

刘东坡

✉ rustin.liu@gmail.com · ☎ +(86) 189-0839-4732 · 🌐 hi-rustin · 🔗 hi-rustin.rs

🎓 教育经历

软件工程本科

2015/09 – 2019/06

- 重庆理工大学

⚙️ 技能

- **编程语言**: 熟悉 Go/Rust/JavaScript 泛语言 (编程不受特定语言限制)
- **开发工具**: 熟悉 Linux, 有 Git、GitHub、Docker、Jenkins、Github Actions 和 Jira 等团队协作工具的使用经验
- **语言**: 英语 (可以流利的听说读写), 中文 (母语)

👤 工作经历

PingCAP Inc. - 数据库 - 数据库内核研发 (Golang/Rust)

2023/08 – 至今

从数据平台部门转组到计算内核部门, 开始专注于分布式数据库 TiDB 优化器的研发。

职责:

- 设计、实现和维护分布式数据库 TiDB 的优化器。

产出:

- 添加 LOCK STATS TABLE 功能, 方便用户锁定统计信息的更新, 生成稳定的执行计划。
- 删除了大量弃用功能代码, 极大的改善了代码可维护性。

PingCAP Inc. - 数据库 - 数据库工具研发 (Golang/Rust)

2021/07 – 2023/07

从社区部门转组到数据平台部门, 开始专注于数据同步工具 TiCDC 的研发。

职责:

- 设计、实现和维护核心模块, 专注于 Sink 和 Sorter, 管理项目日常维护, 包括 issue 追踪、代码审阅和 on-call 客户支持。

产出:

- 将推送模式修改为了拉取模式优化了内存管理和数据同步流程, 极大的改善了 OOM 问题和同步延迟。
- 将 Sink 模块从同步模式修改为异步模式, 优化代码性能, 将增量场景下吞吐提升 30%。
- 为 Kafka Sink 支持了多 Topic 和 OAuth 授权功能, 极大的降低了多 Topic 的任务运维成本。
- 将有损 DDL 从全表扫描优化为不同步任何数据, 极大的降低了有损 DDL 对系统的影响。
- 重构了 TiCDC 的 CLI 所有代码, 标准化了命令添加和编写方式, 极大提升了可维护性。

PingCAP Inc. - 数据库 - 前后端开发工程师 (Golang/JS)

2020/08 – 2021/07

加入了社区部门, 以 Kubernetes 社区 Prow 项目为基础开发 TiChi 来支撑社区协作。

职责:

- 使用 Prow 在 GKE 上设计和实现一个协作机器人项目, 优化开源贡献者的社区协作流程。

产出:

- 按时完成了项目的设计和开发, TiChi 已经成为了公司 CI/CD 的基础设施。
- 为 Prow 修复 bug 和添加功能, 提交了 39 个 PR。
- 为 TiChi 编写了良好的文档, 明确了协作流程, 为开源贡献者提供了更好的协作体验。

晨星资讯 (Morningstar, Inc.) - 金融服务 - 前后端开发工程师 (Java/JS) 2019/06 – 2020/07

毕业后第一份工作, 负责编写和维护数据展示组件, 在公司决定开始关闭中国研发中心后离职。

职责:

- 使用 Java 和 Vue 开发和优化前后端组件, 提升工程效率和前端基础设施。

产出:

- 多个独立开发的组件已经上线使用, 有较好的稳定性和性能。
- 实现了前端的 tracking 系统, 使用 HeapIO 追踪用户信息, 并预留可扩展接口, 可无缝切换至 Google Analytics。
- 为组件和文档发布编写了自动化 Jenkins 脚本, 将发布流程从手动 4 小时优化到全自动 30 分钟。

Cargo - 活跃贡献者

260+ commits

Rust 包管理器, *Cargo* 下载 *Rust* 包的依赖, 编译包, 制作可分发的包。

- 改善了大量 *Cargo* 命令和 *Rust* 编译器的错误提示, 让错误更有效更清晰。
- 为 *Cargo* 添加了 `-crate-type` 功能和相关文档。
- 为 *cargo new* 和 *cargo init* 命令支持了 `[workspace.package]` 字段自动继承。
- 为 *Cargo* 构建脚本添加新的语法, 为未来扩展构建脚本提供基础。
- 添加了 `'target.'cfg(..)'linker'` 配置项, 允许用户通过 `'cfg(..)'` 条件来指定链接器。
- 添加了 `OR_PATTERNS_BACK_COMPAT` lint 来帮助用户迁移至新的 `orpatterns` 语法。
- 使 `authors` 字段在 *Cargo* 和 *rustdoc* 中可选。

cargo-information - 作者

270+ commits

cargo-info 是一个提供 *Rust* 包详细信息工具。

- 兼容所有与 *Cargo* 兼容的注册中心。
- 获取并显示基本的包信息 (名称、版本、所有者等)。
- 显示包依赖及其版本。
- 提供包的 `features` 信息。

crates.io - 维护者

60+ commits

Rust 包管理中心。

- 帮助实现 RFC 3052, 让 *crates.io* 开始接受无作者信息的包。
- 支持下划线前缀的重命名包在 *crates.io* 的发布。
- 对齐了 *crates.io* 和 *Cargo* 对 `feature` 和 `crate` 的名称校验规则。
- 修复不稳定测试, 完善文档, 提升代码可维护性。

Rustup - 前维护者

110+ commits

Rustup 是 *Rust* 的安装器。*Rustup* 从官方发布渠道安装 *Rust* 编程语言, 使用户可以轻松地在稳定版、测试版和夜版编译器之间切换并保持更新。

- 为 *Rustup* 支持了 `self-update` 的不同模式来提供更好的使用体验。
- 改善 *Rustup* 测试, 负责发布 *Rustup*。
- 为 *Rustup* 添加了 UI 测试, 顺利升级了 `clap` 依赖。
- 将 `term` 依赖替换为 `termcolor`, 修复了 `MSYS2` 终端下颜色显示的问题。
- 负责项目日常维护, 包括代码审查、依赖升级、安全漏洞修复等。

ant-design - 前维护者

24+ commits

企业级 UI 设计语言和 *React* UI 库。

- 重构了 `Input` 组件, 添加清理按钮。
- 设计并添加了 `Skeleton` 组件。
- 编写了贡献指南, 引导其他人参与社区贡献。