

RUSTIN LIU

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

🎓 EDUCATION

Software Engineering Bachelor's Degree

Sep 2015 – June 2019

- Chongqing University of Technology

⚙️ SKILLS

- **Programming Languages:** Experienced in Go/Rust/JavaScript, **multilingual** (can adapt to any language)
- **Development Tools:** familiar with Linux-based programming, have experience with tools like Git, GitHub, Docker, Jenkins, Github Actions and Jira, etc.
- **Languages:** **English** (Can read, write and speak fluently), **Chinese** (Native speaker)

👥 WORK EXPERIENCE

PingCAP Inc. - Database - Database R&D

July 2021 – Present

Transferred from the community department to the data platform department and started to focus on the development of data synchronization tools TiCDC.

Responsibilities:

- Designed, implemented and maintained core modules, mainly focusing on Sink and Sorter modules.
- Provided day-to-day maintenance of the project: Tracked issues and resolved them. Reviewed code and was on-call to resolve customer issues.
- Optimized project code structure and CI/CD script. Tracked flaky test to improve engineering efficiency.

Accomplishments:

- Changed push mode to pull mode to optimize memory management and data synchronization process, greatly improving OOM issues and synchronization lags.
- Modified the Sink module from synchronous to asynchronous mode to optimize code performance and improve throughput by 30% in incremental scenarios.
- Added multi-topic and OAuth authorization support for Kafka Sink, greatly reducing the operation and maintenance costs of multi-topic tasks.
- Added the ability to optimize the Lossy DDL from full table scan to not synchronizing any data, greatly reducing the impact of DDL on the system.
- Refactored all of TiCDC's CLI code to standardize the way commands are added and written, which greatly improved maintainability.

PingCAP Inc. - Database - Full-Stack Engineer

Aug 2020 – July 2021

Joined the community department to develop TiChi to support community collaboration based on the Kubernetes Community Prow project.

Responsibilities:

- Designed and built the entire project. Built collaborative robots on GKE using Prow.
- Sorted out and optimized the community collaboration process to provide a better collaboration experience for open source contributors.

Accomplishments:

- Completed TiChi, which has become the company's CI/CD infrastructure, on time.
- Fixed bugs and added features for Prow, submitted 39 PRs.
- Added multiple language documents for collaboration process and TiChi.

Morningstar, Inc. - Financial Services - Full-Stack Engineer

June 2019 – July 2020

First job after graduation, responsible for writing and maintaining data presentation components, left after the company decided to start closing its China R&D center.

Responsibilities:

- Developed front and back-end of the data presentation component in Java and Vue.
- Optimized the release scripts for components and documents to improve engineering efficiency.
- Maintained and developed front-end infrastructure, driving front-end engineering.

Accomplishments:

- Independently developed components that are already online, with good stability and performance.
- Implemented a front-end tracking system that uses HeapIO to track user information, with an extensible interface to seamlessly switch to Google Analytics.
- Wrote automated Jenkins scripts for component and document releases, optimizing the release process from a manual 4-hour process to a fully automated 30-minute process.

Cargo & Rust - Active Contributor

219+ commits

The Rust package manager. Cargo downloads your Rust package's dependencies, compiles your packages, makes distributable packages.

- Improved various error prompts for Cargo commands and the Rust compiler to make errors to effective and clear.
- Added the `-crate-type` function and added related documentation for Cargo.
- Added support for automatic inheritance of the `[workspace.package]` fields for the `cargo new/init` commands.
- Added new syntax for Cargo build scripts to provide a foundation for future extensions to build scripts.
- Added `OR_PATTERNS_BACK_COMPAT` lint to help users migrate to the new `orpatterns` syntax.
- Made `authors` field optional in Cargo, `crates.io` and `rustdoc`.

pyroscope - Active Contributor

37+ commits

Continuous Profiling Platform. Debug performance issues down to a single line of code.

- Refactored all command line flags and added flags descriptions and tests.
- Added a management page to show the status of each component.
- Refactored all APIs to make them easier to write and test.
- Added linter and formatter to the project.

Rustup - Previous Maintainer

110+ commits

Rustup is an installer for Rust. Rustup installs The Rust Programming Language from the official release channels, enabling you to easily switch between stable, beta, and nightly compilers and keep them updated.

- Different modes of self-update were supported for Rustup to provide a better experience.
- Improved Rustup testing and was responsible for releasing Rustup.
- Added UI tests for Rustup and successfully upgraded the `clap` dependency.
- Replaced `term` dependency with `termcolor` to fix the color display problem in MSYS2 terminal.
- Responsible for the daily maintenance of the project, including code review, security vulnerability fixing, etc.

ant-design - Previous Maintainer

24+ commits

An enterprise-class UI design language and React UI library.

- Refactored `Input` component and added `cleanup` button.
- Designed and added `Skeleton` component.
- Wrote contribution guidelines to guide others to participate in community contributions.