

PR #39457 完整报告

sgl-project/sglang

[sgl-router] Prepare dynamo-render dependencies

合并时间: 2026-09-15 10:08

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/39457>

执行摘要

- 一句话: 升 Rust 1.92 并锁定构建依赖, 为 Dynamo 渲染铺路
- 推荐动作: 值得快速阅读, 可作为“为功能迁移做依赖与供应链准备”的范式: 工具链升级留痕、锁文件提交、CI 全 `--locked`、`deny.toml` 随依赖图演进动态收紧。重点关注 `Cargo.toml` 中依赖来源从 git SHA 到 registry 精确版本的迁移理由, 以及 `deny.toml` 中 `unmaintained` 检查范围的权衡。因无运行时逻辑变更, 不建议深度精读。

功能与动机

PR body 明确说明这是为 Dynamo 渲染准备路由器构建 (Prepare the router build for Dynamo rendering): `dynamo-tokenizers` 1.8.1 要求 Rust 1.92; `Dynamo crates` 改为 registry 发布版本并提交 `Cargo.lock`, CI 使用 `--locked`; 当前 `chat renderer` 保留到 #38983 后再迁移。第一个 commit 指出工具链升级是硬性依赖 (`floor_char_boundary` 稳定化), 第三个 commit 记录 `cargo deny` 因 RUSTSEC-2026-0285 失败而必须升级 `rustls` 至 0.23.45, 说明供应链审计是本 PR 的验收门槛。

实现拆解

1. 升级工具链至 Rust 1.92: `experimental/sgl-router/rust-toolchain.toml` 的 `channel` 从 1.90 改为 1.92, 理由是 `dynamo-tokenizers` 1.8.1 依赖标准库 `floor_char_boundary` (1.92 稳定化)。同步更新 `tests/e2e/k8s_integration/Dockerfile.router` 与 `docker/sgl-router.Dockerfile` 的基础镜像为 `rust:1.92-bookworm` 并固定 `RUSTUP_TOOLCHAIN=1.92.0`, 避免构建期 `rustup` 频道同步。
2. 切换 Dynamo 依赖来源并预置渲染器依赖: `Cargo.toml` 删除 `dynamo-protocols`、`dynamo-parsers` 两个 git SHA 固定依赖, `dynamo-tokenizers` 改为 `crates.io` 精确版本 `=1.8.1`, 新增 `dynamo-renderer = "=5.1.2"` (当前代码尚无引用, 属于“先落地依赖、后接入逻辑”的前置步骤)。`minijinja` 从 2 提升到 2.24, 并把 `chat-template` 渲染说明注释移回依赖声明处, 明确该组合是过渡方案。
3. 提交锁文件并全链路 `--locked`: 新增 `experimental/sgl-router/Cargo.lock` (4687 行), 根 `.gitignore` 对该路径开白 (此前仓库级规则忽略了所有 `Cargo.lock` 除 `rust/` 下两个)。CI 中 `cargo check/clippy/build/test` 全部加 `--locked`, 并删除原 `DYNAMO_TOKENIZERS_EXPECTED_REV` 的 SHA 校验步骤——`git pin` 已不存在, 锁文件成为唯一事实源。

4. 收紧供应链策略: deny.toml 中 unknown-git 由 allow 改为 deny (git 依赖已清零); unmaintained 由 none 改为 workspace (只检查直接依赖, 传递依赖等上游移除); 删除 malachite-* 的 LGPL 例外与 NCSA 许可白名单 (依赖图已不含这些 crate); rustls 升级至 0.23.45 修复 RUSTSEC-2026-0285, 使 cargo deny 全绿。
5. 改造镜像构建: docker/sgl-router.Dockerfile 的 chef 阶段不再 cargo generate-lockfile, 改为直接 COPY 提交的 Cargo.lock; cargo chef cook 与最终 cargo build 均加 --locked, 保证镜像依赖图与 CI 审计结果严格一致; E2E 测试镜像同步加 --locked。
6. 验证配套: 格式化、Clippy (-D warnings, 1.92 下干净)、836 个 workspace 测试、cargo deny 许可 / 禁用 / 来源检查全部通过; PR Test 主任务为绿色, Extra 任务为 X 但无详情。

关键文件:

- experimental/sgl-router/Cargo.toml (模块 依赖配置; 类别 config; 类型 configuration): 依赖策略核心变更: dynamo-* 从 git SHA 固定切换为 crates.io 精确版本, 新增 dynamo-renderer 预置依赖, minijinja 提升到 2.24, 是整个迁移的入口。
- experimental/sgl-router/deny.toml (模块 供应链策略; 类别 config; 类型 configuration): 供应链策略收紧的核心文件: 禁止 git 源、unmaintained 改为仅检查直接依赖、删除不再在依赖图中的许可例外, 并记录每条策略的权衡理由。
- .github/workflows/pr-test-sgl-router.yml (模块 CI 流程; 类别 infra; 类型 infrastructure): CI 全链路启用 --locked, 删除 dynamo SHA 校验步骤, 锁文件成为唯一构建事实源, 防止漂移。
- docker/sgl-router.Dockerfile (模块 镜像构建; 类别 infra; 类型 infrastructure): 镜像构建从容器内生成锁文件改为直接使用提交的 Cargo.lock, 保证产物与 CI 审计的依赖图一致。
- experimental/sgl-router/Cargo.lock (模块 锁文件; 类别 other; 类型 dependency-lock): 新增提交的锁文件 (4687 行), 是可复现构建与依赖审计的基础, 也是 git 源依赖切换后的新事实源。
- experimental/sgl-router/tests/e2e/k8s_integration/Dockerfile.router (模块 测试镜像; 类别 test; 类型 infrastructure): E2E 测试镜像同步工具链版本并启用 --locked, 保证测试环境与 CI 构建环境一致。
- experimental/sgl-router/rust-toolchain.toml (模块 工具链; 类别 config; 类型 configuration): 工具链升级的声明点: 1.90 → 1.92, 是 dynamo-tokenizers 1.8.1 的硬性要求。
- .gitignore (模块 忽略规则; 类别 other; 类型 configuration): 仓库级 Cargo.lock 忽略规则为 sgl-router 开白, 是提交锁文件的前提。

关键符号: 未识别

关键源码片段

experimental/sgl-router/Cargo.toml

依赖策略核心变更: dynamo-* 从 git SHA 固定切换为 crates.io 精确版本, 新增 dynamo-renderer 预置依赖, minijinja 提升到 2.24, 是整个迁移的入口。

```

[workspace]
resolver = "2"
members = [".", "sgl-kv-indexer"]

[package]
name = "sgl-router"
version = "0.1.0"
edition = "2021"
license = "Apache-2.0"
publish = false # 仅二进制产物，不发布到 crates.io

[dependencies]
# 此前：dynamo-* 三件套按 git SHA 固定（rev = "1efdd4dc..."），CI 另需人工
# 校验 SHA 一致性，deny.toml 也得对 git 源开白。
# 现在：改为 crates.io 精确版本 + 提交 Cargo.lock + cargo --locked,
# 可复现构建不再依赖 git 源，供应链来源随之收口。
# dynamo-tokenizers 1.8.1 依赖 Rust 1.92 标准库 floor_char_boundary,
# 因此本 PR 同步把 rust-toolchain.toml 从 1.90 提升到 1.92。
dynamo-tokenizers = "=1.8.1"

# dynamo-renderer 5.1.2 为渲染器迁移预先落地，当前代码尚未引用；
# 属于 PR 栈 #39457 → #39458 → ... → #38983 的前置依赖步骤。
dynamo-renderer = "=5.1.2"

# Chat-template 渲染（cache-aware routing 依赖）：引擎在应用模型聊天模板
# 之后才缓存 token，router 必须在哈希前渲染同一模板，否则查询 token_ids
# 与引擎存储的 block 不一致。minijinja + pycompat 提供模板调用所需的
# Python str/dict 语义；chrono 支撑 strftime_now 日期注入。
minijinja = { version = "2.24", features = ["loop_controls", "json"] }
minijinja-contrib = { version = "2", features = ["pycompat"] }
chrono = { version = "0.4", default-features = false, features = ["clock"] }

```

experimental/sgl-router/deny.toml

供应链策略收紧的核心文件：禁止 git 源、unmaintained 改为仅检查直接依赖、删除不再在依赖图中的许可例外，并记录每条策略的权衡理由。

```

[graph]
all-features = true

[advisories]
yanked = "warn"
# unmaintained 只对直接依赖生效：传递依赖中无法就地修复的 unmaintained
# crate（paste 经 tokenizers、number_prefix 经 indicatif、backoff/instant 经
# kube-runtime、rustls-pemfile 经 kube-client）只能等上游移除后再收紧。
# ignore 保持为空，新 CVE 依然会告警。
unmaintained = "workspace"
ignore = []

[licenses]

```

```

# 许可白名单随依赖图演进：移除 NCSA (libfuzzer-sys/rav1e 已不在图内)，
# 保留 CDLA-Permissive-2.0 (webpki-roots)。
allow = [
    "Apache-2.0",
    "Apache-2.0 WITH LLVM-exception",
    "MIT",
    "BSD-2-Clause",
    "BSD-3-Clause",
    "ISC",
    "Unicode-DFS-2016",
    "Unicode-3.0",
    "Zlib",
    "CC0-1.0",
    "MPL-2.0",
    "CDLA-Permissive-2.0",
]
confidence-threshold = 0.93

[bans]
multiple-versions = "warn"
wildcards = "deny"
# 仅剩 workspace path 依赖没有 semver 版本号，
# 其余 crates.io 依赖不允许裸 * 通配符。
allow-wildcard-paths = true

[sources]
unknown-registry = "deny"
# 已无 git 依赖 (dynamo-* 全部转为 registry 版本)，
# 供应链来源彻底收口，未来引入 git 依赖必须显式修改本策略。
unknown-git = "deny"

```

评论区精华

本 PR 没有任何 review 评论 (comments_count 与 review_comments_count 均为 0)，两位 reviewer (Kangyan-Zhou、ispobock) 直接 APPROVED 且未附说明。设计决策全部沉淀在 commit message 与配置内联注释中：

- 工具链升级是依赖硬约束而非可选优化 (dynamo-tokenizers 1.8.1 需要 floor_char_boundary)；
- rustls 0.23.45 升级是被 cargo deny 的 RUSTSEC-2026-0285 检查卡住后的必要动作；
- deny.toml 中 unmaintained 改为 workspace 是有意识的权衡：paste、number_prefix、backoff、instant、rustls-pemfile 等传递依赖的 unmaintained 状态无法从本仓库修复，只能等上游移除，同时刻意保持 ignore 为空以确保新 CVE 仍会告警。
- 暂无高价值评论线程

风险与影响

- 风险：

1. 工具链升级风险：所有使用 rust-toolchain.toml 的 sgl-router 构建（CI、本地、Docker）必须能在 Rust 1.92 下工作，若 CI runner 镜像或 rustup 缓存滞后会立即失败；影响范围被 toolchain 文件限制在 experimental/sgl-router 目录内。
2. 锁文件策略切换：Cargo.lock 落地后依赖变更必须显式更新并提交锁文件，否则 --locked 构建失败；对习惯在容器内生成 lockfile 的本地开发流程是行为变化，换来的是构建与 CI 严格一致。
3. 供应链收紧的摩擦点：unknown-git = deny 意味着未来引入任何 git 依赖都必须先修改策略，这是刻意的安全取舍；registry 精确版本 = 1.8.1 即使在 crates.io 被 yank，已提交的锁文件仍可完成解析构建，风险较低。
4. 新增未使用依赖：dynamo-renderer 5.1.2 当前零引用，若迁移计划变更或版本不兼容需及时清理；其传递依赖会增加编译时间与镜像体积。
5. 告警盲区：unmaintained 改为 workspace 后传递依赖的 unmaintained 状态不再告警，监控面缩小，注释中已说明等待上游修复。
6. CI Extra 状态：PR 状态栏显示 Extra 任务为 X（无详情），主 PR Test 为 ✓；材料中无足够信息判断失败原因，建议合并后留意后续任务是否持续失败。- 影响：影响范围集中在 experimental/sgl-router 及其 CI、镜像链路，不影响 Python 运行时与其他模块；根 .gitignore 变更影响全仓库 clone 行为但仅是忽略规则。对用户无功能可见影响；对团队而言，sgl-router 贡献者必须采用 Rust 1.92 + 提交锁文件的开发流程。对后续演进，本 PR 解锁了 dynamo-renderer 迁移 PR 栈（#39458 → ... → #38983），是 sgl-router 侧对齐引擎 chat-template 缓存语义的关键前置。- 风险标记：工具链升级，构建锁定策略变更，供应链来源收紧，新依赖暂未使用，传递依赖告警盲区

关联脉络

- PR #38939 [Rust] Use Dynamo native renderers when chat templates are missing: 同一 Dynamo 渲染能力演进线程：该 PR 在 rust/sglang-server 接入 Dynamo native renderers，本 PR 为 sgl-router 侧同能力迁移准备依赖与供应链基础。