

# 徐梓航

Tel:(+86)18782697376 ◊ Email:xuzihang829@outlook.com ◊ Github:[denjixu](#) ◊ Blog:[denji.top](#)

## 教育

西南交通大学 2023.09 - 2026.06  
硕士，计算机科学与技术

- GPA: 92/100, 优质生源一等奖学金 (1%)

西南交通大学 2019.09 - 2023.06  
本科，计算机科学与技术 (拔尖计划)

- GPA: 90.94/100, Rank: 2/20, 校级百篇优秀毕设 (1%)

## 荣誉奖项

ICPC 四川省大学生程序设计竞赛	金牌	2022.10
CCPC 中国大学生程序设计竞赛 (重庆站)	铜牌	2024.11
ICPC 国际大学生程序设计竞赛 (杭州站)	铜牌	2022.12
CCPC 中国大学生程序设计竞赛 (绵阳站)	铜牌	2022.11
中国大学生服务外包创新创业大赛全国赛区	二等奖	2021.08

## 发表论文

- Huajun He, **Zihang Xu**, Ruiyuan Li, Jie Bao, Tianrui Li, Yu Zheng. TMan: A High-Performance Trajectory Data Management System Based on Key-value Stores. *ICDE'24(CCF-A)*.

## 实习经历

微信事业群 (WXG)-腾讯 2024.7 - 2025.7

后台开发实习生，搜索技术中心

参与搜索框架在离线数据流的特性开发和维稳，完善链路保障，构建资源管理链路，编程语言 C++。

- 建设与优化离线流数据链路：参与图文/视频号离线数据流和在线召回链路的**现有业务维稳**，维护业务数据流，通过单粒度和多粒度指标接入图文视频的产出特征质检，使顺排/正排计算阶段的质量产出得到保证，99% 异常产出能够在 5min 内及时抛出并感知；参与图文划线索引的离线数据流和在线召回链路**的新增业务建设**，构建精排算子，处理原始划线信息，建立 FeatureKV 表，支撑总量 7 亿，日增 90 万 Doc 的从离线到 L4 层相关性排序的数据通路。
- 搭建在线召回与离线索引构建保障体系：构建一系列的检测兜底手段，保障在线召回架构与离线数据流的可用性/准确性/稳定性。通过全景协议和监控能力，从离线数据流到在线召回 L1/2/3，梳理链路重要指标并埋点，构建图文/视频链路监控树，协助业务/平台同学快速定位多起故障根因；基于函数告警与监控树体系，通过链路点剖面分析，根据线上影响因子归类流程指标，梯度搭建告警系统，减少 30% 的冗余告警，提升 59% 告警结单速度；结合搜索引擎召回结果特性，构建多维如 L3/L2/L3 召回模块的周期拨测与变更测试，从结构化对比和语义化对比指标发掘索引滚动和引擎变更带来的召回 Diff，抛出多次预期外变更样例，隔断上线风险。
- 构建在线资源管理链路：在线召回业务的可观测性和易用性能力缺乏，需要实现一套**可观测/可推导/规范化**的资源管理链路。以指标收集-聚合流程-数据呈现为路径进行离线/在线资源的自动化收集，完成资源观测；针对已有业务和新增业务，采用不同的资源推导逻辑，现有业务采用字段特征映射与性能压测进行推导；新增业务离线数据建模、流量仿真、动态调优进行推导，从而明确资源需求，与资源方、策略方达成资源共识。

诺司时空科技 2024.01 - 2024.07

数据库内核实习生，研发部

维护高性能开源分布式时序数据库项目 CnosDB，适配相关插件，增加新特性，编程语言 Rust。

- 实现项目 Feature；修复项目 Bug；编写测试用例增加代码覆盖率，保障高性能及高可用性。
- 开发生态适配工具：PostgreSQL FDW 外部数据包装实现数据融合；Telegraf 服务器代理实现数据获取；lot-BenchMark 测试工具实现时序数据库基准测试等。

解决方案实习生, 解决方案中心 @ 内核引擎实习生, 引擎研发部

实现业务侧前后端接口的需求 @ 参与京东城市时空数据引擎 (JUST) 的研究/设计/实验, 编程语言 Java/Scala。

- 面向业务需求, 使用 MyBatis 和 MySQL 编写映射文件和 SQL 语句实现数据持久化与对象关系映射。
- 搭建大数据集群框架, 基于 Hadoop 大数据框架包括 Spark 并行处理架构、HBase 数据库进行业务的数据分布式处理与存储。
- 协助对轨迹数据方面的时空存储索引进行改进, 将轨迹编码问题通过相似度衡量转化成 TSP/NP-Hard 多项式问题, 使用多类方法进行近似解求解, 提高轨迹查询速度, 该成果发表在 ICDE'24。

## 开源 / 项目经历

### CnosDB

2024.01 - 2024.06

由社区驱动的高性能、高压缩率、高易用性的开源分布式时序数据库。(Starred 1.7k / Contributor)

Github: <https://github.com/cnosdb/cnosdb> [Related PR](#)

- 参与文件格式 TSM 的设计/优化/实现, 封装数据层级, 保障数据落实存储/读取的可靠性和完整性。
- 对错误码跨节点传输进行序列化设计, 保障网络框架下节点错误信息能够完整传输并进行解析。
- 基于 PGRX 框架开发对接 CnosDB 的 PostgreSQL FDW (Foreign Data Wrapper), 实现回调函数支持 PostgreSQL 远程对外表进行查询/插入/批量插入/更新/删除。
- 升级适配 Telegraf, 由 Flatbuffers 导入升级为 Arrow-RecordBatch, 提高数据订阅传输效率。

### Mini LSM

2024.02 - 2024.04

独立开发

基于 Rust 开发的 LSM-Tree 存储引擎。 [Related Notes \(Starred 475+\)](#)

- 搭建 LSM-Tree 存储引擎的框架, 设计完成内存 (imm/memtable) 和磁盘 (datablock/SST) 结构, 实现 get/put/delete/scan 接口, 实现内存结构到磁盘的 flush 逻辑;
- 参考 RocksDB 的 Leveled/Universal Compaction 实现多种 Compaction 算法; 设计 Manifest 与 WAL 实现 LSM-Tree 的状态持久化/恢复完成一个完整的存储引擎设计;
- 实现 MVCC, 支持事务在基于乐观锁实现的快照隔离下对 LSM 存储引擎进行 get/scan/put 等 API 调用, 同时实现基于 Watermark 和 Compaction Filter 的 GC 释放空间;

### Raft KV

2023.08 - 2023.12

独立开发

基于 GO 开发的分布式系统 (MIT 6.824)。

- 实现 MapReduce 架构, Workers 与 Master 通过 RPC 实现 Map/Reduce Task 的分发, 中间键值的存储及处理, 优化 Straggler 带来的 Tail Latency;
- 实现 Raft 架构, 通过实现 RequestVote/AppendEntries/InstallSnapshot RPC 支持 Leader 选举, 日志复制, 日志压缩, Snapshot 等功能;
- 编写基于 Raft 架构的分布式 K-V 数据库, 通过在 Clerk 客户端中维护唯一请求 ID 同时在 KVServer 端记录上次请求 ID 实现 RPC 语义中的 Exactly Once 避免请求多次执行;

## 技能

- 熟练使用 C/C++ 语言, 熟悉 STL 容器, 理解多态、RAII 思想、智能指针、移动语义等特性; 了解内存序模型与缓存一致性协议; 了解 Rust, Go, Java 和 Python 语言, 能用以辅助工作。
- 了解 Elasticsearch 搜索引擎中间件, Kafka 消息队列架构与原理;
- 了解经典召回模型/索引结构/向量检索/排序算法;
- 熟悉 Raft 分布式一致性协议和常见的优化方式; 了解 Paxos 共识算法和 ZAB 协议。
- 英语水平: CET-6, 能流畅阅读英文文献。
- 热爱并有强烈的驱动力学习搜索引擎/存储引擎/分布式系统/数据库相关领域知识, 个人有写技术博客的习惯。