

# 李小龙

11年工作经验 | 求职意向: Golang | 期望薪资: 35-45K | 期望城市: 成都

男 | 33岁 | 18782938024 | lixiaolongcuit@gmail.com

## 专业技能

- 精通 Golang 语言，理解语言各种数据结构的底层原理与适用场景，熟练运用高并发编程技巧等；对 kratos 框架有深入掌握和丰富实践经验，能高效构建稳定的微服务。
- 熟读openim源码，深刻理解IM相关架构与原理。
- 熟悉 Kafka 的原理与架构，善于利用其进行模块解耦，实现系统的异步通信，具备性能调优和故障排查能力。
- 深刻理解 Redis 的数据结构和使用场景，并在项目中作为缓存大量应用。
- 深刻Elasticsearch 的原理和架构，能进行海量数据查询和日志分析，擅长索引设计和查询优化。
- 深刻理解 Docker与Kubernetes，能够熟练进行微服务部署，有效管理资源、实现服务发现和负载均衡。
- 深刻理解操作系统原理与计算机网络知识
- 熟悉 GitLab CI/CD 的流程和配置，可搭建自动化的代码构建、测试和部署管道，确保项目的持续集成与持续部署，且能封装大量基础脚本。
- 深刻理解Google API设计指南，能够利用API大仓实现API的统一管理。
- 对 Web3 技术栈（如区块链原理、智能合约交互、去中心化存储等）有浓厚兴趣和一定的理论了解。具备将现有 Go 语言及分布式系统经验迁移至 Web3 后端开发的强大学习能力和潜力。
- 善于新人培养，能够制定有效的培养计划，传授技术知识和经验，帮助新人快速成长。

## 教育经历

成都信息工程大学 本科 软件工程 2010-2014

在校期间认真学习专业知识，并在校两次获得国家励志奖学金，一年校一等奖学金。并在大一参加C语言程序设计大赛获得三等奖。大三暑假开始在校外实习，参加实际项目开发。

## 工作经历

珠海新海通电子商务有限公司 Golang高级开发工程师 2024.11-至今

内容:

- 主导公司核心IM系统的服务端与SDK架构设计与研发，基于OpenIM进行深度定制与二次开发，基于napi封装node sdk。
- 深耕全链路性能调优，运用pprof、trace等工具进行持续 profiling，精准定位并解决了包括海量群聊消息广播、热点用户在线状态同步在内的多个性能瓶颈，以及客户端sdk导致应用程序崩溃问题
- 驱动技术前瞻性探索与落地，主导研究大模型技术并将其引入IM产品，设计并实现了智能问答助手与消息总结等功能，提升了用户粘性与产品竞争力

业绩:

- 通过引入批处理与优化缓存，使app卸载重装后消息同步速度提高10倍以上
- 通过race工具以及代码层错误处理规范，以及优化goroutine生命周期管理等，大大降低了sdk导致的客户端崩溃问题
- 通过全量优化到增量同步，以及网络层优化，大大提高了消息同步效率，降低了移动端电耗
- 结合自身IM业务与大模型应用技术，大大提高了用户的使用效率与用户粘性

内容：

- 负责领导团队进行基于Go语言的后端服务器开发，并采用微服务架构和k8s进行部署和管理，提升系统的可扩展性和稳定性。
- 引入Google api设计指南，并采用kratos框架实现，基于一套业务代码提供grpc与restful两种接口满足web端调用与服务间高效调用，从而实现了服务的快速开发和治理。
- 主导基于kafka的消息队列系统搭建，实现系统内各个模块的异步解耦，提升系统的并发处理能力。
- 运用Elasticsearch构建搜索和数据分析平台，优化数据检索速度，实现数亿条的URL分类的高效查询。
- 借助Redis实现缓存机制，提升系统的读写性能，缓存命中率达到95%以上。
- 建立并完善GitLab CI/CD流程，实现代码自动化构建、测试和部署，开发效率大大提升。
- 制定并推行Go工程化与代码规范，以及code review机制，提升了团队代码质量，降低代码缺陷。
- 基于Vue搭建前后端分离的管理系统项目模版。

业绩：

- 成功引入GoogleAPI设计规范，并在团队内部建立统一的API仓库，有效规范Go开发流程。
- 推广并实施GitLab CI/CD开发流程，提升团队开发效率与代码质量。
- 强化团队内部代码审查机制，确保代码提交符合高标准要求。
- 通过建立和维护基础库，显著降低了重复开发工作，提升资源利用效率。

内容：

- 实施高性能Java后端服务开发，参与核心媒资库系统，确保系统稳定运行，优化处理速度。
- 领导团队重构共享基础库，减少代码冗余，增强系统可维护性。
- 独立设计并实现了分布式爬虫系统，实现了无损音乐数据以及其元数据的爬取与分析。
- 成功解决多个生产环境下的复杂技术问题，包括但不限于内存泄漏和死锁，有效减少系统宕机事件。

业绩：

独立设计完成了分布式爬虫系统，大大加快了公司无损音乐及其元数据的爬取，并在当时获得了优秀员工奖

内容：

和别人一起创业的一段经历：

- 完成软件的设计、开发、测试、修改bug等工作,包括业务需求的沟通，功能模块详细设计，业务功能实现与单元测试，系统维护。
- 参与产品构思和架构设计。
- 撰写相关的技术文档。

业绩：

完成编务系统的设计与开发，大大提高了编辑人员的出书效率

## 项目经历

内容：

- 基于OpenIM结合公司业务进行深度定制与二次开发自己的IM系统
- 优化服务端架构与性能，以及基于公司业务进行多身份改造
- 优化客户端数据同步性能、消息加载速度等，以及基于公司业务的新功能开发

4. 调研与引入大模型能力到我们的IM系统中

业绩:

- 1. 通过引入批处理与优化缓存，使app卸载重装后消息同步速度提高10倍以上
- 2. 通过race工具以及代码层错误处理规范，以及优化goroutine生命周期管理等，大大降低了sdk导致的客户端崩溃问题
- 3. 通过全量优化到增量同步，以及网络层优化，大大提高了消息同步效率，降低了移动端电耗
- 4. 结合自身IM业务与大模型应用技术，大大提高了用户的使用效率与用户粘性

URL分类服务

项目负责人/核心开发

2017.02-2024.08

内容:

该项目主要是为上网行为管理设备以及邮件网关设备提供准确的URL分类查询服务。该项目包含多个子模块:

- 1. URL分类查询服务: 对设备端提供基于site的高效查询服务，后端采用elasticsearch作为持久化存储库提供高效的查询能力，并采用redis cluster作为高速缓存层，查询中将未命中的URL发送到kafka进行异步动态分类处理。
- 2. 数据爬取模块: 将一些站点排行网站的数据爬取清洗后入库到URL分类分层索引，增加URL分类的覆盖率
- 3. 数据整合模块: URL分类各个来源数据采用分层存储的方式存放到多个索引中，然后根据优先级生成一个整合索引供查询服务对外提供查询服务。分层索引数据变更时会发送整合消息到kafka，整合模块进行整合。
- 4. URL动态分类模块: 利用现有URL分类数据进行爬取作为训练样本，基于RNN进行动态分类模型训练，并提供一个预测服务异步从kafka消费未分类URL进行动态分类，最后将预测分值大于阈值的直接入库到分层索引，小于阈值到进行人工审核再入库。然后会基于新的数据进行模型训练与优化。

业绩:

- 1. 最开始未分类数据均通过人工审核完成，随着用户的增加，未分类数据增加越来越快，导致大量待人工分类数据堆积。通过引入基于机器学习的动态分类后大大降低了运营人员的工作量。
- 2. 通过爬虫技术，当公司拓展一个新的国外市场时，可爬取该国的一些站点排行网站来快速提高该国用户的URL分类命中率与准确率。大大提高了新市场用户的满意度
- 3. 通过设备端改造为基于站点的查询与缓存，大大提高了设备端查询速度，且降低了云端的带宽成本。

安全URL服务

项目负责人/核心开发

2017.02-2024.08

内容:

该项目主要是为上网行为管理设备以及邮件网关设备提供准确的安全URL查询服务。该项目包含多个子模块:

- 1. 安全URL查询服务: 对设备端提供基于site的高效查询服务，后端采用elasticsearch作为持久化存储库提供高效的查询能力，并采用redis cluster作为高速缓存层。
- 2. 数据整合模块: 安全URL各个来源数据采用分层存储的方式存放到多个索引中，然后根据优先级生成一个整合索引供查询服务对外提供查询服务。分层索引数据变更时会发送整合消息到kafka，整合模块进行整合。
- 3. 0day钓鱼分析模块: 当查询模块未命中安全URL，且在URL分类模块分类为未分类或代售域名或新注册网站时，会同步或异步调用0day分析模块进行钓鱼分析。0day模块通过爬取网站内容然后进行模版匹配，若命中特殊模版则表示为钓鱼网站。

业绩:

在安全URL的基础上为用户拦截了大量垃圾邮件，且通过0day钓鱼分析模块拦截了大量银行类、快递类、报税类、工资返还等类别的钓鱼邮件，得到了华为等诸多大客户的认可。

统一运营平台

项目负责人/核心开发

2017.02-2024.08

内容:

项目基于Vue搭建的前后端分离的统一运营平台，主要提供对URL分类、安全URL、垃圾邮件运营、CloudApp等模块的统一后台运营管理系统。前端使用Vue+ElementUI提供Web UI，并通过Ajax调用后端各模块等admin子模块进行数据运营管理。后端接口通过引入APISIX作为API网关来做统一认证与鉴权。APISIX会将请求token转换为内部token传递给后端各个admin子模块微服务来提供用户信息。且在鉴权模块实现了统一的操作日志记录。

业绩:

1. 引入Vue实现了前后端分离架构，并封装了前端诸多基础组件来加快开发
2. 引入APISIX作为API网关，使各个模块的API更方便测试，且认证中心统一记录用户操作日志，减少了各个模块记录的工作量，提高了团队内的开发效率。

## CI/CD改造      项目负责人

2020.05-2020.10

内容:

将公司原有的基于Jenkins的CI/CD改造为基于GitLab的CI/CD，并规范了团队内部开发流程，其中包括：

1. 开发基于Pull Request的方式来实现新功能开发与bug修复，然后提交给项目负责人进行Code Review和代码合并
2. 代码一提交即会进行lint，build，以及单元测试。
3. 通过打标签的方式来实现线上发布流程，打标签必须记录本次改动包含的内容，便于后期跟踪。打标签后会自动将Docker镜像推送到云端，然后将服务更新到K8S集群。
4. 通过提供不同操作系统，不同CPU架构的GitRunner实现了多平台的大包发布。

业绩:

1. 通过引入lint，code review流程，提高了团队的代码质量，得到了领导高度认可
2. 将原来Jenkins基于手动触发的CD流程改为基于打标签的方式，让发布版本变更记录更加清晰
3. 公司在设备端提供多平台的软件包，开始都是手动打包发布，通过引入自动打包方式，大大提高了公司出包的效率

## 个人优势

---

1. 技术领导力：拥有多年Go语言开发经验，精通grpc，kratos，gin等技术框架，在微服务、IM及时通信、k8s等领域有深入实践经验，能够为团队提供强大的技术指导与决策支持。
2. 工程化管理：建立并完善GitLab CI/CD流程，实现代码的自动化构建、测试和部署，大幅提升开发效率。同时，制定并推行 Go 工程化与代码规范，以及API设计规范，有效提升团队代码质量。
3. 项目经验丰富：熟悉IM相关技术架构，并基于openim进行二开与优化；实现过海量数据查询服务，以及多级缓存优化；熟悉大模型应用开发，以及eino开发框架；
4. 团队协作与沟通：善于组织和协调团队成员，能够有效的传达技术理念和项目需求，促进团队高效协作，共同攻克技术难题，达成项目目标。
5. 学习与创新：对新技术保持高度敏感和热情，不断学习和探索；也喜欢研究技术原理与底层实现。