

个人总结

5 年 Go 后端开发经验，长期参与商业广告、音乐推广、交易和用户增长系统研发，具备高吞吐服务、异步任务、缓存与存储优化及生产稳定性治理经验。负责内部用户分群平台整体方案并作为主要开发者，具备从 0 到 1 建设与存量系统重构经验。熟悉 Go、Redis、MySQL、Kafka，有日均亿级查询、批处理和订单链路优化经验。

教育经历

卡内基梅隆大学 | 信息技术硕士 | 2019.08–2020.12

Distinction, GPA 3.8 / 4.0

爱丁堡大学 | 计算机科学学士 | 2017.09–2019.05

一等学位, GPA 4.0 / 4.0

工作经历

腾讯音乐娱乐集团 | 后端开发工程师 | 商业广告部 | 2021.04–2026.03

· 负责用户分群平台整体方案与核心开发，并参与音乐推广、广告中台、交易商城和用户增长系统研发。

项目经历

用户分群查询平台

整体方案设计、主要开发者

业务背景：腾讯音乐内部共享用户分群平台，为广告投放、会员权益和 A/B 灰度提供在线查询；上线约半年接入 57 个服务、服务 2000 个号码包。

- 重构存储模型：将“每个用户—号码包关系一个 Key”改为“每用户—Key，紧凑存储号码包 ID 列表”，平台存储实际占用从约 200GB 降至约 30GB。
- 优化批量导入：设计 Kafka 异步导入和限流批处理链路，结合 slot 分组与 Lua 批量更新；3000 万条数据耗时约 101 秒，相比 Pipeline 提速约 7.2 倍。
- 完成在线迁移：新旧链路并行导入，查询失败时回退旧系统，验证稳定后下线旧链路，降低切换风险。
- 支撑高频查询：系统日均调用约 19 亿次、峰值约 200 万次/分钟；通过 Maglev + BigCache 缓存热点查询，命中率约 75%，缓存相关监控平均耗时 1.76ms、P99 9.90ms。

音乐推平台

核心模块开发、长期维护

业务背景：面向用户、音乐人和唱片公司的歌曲推广与粉丝互动平台，年订单量超过 100 万、年流水超过 1 亿元。

- 设计小单聚合机制：按歌曲聚合订单统一投放，再向子单分配播放量，通过明细表和定时任务支持重试与对账；累计处理约 85 万条订单，使下游投放任务量整体下降约 40%，一次大型活动中下降 91%。
- 优化订单链路性能：使用 Redis 缓存并异步化部分订单逻辑，使下单平均延迟从约 210ms 降至约 90ms，页面请求在 200ms 内完成的比例从约 65% 提升至约 94%。
- 参与钱包统一迁移：负责钱包流水同步，完成约 11.4 万个账户、46.4 万元存量余额迁移，切换期间业务保持平稳。

商业化广告中台

核心模块开发

业务背景：广告平台包含 18 个以上微服务，日均处理约 164 亿次请求；负责上报补全、下单激励和监控治理。

- 建设下单激励链路：对接 10 个以上上下游和 6 类协议，通过 Token 路由、通用接口和分层配置，将大部分新媒体接入周期从周级缩短至天级；相关工程支撑 K 歌实验组大盘收入提升 1.8%（团队与实验共同产出）。
- 参与上报补全服务建设：通过 Kafka 批量消费、Redis 数据补全、规则校验和异常标记，使服务每分钟处理约 350 万条数据。
- 治理监控高基数问题：将场景维度从 10 万以上收敛至约 600，压缩属性项规模并保留已定义故障场景覆盖，提升告警可操作性。

其他项目

- 商城与用户增长：开发聚星商城的 SPU/SKU 商品管理与库存模块；从 0 到 1 开发网赚配置服务，支撑约 3000 QPS、成功率 99.99% 以上。
- AI 辅助研发：将内部 Agent 以只读权限接入音乐推广代码库，支持调用方自助定位复杂参数校验问题；人工咨询从约 5 次/天降至约 1 次/天，重复排查时间减少约 80%。

技术栈

- 编程语言：Go（主力，约 5 年）、SQL；具备 Java、Python、Bash / Shell 使用基础。
- 后端技术：微服务、RPC / tRPC、REST、Protobuf、异步任务、并发与背压；Redis、MySQL、Kafka、MongoDB、Elasticsearch。
- 工程实践：缓存与批处理、Lua / Pipeline、数据一致性、重试补偿与对账；Prometheus、Grafana。

论文发表

[Retrofitting Security and Privacy Measures to Smart Home Devices](#)

第一作者 | IoTSMS 2019