

黎泽仁

✉ kould2333@gmail.com · ☎ (+86) 180-1190-6217 · 🌐 <https://github.com/KKould>

👤 工作经历

- InfiniFlow**2023.10-2024.2
- 数据库内核工程师
 - 负责算子实现、优化器以及日常 Bug 维护
- 开源中国 Gitee**2022.12-2023.7
- 后端开发工程师
 - 维护 Gitee 企业版功能迭代以及问题修复
- 绿狐科技**2022.2-2022.11
- 后端开发工程师
 - 负责绿狐企业内部培训系统内的功能开发

👤 项目经历

- Infinity**2023.10-2024.4
- CPP* 全职开发
- The AI-native database built for LLM applications, providing incredibly fast full-text and vector search repo:
<https://github.com/infiniflow/infinity>
- **LateMaterialization**: 延迟加载所需的列数据, 计划存在多数据源算子时导致重复加载列数据而造成多余的物化开销, 降低单次计划执行的内存占用
 - **MergeLimit Executor**: 将 Limit build 为多个子 Fragment, 使其能够在分布式环境下, 让多个节点并行进行 Limit 计算
 - **Sort Executor**: 在算子每次在被调度器执行 Sort 计算时将此次计算输入数据进行排序, 并物化, 在最后一次计算时进行归并排序
 - **WAL 加载优化**: 使用多层迭代器延迟对 WAL 以 Block 为单位进行读取并反序列化而避免存储大数据量时, 服务重启导致恢复运行状态需要占用大量内存资源
 - **Query Profiler**: 基于 DuckDB 的 Profiler 基础上对执行进行更细粒度的记录: 计划执行时间、Binder 到 Executor 的各个阶段计算时间、算子之间计算时间、算子每次被计算的时间, 并实现非持久化的 Profile 记录查询, 提供 Show Profile 使用户可以查看对应计划的具体执行信息
- FnckSQL**2023.6-至今
- Rust* 作者
- SQL as a Function for Rust repo: <https://github.com/KipData/FnckSQL>
- 覆盖 SQL 2016 绝大部分 Case, 例如: In Subquery、Unique、Group By、Inner/Left/Right/Full/Cross (Outer/Semi/Anti) Join
 - RBO + CBO 实现高效的计划查询以及索引选择, 提供多种 Index: PrimaryKey/Unique/Normal/Composite
 - 参考 DuckDB 结合 Memcomparable 实现 Radix Sort, 在大数据量下提供高效的排序计算性能
 - 主体 Volcano 执行器 + 实验性质的基于 Luajit 的 Codegen 执行器实现, 允许用户根据计算场景进行选择
 - 支持 PGWire 协议, 允许使用 psql 等客户端进行远程连接
 - 支持 UDF 并提供 function 宏使用户能够针对具体场景而设计自己的计算函数在 SQL 中
 - 高写入性能, Benchmark 与 SQLite 进行对比测试, 在 20 万行数据量下, SQLite 总时间为 5 分钟, 而 FnckSQL 总时间为 5 秒

KipDB

2022.7-至今

Rust 作者

Lightweight, asynchronous based on LSM Leveled Compaction KV database repo: <https://github.com/KipData/kipdb>

- Leveled Compaction 压缩策略, 参考 LevelDB 实现, 提供更高的查询速度
- MVCC 事务支持, 参考 RocksDB 结合 SnapShot 与 WriteBatch 以及乐观冲突检测实现
- 支持多种 Compaction: Sized/Seek/Manual, 使数据库的 Compaction 触发更加灵活
- 以 Kiss(Keep It Simple, Stupid) 作为开发理念, 设计以简单而高效为主
- 基于 ProtoBuf 提供 RPC 服务, 允许用户使用远程调用

🐼 文章博客

InfiniFlow

2023.10-2024.2

- SQL Sort 算子实现解析以及 DuckDB Sort 模仿实现: <https://zhuanlan.zhihu.com/p/673945166>
- SQL Filter 的表达式代换与常量关系提取: <https://zhuanlan.zhihu.com/p/676332648>
- SQL Codegen 全阶段代码生成执行器实现: <https://zhuanlan.zhihu.com/p/677154366>
- Cost-Based Optimization 之行数预估: <https://zhuanlan.zhihu.com/p/680465610>

⚙️ 开源贡献

- GreptimeDB: 'Create Table .. Like': <https://github.com/GreptimeTeam/greptimedb/pull/3372>
- GreptimeDB: 'is_null' function: <https://github.com/GreptimeTeam/greptimedb/pull/3360>
- skip-list: add 'low_bound'/'upper_bound' api: <https://github.com/JP-Ellis/rust-skiplist/pull/19>

🎓 教育背景

广东机电职业技术学院, 广东, 广州

2019 – 2022

专科 软件技术

🧑 其他

- 开源爱好者, 维护有多个开源项目同时为 GrepTimeDB、skip-list 等热门仓库做出贡献
- 数据库爱好者
- 语言: 日语 - N3