

# 矩木

📞 1998-3-30 · 📧 wwc7033@gmail.com · 📞 (+86) 123-4567-8910 · 🌐 CanftIn

## 工作经历

### 上海橙鹏汽车科技有限公司 (小鹏汽车), 上海

2024.05 – 至今

Software Engineer 负责小鹏汽车智能编译部 AI 编译器开发工作

- 负责 NPU 编译器中后端计算图与多核性能优化 | ft.Owner 2024.05 – 至今
  - 实现后端 MLIR 多层 pass 中的功能, 包括根据硬件数据布局进行数据格式的推断检查, 并进行硬件相关算子的适配与优化, 验证在芯片上编译通过 20 余种大模型。
  - 实现 L1 cache 的地址分配、对齐与赋值, 实现 L1 和 L2 的地址同步与统筹管理, L1 实现 pingpong 分配, 三级缓存间实现自动 Partition 和自动 Schedule。
  - 实现多 batch 情况下的硬件相关算子适配, 泛化并实现多种情况下的单算子拆分与多算子融合 pattern, 简化模型大小, 提升计算性能。
- 负责 triton-lang 在编译器前端的功能接入 | ft.Owner 2024.06 – 至今
  - 给 triton 接入 mlir 后端 pass, 接入新一代 XP6 芯片指令集, 利用 triton py-like 语法的优势下, 实现后端 ptx、risc-v 宏指令异构, 提升编译器易用性。

### 图灵量子科技有限公司, 北京

2022.09 – 2024.03

编译器开发工程师 负责光计算芯片上编译器工具链开发工作

- 负责构建基于 LLVM 的支持 C++ 与 Python 语法的 DSL 编译器 | Owner 2022.10 – 2024.03
  - 基于 ChaiScript 设计并预言性开发支持量子线路的 qscript 语言, 以 C++ 语言库的形式体现, 构建词法、语法解析、部分前端优化以及 bootstrap 与构建标准库的执行过程。将量子线路工具包 (qiskit-aer) 以模块接入标准库, 可构建基础量子模拟程序, 同时外部库接入 taskflow, 依赖于 cudaflow 的条件下初步实现 cpu/gpu/qpu 的异构计算能力。期间同步学习 lua 语言设计及源码分析, 以及实现简易任务调度框架 CanftIn/meepo。
  - 调研 LLVM 后端模块以及 Clang 源码中前端组件功能, 挖掘 Clang 对 C++ 及 Cuda 语法的编译执行过程; 主导全流程跟踪 [Google/carbon-lang](#) 从初期迭代至今的提交, 深入分析语言开发和设计过程, 包括 explorer 和 toolchain, 并贡献 pr, 基于 toolchain 重写版本: cocktail-lang, 构建 Lex/Parse/Diag/Check/SemIR/CodeGen 等模块, 掌握工业级传统语言编译器前端的设计流程和构建能力; 同时调研 Swift 早期实现, 基于其在 LLVM 之上验证性进行 radium-lang 手写编译器的开发, 探索复杂语言项目中的工程实现。
  - 主导全流程分析 codon(py-like) 工具链, 重写版本: pud-lang, 针对性研究了前端实现中 PEG Parser、Simplify、TypeCheck、Translate 等功能及 AST 层级优化, 中端定义 IR 层中数据流分析以及 Transform 优化, 后端接入 LLVM IR 及 Cuda Runtime, 同时调研内置的 jit 编译功能与 LLVM JIT 功能。验证编译器在 llama2 模型下的执行效率提升。期间同步学习分析 [pytorch c10/aten](#) 模块代码, 理解基础张量结构设计和计算。
  - 调研 triton、taichi、tvm、halide 实现, 分析内部 IR 层定义与设计。调研 [morganstanley/hobbes](#), 分析其中 parser 以及类型推理模块。调研 [jank-lang](#) 实现。学习 Graph-IR 及 SSA-IR。
  - 调研 NVIDIA/cuda-quantum, 探索基于 mlir 的 DSL 编译器构建技术。
- 负责光计算芯片驱动侧接口的开发 | ft.Owner 2022.10 – 2023.10
  - 主导开发网络调用式光计算芯片的向量乘接口, 制定收发数据包规格, 提供 C++ 和 Python 版本的上层矩阵调用接口, 绑定驱动, 支持向量拆分到 1x5 大小进行对应元素乘。实现初步的芯片上计算跑通。
  - 主导 pcie 光计算芯片的 4x4 和 16x16 矩阵乘接口, 配合构建 fpga 驱动侧数据包规格, 提供封包解包功能以及单包多包验收用例。在芯片上层实现基础 gemm 通用矩阵计算库。完成手写体 resnet 模型推理的功能性验证。
- 负责计算层资源调度和分配服务的开发 | Owner 2022.09 – 2023.03
  - 实现计算层资源服务的设计、开发与部署。支持分布式容器化分配资源, 支持 slurm 任务调度计算, 实现对底层 cpu/gpu/mem 计算资源做细粒度分配, 提供简易的任务调度策略, 期间学习过 [nicolargo/glances](#) 中监控部分功能以及 [sogou/workflow](#) 和 [apache/thrift](#) 协议代码。
  - 实现计算层实际资源和预约资源的监控客户端开发。同时为团队搭建容器镜像 cicd 流程, 提升开发效率。

### 腾讯北京科技有限公司, 北京

2020.07 – 2022.08

后台开发工程师 至腾讯北京 CSIG 地图平台部负责后台开发工作

- 参与高精交换数据平台建设, 负责全国高精数据编译工具开发 | ft.Owner 2021.07 – 2022.08
  - 开发全国高精地图生产数据编译工具中如地标和道路等多个要素的编译业务逻辑, 完成从生产数据到产品数据的转换, 解决人工难以切分几何、难以建立要素关联关系等痛点。
  - 独立负责并解决编译过程中要素 ID 唯一性与版本继承问题, 并根据空间索引优化查找方式, 提高编译效率。编译工具支持全量 Sqlite 数据导出与线上 MySQL 入库, 支持分布式编译全国数据。
  - 合作开发高精数据的检查工具, 规范化输出日志与问题数据以支持产品数据检查。
  - 独立开发数据清洗工具、多个数据统计工具与数据差分工具, 支持数据结果在腾讯 BI 报表平台可视化展示。
- 参与生产平台数据服务与轨迹后处理平台中接口功能开发 | ft.Owner 2021.02 – 2021.07
  - 开发数据上传下载、历史查询等功能, 实现数据断点续传功能, 期间学习并加深对 C++17 的运用以及后端数

据组织的理解。

- 负责设计和实现自动化算法流转服务平台及实现对比工具的编码与接入 | Owner 2020.07 – 2021.02
  - 设计并开发消息队列模式自动化流转服务, 定义 RESTful API, 完成任务状态心跳进程与监控进程的开发, 参与部分前端页面开发, 完成前后端流水线搭建与部署。平台支持多个道路要素识别算法构建 DAG 自动化运行, 支持多个流程图并行, 图子环节人工增删改查友好。
  - 实现了对比工具产出准召率结果以评测算法, 支持图形化对比差异, 支持前端进行算法比较的展示, 对比工具可接入流程图子环节进行自动计算。

## 🎓 教育经历

武汉轻工大学, 通信工程, 本科

2015.09 – 2019.06

专业卓越班, 学校电子设计比赛第一名, 17 年全年阅读量全校第二 (约 420 本, 主要自修计算机相关书籍)。在校有 C++/Java 项目的合作开发经验以及社群沟通经验。

## 🔗 开源项目

CanftIn/cocktail-lang | 基于 Carbon 的下一代 C++ interop 语言



- 参考 carbon/Swift 覆写基于 LLVM 的上层 Tokenizer/Parser/Diagnosis/Semantic/Driver 实现。
- 覆写完成 CppRefactor 功能实现, 支持 Cocktail 语言和 cpp 互调用, 期间分析 [hsutter/cppfront](#) 实现
- 基于 Flex/Bison 实现实验性解释器, 支持类型检查及栈帧表达式。

CanftIn/jmlang | py-like 数据并行计算语言



- 提供基础的 IR 定义和计算操作。
- 提供 C++ 语言层面的函数、变量和表达式定义接口。
- 提供基于 LLVM OrcJit 的 jit 编译模块。支持 codegen 到 cpu/gpu(WIP)。

CanftIn/RegGen | 支持 LALR(1) 语法规则的语法生成器



- 实现简易正则解析到词法自动机构建。
- 实现 LALR(1) 语法自动机, 支持手写语法规则解析, 支持 codegen 到 C++ 代码。

jm-research/graph-ir | 图中间表示的实现, UCI 大作业 PL241



- 基于 boost graph 实现 graph 和 node 定义, 提供建议图语法的 parse 功能, 并测试功能, 完成 graph ir 上的窥孔优化 (图约简)、图公共子表达式消除、mem2reg 优化等。

jm-research/omilang | LLVM 之上 clojure 前端实现



- 提供 lisp 语法层的 lex & parser 功能。
- 提供 lisp runtime 层面的对象类型封装, 接入不可变数据结构, 提供装箱操作, 提供 eval 执行接口。

## ⚙️ 专业技能

- 编程语言: 尤为熟悉 C++/Python, 使用过 Go/Rust/Typescript/Java/SQL/Haskell/Shell。
- 开发工具: 日常在 MacOS/Ubuntu 下使用 VSCode/Vim/Emacs, 熟练使用 Git/CMake/Bazel/xmake 等开发工具, 有使用 GitHub 协作开发经验, 有独立搭建 ArchLinux+i3wm 开发环境并长期使用经历, 能适应任何编辑器/操作系统。
- 关键技能:
  - 基础: 熟悉常用数据结构、算法和设计模式, 具备良好的编程风格。
  - 编译: 熟悉 MLIR 与 LLVM 编译框架, 熟悉 MLIR 与 LLVM pass 过程编写, 了解 JIT/AOT 编译, 对 clang 前端有代码分析及改动经验, 熟悉编译器前端, 熟悉 lex/bison/antlr4 生成器的语法及使用, 接触过 tvn 的使用, 了解 graphir 层面的优化及图调度。
  - 数据库: 熟悉 MySQL/Redis/Sqlite 等常用数据库使用, 了解 MySQL 事务和 InnoDB 存储引擎, 完整分析过 leveldb 及 duckdb 项目代码 (理解 lsm 树存储结构, 理解 OLAP 数据库分析及优化部分, 接触过 SQL 解析), 分析过部分 redis 代码 (kv 数据结构、网络接口)。
  - 云原生: 熟悉 docker 的命令及构建, 熟悉 ci/cd 编译流水线的构建, 有 k8s 的使用经验。
  - 业务: 基于 TangSengDaoDaoServer 构建 im 服务器 (golang), 了解 goframe 框架, 了解复杂业务的建表逻辑, 使用过 netlify 部署网站。

## 📖 其他

- 博客: [canftin.github.io](#), 编写《基于 Carbon: 一步一步构建现代语言编译器》系列开源文章, 文章已公开至 [carbon-blog](#)。
- 社区参与/实践: 活跃于 Github, 日常习惯跟踪各种类型项目, 乐于参与开源社区讨论, 参与优化多个 C++ 开源文档, 作为社区成员曾活跃于 Emacs China 社区与 ArchLinux 社区, 向 spacemacs, llvm-project, carbon-lang, chaiscript, mastodon 等项目提交过代码并通过审查与合并。经常浏览 StackOverflow、Hacker News 等技术网站。
- 技术会议及获奖: 参加 2016 年 C++ 及系统软件技术大会, 参加 2023 北京智源大会, 参加 2023 TVM 北京, 获腾讯云开发通用认证证书。
- 外语: 英语 - 良好 (流畅阅读英文书籍及文档等)。
- 业余: 喜欢交朋友, 打羽毛球六年, 拳击半年, 日常长跑, 抗压能力较强, 有自我驱动学习能力。