



王日昱

微信: i664929402

邮箱: wangriyu1997@gmail.com

resume

> Links

- Github: <https://github.com/wangriyu>
- Github Analysis: <https://hacknical.com/wangriyu/github?locale=en>
- Homepage: <https://home.wangriyu.wang>
- Blog: <https://blog.wangriyu.wang>
- Resume: <https://blog.wangriyu.wang/pages/resume>

> Education

华北电力大学(保定) – NCEPU

计算机系 – 软件工程

本科 2015 – 2019

> Profile

- 有两年两家公司的实习经验, 17 年以前端为主, 18 年转后台, 现在从事后台开发
- 熟悉团队作业, 合作分工明确, 熟悉产品设计、前端后台之间的交互、管理和沟通
- 得益于长期阅读文档和英文搜索的好习惯, **英语阅读理解能力尚佳**, 已过 CET6
- 具备程序员基本的职业技能和素养, **代码规范严谨**, 热爱编程, 喜欢电子产品, 关注前沿技术
- 工作时搜索只用谷歌, 别跟我提百度, 这辈子都不可能百度的, 资料找不齐全, 只能靠翻墙才能维持的了生活这样子, 翻墙的感觉好多了, 外面个个都是人才, 说话又好听, 我超喜欢的

> Skills

- **主力语言: Golang**, 副语言: JavaScript, 其他接触过的: Dart、kotlin、java、C++
- 前端技术栈以 **react**、react-native 为主
- 后端技术栈以 **Golang** 为核心。熟悉 **Golang** 的大部分中高级特性的使用方法, 了解一些底层实现, 比如切片、管道、内存管理、GC、map、sync.Map、http/server 等, 熟悉单元测试、基准测试、pprof 性能分析等工具
- 对网络协议栈较为了解, 对 HTTP 层较为熟悉, 曾详细研究过 **HTTP2**、**TLS(HTTPS)**, 记录在博客中; 对 TCP\UDP、Websocket 也算了解
- 数据库接触过 **Mysql**、**redis**、**ScyllaDB**, 但接触不多, 了解其底层数据结构
- 熟悉 **Linux** 环境和基本操作, 掌握基础的命令和 **VIM** 操作
- 了解分布式相关概念以及分布式事务, 有一定高并发经验, 熟悉多线程、锁、原子操作等
- 有一点基础架构经验; 熟悉 **RabbitMQ**、**Kafka** 消息中间件

> Experience

第一家公司是北京极智人科技有限公司旗下的技术猫编程俱乐部(微信小程序中可以搜索技术猫编程俱乐部), 参与的项目主要是内部前瞻性的尝试性项目, 期间接触并尝试过 Flutter、Electron、Beego、EggJS 等框架

► 前端开发 2017.02 – 2017.11(这块不感兴趣可以跳过)

这半年处于转型期间，开始学习 **Golang**，补充后端知识，阅读代码等，参与的项目以练手为主

1. 使用 *beego* 开发的小程序商城后台服务

- 负责商品、广告页、分类模块的编写
- 使用 **Beego** 创建 **MVC** 模型，使用 **gorm** 实现数据库访问和 **CRUD**
- 使用 **Mysql** 存储数据

2. *IP GEO Service*

- 使用 **maxmindDB** 提供的数据库编写的 **IP** 地理位置信息查询接口
- 支持数据库自动更新

3. 腾讯云 *COS SDK* 封装

- 使用腾讯云对象存储 **SDK** 编写简单易用的 **COS** 调用接口服务
- 支持创建删除桶对象、文件等
- 支持修改 **CORS** 等属性
- 支持获取文件、桶列表
- 支持上传下载文件

4. *nsq* 消息转发

- 使用 **nsq** 创建消息转发服务接口
- 典型的 **Pub/Sub** 模型

第二家公司是北京蓝城兄弟信息技术有限公司，主营项目是 **APP – Blued**，我所在的部门是基础架构部 & **IM** 服务组，我主要参与了日志系统、**CICD** 流程的搭建以及消息推送服务端重构

1. 搭建一套完善的高可用可扩展日志系统

- 架构组织: **Filebeat + Kafka + ELK + Grafana**
- **ELK** 使用 **docker-compose** 加自定义镜像和配置实现简易部署，项目的基础模板分享到 <https://github.com/wangriyu/docker-elk>
- 给 **Grafana** 添加企业微信报警设置并打包成镜像，项目分享到 <https://github.com/wangriyu/docker-grafana>

2. 给 *Golang* 项目搭建一套 *CICD* 系统

公司原本的项目以 **PHP**、**Node** 为主，只有我们部门正在转型 **Golang**，旧项目像 **Node** 使用的是 "**shipit + pm2**" 一键部署和维护，**Golang** 项目当时没有类似的工具，然后第一套方案我尝试了 "**yml 配置 + docker 容器编译 + go 脚本部署**"，虽然方案可行也能实现一键部署但配置麻烦不好用；后续方案是使用 **Jenkins** 为核心的 **CICD** 流程

3. 双向会话统计

- 数据源是每天的所有私聊消息记录，大概每天 **8000** 万条左右的文件(未压缩前 **50GB** 左右)
- 从中筛选出新会话的信息，新会话指两个陌生人第一次搭讪
- 再统计出这些新会话中对方在两小时内回复的双向会话信息
- 将上述两条按天为单位写入文件
- 此项目有一定设计难度，难点如下: 消息源不完全时间排列，机器配置有限尽量少占资源，跨天的双向会话需要额外处理，数据量不小存数据库的话查询需要优化

4. 消息推送服务端重构

消息推送是 **IM** 模块其中一个分支的下游，负责 **APP** 端上消息通知推送的后端服务， 此模块设计到多个项目，不同项目间通过 **MQ** 解耦，原先均为 **Node** 编写，后用 **Golang** 设计重构

- **PNS-Sender:** 负责推送消息中两种特殊类型的图文消息格式转换
- **PNS:** 负责整个推送模块的业务逻辑相关的项目，也是核心， 主要为解析上游七荤八素的消息、转换、过滤(对方关闭通知、屏蔽用户、黑名单等)、分发(华为、小米、谷歌、苹果，不同平台走不同推送)
- **FCM:** 谷歌推送 **SDK**，华为和小米的推送直接在 **PNS** 中发送， 而谷歌推送因为特殊原因需要放到海外服务器上，单独拎出来发送，项目按照 **FCM** 格式发送请求
- **APNP:** 苹果推送 **SDK**，苹果推送和谷歌推送占推送中的大头，请求服务单独拎出来的项目