

GOLANG 后台研发-赵轩超

(+86) 177-1164-4027 · ffzxc.do@gmail.com · GitHub @zput

个人信息

- 赵轩超/男/1993-12
- 本科/湖南理工学院/自动化 (2012.09-2016.06)
- 个人技术博客:<https://zput.github.io>
- Github:<https://github.com/zput>

工作经历

深圳乐天童创科技有限公司, 平台架构部 golang 高级开发工程师 2019.7-至今

- 企业上云, 整合服务, 上 k8s。整理前有十几个 beego 独立的微服务, 逻辑混乱, 前端调用的接口 URI 相同前缀但不同后缀, 可能各自分布在几个微服务中, 无法进行负载均衡, 扩容分散流量, 作为发起者, 重构服务, 商讨路由整合, 缩减微服务到 7 个, 并编写 helm, 服务都上到 k8s。性能从 10QPS 提升到 2000QPS, 测试加正式服务器由 10 台减少到 5 台。
- 规划编写服务 CI/CD 基于 gitlab 自动发布 k8s 整个流程。包括.gitlab-ci.yml 的编写, gitlab-runner 部署以及它默认 docker 等。当自动发布上线后, 开发打包部署到交给测试人员测试, 从至少 15 分钟, 降低到 5 分钟内。
 - https://zput.github.io/2020/02/25/tool/gitlab/conclusion_gitlab_ci_cd/
 - https://zput.github.io/2020/01/22/tool/gitlab/the_process_structure_ci_cd_base_on_gitlab/
- 构建基于 go-micro 后台微服务框架。为适应团队新的开发流程, 和兼容以前的服务与小程序, 对外提供了 grpc 与 grpc-gw(json) 两个服务, 内部模块分为 API 层, work 层, core 层, 内部服务交流使用 grpc。参与整个框架制定, 实现对外提供的两个服务 (grpc, grpc-gw) 与 core 层等, 开发调研服务周边的设施 (log, token, sql driver)。缩短与前端对接时间, 定好 *.proto 协议, 即可分发给个模块人员。
- 参与公司两条新产品 AI 课, 语文课的开发。领导参与的主要模块包括登录相关, 权限相关, 课程, 题库, 学生排课, 作业, 公众号提醒与消息推送等, 学生人数从 0 到 3.5 万人增长。现在为这两条线的后台项目负责人, 开发新需求与维护线上服务稳定。
- 负责研发销售系统, 期间开发销售个人客户关系维护, 电话推广拨打系统, 提高销售人员工作效率 3 倍以上。

深圳云之梦科技有限公司, 研发部开发工程师 2018.9-2019.7

- 开发小程序, 微信公众号, 淘宝内的 H5 架构平台。
- 构建开放平台, 为商户提供统一公司服务的接口。
- 为外部商家提供总店门店操作, 会员注册, 衣服商品上下架, 提供与线下设备传输交流打通 3D 试衣等功能等开放平台的 API。

深圳优科数字化制造有限公司, 研发工程师 2016.6-2018.9

- 参与基于工厂改造, 生产数据采集, 开发工业软件的数据通信, 不仅负责与 AGV 等智能设备数据连接收集, 还打通了工厂看板与移动端小程序的数据展示, 较大提高甲方验收工厂数字化转型满意度。
- 负责智能咖啡机项目, 从手机点餐下单, 到数据传输到咖啡机机器人, 完成咖啡的制造, 咖啡完成提示, 有效结合了工业与信息的结合, 在展会中吸引大量潜在客户。

一些开源项目

- **innodb_view** 用于分析 MySQL 的 innodb 引擎中物理存储文件。(https://github.com/zput/innodb_view)
 - Innodb_view 是一个直接访问 MySQL InnoDB 存储引擎文件的 Golang 实现。通过命令行可以遍历所有已经使用的页, 分析每个页的类型; 分析 Inode page 页面组成; Index page 页面的组成等。此外, 这个项目可加深对 MySQL innodb 物理页面内部结构理解。
- **zput_net_golang** 基于事件驱动 (Reactor 模式) 的高性能, 非阻塞和轻量级网络框架, 不使用标准 golang 语言 net 网络包, 它的多路复用根据不同系统使用不同的系统函数 (epoll(linux 系统) 和 kqueue(FreeBSD 系统)), 轻松快速搭建高性能服务器。(https://github.com/zput/zput_net_golang)

- 非阻塞 I/O。
- 多 Goroutine 支持，每个 Goroutine 运行一个事件驱动的事件循环。
- 读写缓冲区使用可伸缩的环形缓冲区。
- 支持端口重用 (SO_REUSEPORT)。
- 支持事件定时任务。
- **ringbuffer** 多功能环形缓存，可配置加锁（线程安全）与不加锁（性能更好），当空间满自动扩展，可预先查看缓存中的数据，探索 API 的方式读等。(https://github.com/zput/ringbuffer)
 - 在 New 构造函数的时候，通过参数决定是加锁（线程安全）还是不加锁。
 - 当环形缓存空间满后，可以自动扩展内存。
 - 当使用探索类函数 (ExploreBegin()—ExploreRead()/ExploreSize()—ExploreCommit()/ExploreBreak())，可以预先探索缓存中的数据，最后可以决定是提交还是放弃。
- **zxc_netzxc_net** 是基于 Reactor 模式的多线程 C++ 网络库。(https://github.com/zput/zxc_net)
 - 依赖于 C++11 提供的 std::thread 库，多线程表现在：
 - * 为 accept socket 的线程中的 acceptFd 开启 SO_REUSEPORT 选项，多个线程拥有自己的 Acceptor 进行接收客户端连接。
 - * 为每个拥有 acceptFd 的线程建立了线程池，便于分发接收到的 clientFd。解决了 TCP 连接负载均衡的问题，减少客户连接等待时间。
 - * 非阻塞的 poll 模式，
 - 拥有 read/write buff 缓冲区，readv + LT 触发可以读取较大数据。

杂谈

- **golang 调度** https://zput.github.io/go-goroutine/
- **内部做的技术分享 PPT:**
 - **chan:** https://docs.google.com/presentation/d/1qONP8jtJacfSjFaKFSTi_5MMXNVuSomji3ytbSLKgts/edit?usp=sharing
 - **goroutine:** https://docs.google.com/presentation/d/1JoRSCB_UVHC1v3JpkyvM5CWfu2reFvxAXgD1LgvX/edit?usp=sharing

技能清单

- 编程语言: Go, C++, C, SQL, Shell
- 数据库相关: MySQL/PostgreSQL/Redis/MongoDB
- 消息队列: kafka
- 一些工程构建: kubernetes, docker, helm, gitlab, ci/cd, beego, go-micro, grpc