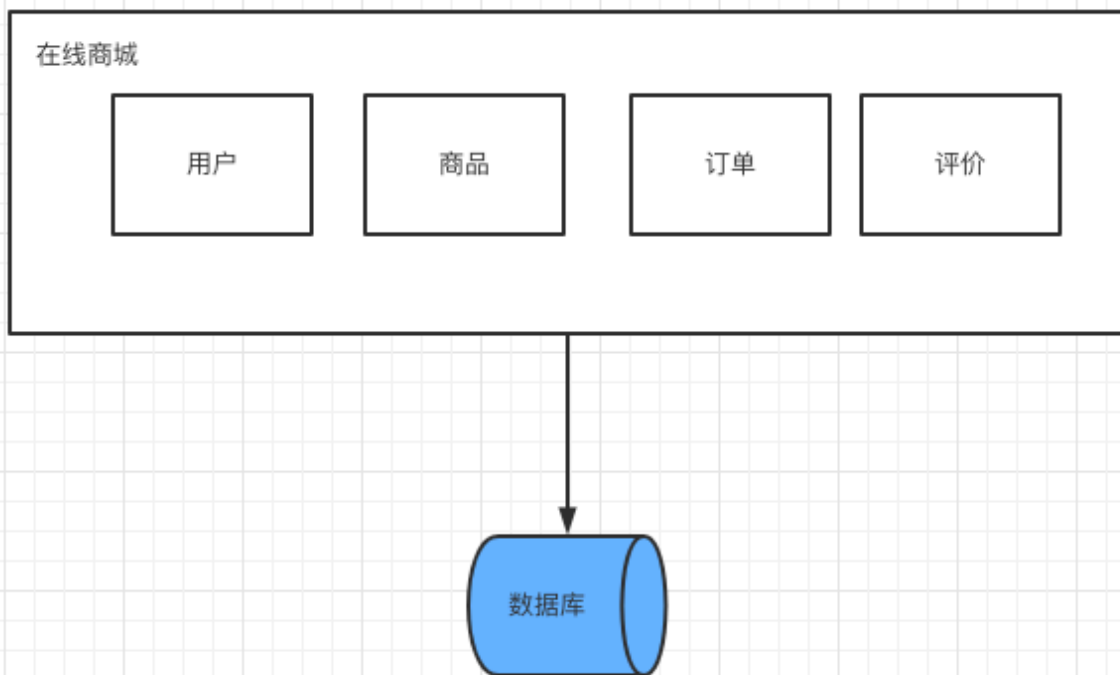
- name:曾阿牛
- 全栈开发工程师【目前gopher】
 - 后端：
 - php:laravel
 - nodejs:eggjs
 - golang:gin
 - 前端：
 - vuejs
 - jquery,es6常用特性

本次演示用到工具

1. Typora:笔记工具
2. Docker:安装mysql,redis
3. Mysql:数据库
4. Sequel pro:mac下 mysql管理工具
5. Redis:缓存数据库
6. Rdm:mac下redis可视化工具
7. Goland:开发ide
8. Prometheus:数据采集
9. Grafana:数据可视

内容简介

1. 微服务的介绍
 - 传统单体架构
 - 所有功能集成在一个项目中
 - 项目整个打包，可以部署到服务器运行
 - 应用与数据可以分开部署，提高性能
 - 优点：
 - 小项目首选
 - 缺点：
 - 项目复杂后，难扩展与维护
 - 扩展成本高，有瓶颈
 - 技术栈受限



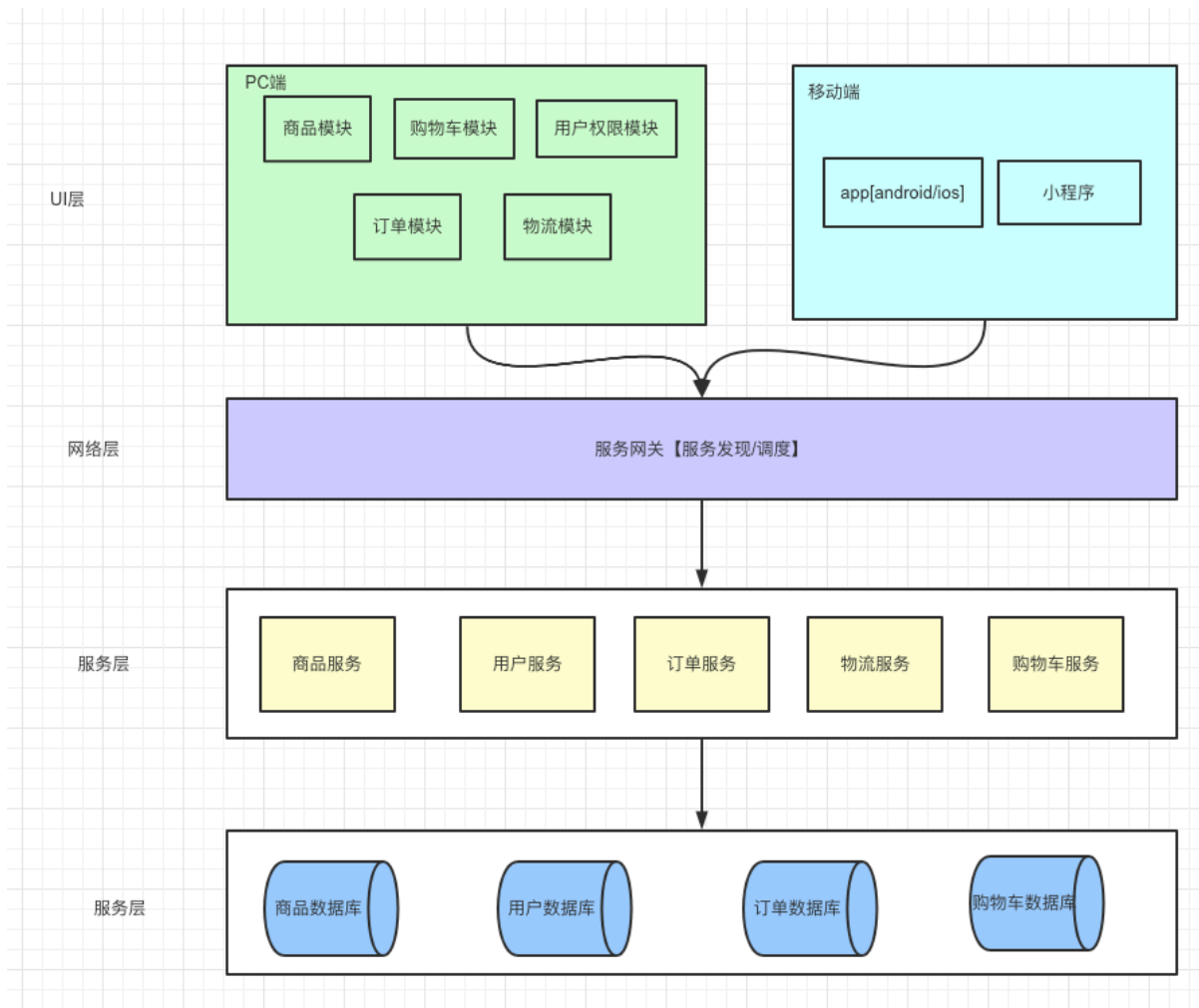
○ 微服务架构

■ 微服务优点：

- 独立性
- 使用者容易理解
- 技术栈灵活(java/golang/php/rust/c/c++)
- 高效团队

■ 微服务架构的不足

- 额外的工作，服务的拆分
- 保证数据一致性
- 增加了沟通成本
- 服务治理成本高

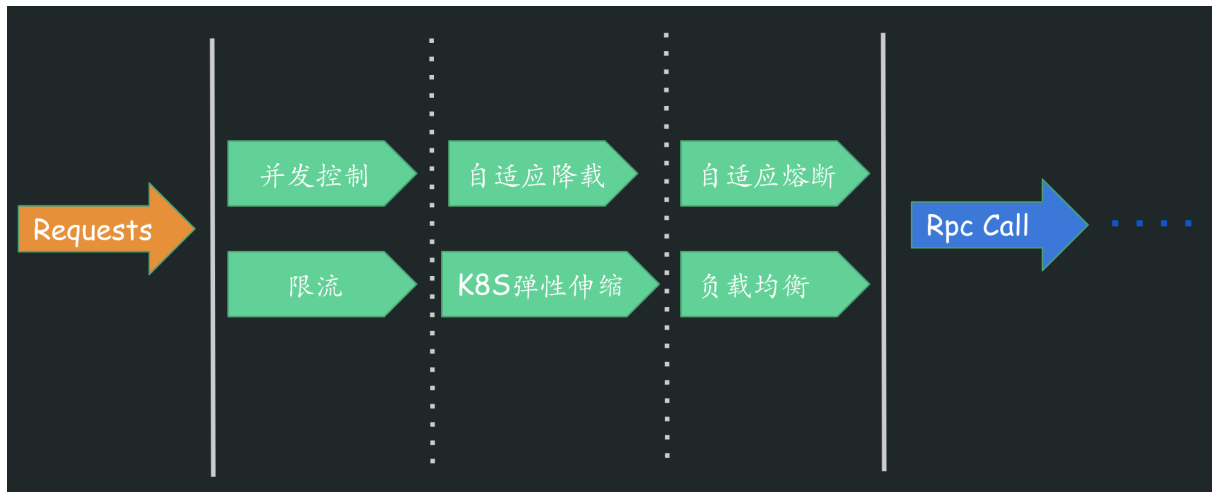


2. go-zero

go-zero 是一个集成了各种工程实践的包含 web 和 rpc 框架，有如下主要特点【收益】：

- 强大的工具支持，尽可能少的代码编写
- 极简的接口
- 完全兼容 net/http
- 支持中间件，方便扩展
- 高性能
- 面向故障编程，弹性设计
- 内建服务发现、负载均衡
- 内建限流、熔断、降载，且自动触发，自动恢复
- API 参数自动校验
- 超时级联控制
- 自动缓存控制
- 链路跟踪、统计报警等
- 高并发支撑，稳定保障了疫情期间每天的流量洪峰

如下图，我们从多个层面保障了整体服务的高可用：



觉得不错的话，别忘 **star** 🙌

1. go-zero介绍文档 <https://github.com/tal-tech/go-zero>
2. 参考文档 <https://github.com/tal-tech/zero-doc/blob/main/doc/shorturl.md>

微服务指标监控

- Prometheus

1. 参考文档 <https://gocn.vip/topics/10961>
2. 从零搭建Prometheus监控报警系统 <https://www.cnblogs.com/chengqionghe/p/10494868.html>
 1. 官网 <https://prometheus.io/>
 2. 下载对应版本
 3. 运行

```
./prometheus --config.file=prometheus.yml  
// 验证是否启动 http://localhost:9090/targets
```

4. metrics <http://localhost:9090/metrics>
5. graph <http://localhost:9090/graph>
3. 查询语法参考
 - <https://www.jianshu.com/p/3bdc4cfa08da>

- Grafana

```
// 1 安装  
brew install grafana  
// 2 运行  
brew services restart grafana  
// 3 浏览器打开  
// 4 admin admin
```

测试

```
wrk -t10 -c1000 -d30s --latency "http://127.0.0.1:8888/check?book=go-zero"
```

go-zero 演示

- 演示基于官方文档 <https://github.com/tal-tech/zero-doc/blob/main/doc/shorturl.md>
- 启动 API Gateway 服务，默认侦听在 8888 端口

```
cd /Users/rongqiangqin/Desktop/project/golang/shorturl/api
go run shorturl.go -f etc/shorturl-api.yaml
```

- 启动rpc

```
cd /Users/rongqiangqin/Desktop/project/golang/shorturl/rpc/transform
go run transform.go -f etc/transform.yaml
```

- query

```
//由于我grafana版本没出数据,
sum(rate(http_server_requests_duration_ms_count{app="$app",mode="$env"}
[5m]))by(path)
{path="/shorten"}
sum(rate(http_server_requests_duration_ms_count{job="shorturl-api"}
[5m]))by(path)
```

一些注意点

0. 提前安装etcd,mysql,redis;并确保正常运行和正常连接
1. redis配置密码
2. 生成model文件需要简单修改

下一步练习《做一个开源小工具》

基于微服务，开发一个日程安排功能，功能包括：

1. 前端以日历方式显示
2. 在日历上可添加事件、修改事件、删除事件
3. 代码仓库创建 <https://github.com/xiaopenggithub/schedule>

结语

1. 案例演示相对简单，但也包括了微服务开发的整个流程。
2. 在以后工作场景中继续用到微服务架构开发模式，总结更多经验。
3. 个人微信(备注go-zero)



扫一扫上面的二维码图案，加我微信

