# 蒋国华

13971153613 754635415@qq.com 职位: 搜广推-特征工程

## 教育背景

**华中师范大学** 软件工程 2021.09.01 ~ 2025.06.30

## 自我介绍

参加过一些竞赛：
2023 ICPC 程序设计竞赛 西安站 银奖
2023 CCPC 全国邀请赛 湘潭站 银奖

## 个人技能

- 【书籍】学习过C++ Primer,Effective Moder C++
- 【C++特性】智能指针、模板、SFINE、if contexpr、右值引用、lamda表达式、stl容器、bazel构建系统；
- 【多线程】了解多线程编程的基本范式，掌握多线程编程&无锁编程的spsc&brpc协程，对高吞吐量的spmc具有一定的调优经验和心得。
- 【Java语言】，理解面向对象思想，熟悉JAVA常用集合类、多线程编程、springboot；
- 深入学习了数据结构算法、操作系统、计算机网络、计算机组成和数据库原理等专业课程

## 工作经历

**虾皮科技有限公司** 搜广推-特征工程 2025.02 至今

工作时间2025-02~2025.05，2025-07~2025-09，中间几个月回学校毕业了～
【介绍】虾皮特征工程是面向整个公司内部所有SRA部门的统一平台。职责是负责拉取特征&处理特征（featrure process）。因为一些历史原因，虾皮特征工程是一个lib而不是完整的服务。
【性能】入职期间，为了提高系统吞吐，降低cpu利用率，降低延迟，我对系统做了一次性能perf、新增算子的耗时计算模块，得到了系统的某些算子存在性能瓶(比如IntersectList、TagMatchingValueList2List、FindMatchingValueList)，对热点算子采取了更合适的数据结构&算法，做出优化。成果：TP99降低20～30%，cpu利用率降低18%，为searchV2场景节约了5000core。
【重构】系统存在访存瓶颈，为了提高系统吞吐量，我们重构了整个项目，采取apache::arrow列式存储、计算和零拷贝（但是也不是完全的零拷贝）、复用内存的方式提升系统性能。比如：原先是一个item的所有特征在内存上面连续排布，现在是许多item的某一个特征在内存上面连续排布。这个重构优化的成果非常大，search&rcmd约节省了5w core。
【计算图优化】熟悉计算图的优化&AI编译器优化，并开展关于计算图&AI编译器的技术分享。比如：算子融合、常量折叠、图内并行、公共表达式消除。
【内存优化】使用alignas缓存行对齐，防止多线程伪共享现象，降低了约10%延迟。
【类型系统】系统内部面向许多模型，每一个模型的计算图不一样、算子输入输出数据类型不同、数据类型众多（int8 16 32 64 float32 64 string）✖ (scarlar array map)，我们设计了一个基于配置文件调整的运行时类型校验&计算系统，如果ALGO同学输入的算子数据类型不匹配将不能通过校验且不支持数据类型的强制转换。

**美团** 骑行事业部-iot部门 2024.04 ~ 2024.08

美团IOT团队是负责单车设备进行信息交互。
项目一：建设设备影子服务的配置触达率和触达时延指标。
1.设备影子服务配置触达时延统计：应用了Redis缓存技术来有效统计配置触达时延。触达时延定义为单车上报配置指令1到设备影子，设备影子下发修改指令，并在单车上报指令2之间的时间差。
2.设备影子服务配置触达率统计：触达率定义为单车上报指令1存在且指令2不存在的比例。采用Kafka的延迟

队列技术，确保未收到的消息能够兜底处理。
3.消息积压优化：优化前，Kafka消息实时稳定积压约360万（15*60*4000）。优化后，应用了15秒“试探消息”来检测是否收到配置应答，大幅度减少了消息积压问题。
工作收获：掌握项目开发流程、上线和维护。熟悉监控打点和打日志并意识到重要性。能够冷静处理多项紧急任务，并学会高效沟通。熟悉美团框架使用：lion、mafka、raptor、rihno、avatar、logcenter。有排查问题的基本思路。

## 项目经验

**cluster_lru_cache** 2024.08 - 2024.09 开发人员

项目描述：工业级别的单机缓存系统。是一个有锁的spmc缓存系统。
主要工作：缓存系统采用两层架构设计：1.ClusterCache (集群层): 负责分片管理和负载均衡 2.LRUCache (存储层): 负责具体的数据存储和淘汰策略
读取从集群层通过mget分片读取，写入采用asyncMSet异步写入。
1.分片集群设计 (ClusterCache)
分片数量: 64个分片 (2^6 = 64)
分片算法: 使用哈希函数进行一致性分片
2.双层LRU算法 (LRUCache)
模仿linux的lru淘汰策略，分为冷热双lru链表。
使用alignas防止多线程伪共享。
压测链接：https://docs.google.com/document/d/1G8K-8Usvyq7r39ZkqA6pmhDIRCr-vI2Lr6kQXMloFQc/edit?tab=t.0