

PR #33257 完整报告

sgl-project/sglang

Revert "[rust-server] fix TCP-layer TTFT stalls"

合并时间: 2026-08-02 16:20

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/33257>

PR #33257 分析: Revert "[rust-server] fix TCP-layer TTFT stalls"

执行摘要

本 PR 完全回滚 #33026 对 rust-server 的 TCP 层优化, 恢复旧 socket 绑定与默认 TCP 行为。回滚原因未在 PR 中说明, 但合并后 rust-server 网络路径回归到 #33026 之前的稳定基线, 同时可能重新引入原修复针对的 TTFT 卡顿问题。

功能与动机

PR body 只有一句 "Reverts sgl-project/sglang#33026", 没有展开原因。结合上下文可以推断: #33026 为修复 rust-server 的 TCP 层 TTFT 卡顿, 设置了 16 MiB 的 `SO_RCVBUF`, 并在 keep-alive 连接上强制 `TCP_NODELAY`。这些改动可能带来内存占用上升、与某些环境的 socket 行为不兼容等副作用, 因此团队选择整体回滚, 先恢复已验证的稳定路径。

实现拆解

1. runtime.rs (核心逻辑): Runtime::start 从原先的 tokio::net::TcpSocket 预绑定 (含 set_reuseaddr、set_recv_buffer_size(16 MiB)) 改回标准的 std::net::TcpListener::bind 同步绑定, 并调用 set_nonblocking(true) 交由 tokio 接管。改动保证端口冲突在启动期即暴露, 同时移除了自定义 rcvbuf 设置。
2. api_server.rs (入口层): serve 函数签名从 tokio::net::TcpSocket 改回 std::net::TcpListener, 内部用 tokio::net::TcpListener::from_std 将同步 listener 接入异步 reactor; 同时删除 ListenerExt::tap_io 与 set_nodelay(true) 调用, 恢复 Nagle 算法默认行为。
3. 配套改动: 无测试、配置或部署文件变更, 纯代码回滚。

rust/sglang-server/src/runtime.rs

回滚核心逻辑: 恢复 std TcpListener 同步绑定, 移除 SO_RCVBUF 设置, 这是原修复的主要改动之一。

// 绑定 API

服务端口: 必须在任何线程派生前完成, 使端口占用 (EADDRINUSE) 成为启动期的硬错误。

```
let listener = std::net::TcpListener::bind(cfg.rust_server_args.http_addr)
    .map_err(|e| format!("bind {} failed: {}", cfg.rust_server_args.http_addr, e))?;
```

// 必须设置非阻塞, 才能交由 tokio 的 reactor 驱动。

```

listener
    .set_nonblocking(true)
    .map_err(|e| format!("listener set_nonblocking failed: {e}"))?;
// ... 后续线程派生与 runtime 构建 ...
rt.block_on(api_server::serve(
    listener,
    senders,
    cfg.rust_server_args.channel_cap,
    cfg.server_args.clone(),
    api_activity,
    shutdown_rx,
))

```

rust/sclang-server/src/api_server.rs

回滚入口改动：serve 函数改回接受 std TcpListener，并移除 tap_io set_nodelay 优化。

```

pub async fn serve(
    listener: std::net::TcpListener, // 由 runtime::start 同步绑定后传入
    senders: Senders,
    egress_buf: usize,
    server_args: Arc<ServerArgs>,
    egress_activity: ActivityCounter,
    shutdown: flume::Receiver<()>,
) {
    // ... AppState 构建与路由注册 ...

    // listener 已在 runtime::start 中同步绑定（端口冲突直接启动失败），
    // 这里通过 from_std 把它接入 tokio 的 reactor 开始监听。
    let listener = match tokio::net::TcpListener::from_std(listener) {
        Ok(l) => l,
        Err(e) => {
            tracing::error!(error = %e, "failed to adopt pre-bound listener");
            return;
        }
    };

    // 回滚后不再设置 TCP_NODELAY，恢复默认 Nagle 行为（原 #33026 曾用
    // tap_io(set_nodelay) 消除 keep-alive 连接上的 ~13 ms 延迟）。
    let serve = axum::serve(
        listener,
        app.into_make_service_with_connect_info::<std::net::SocketAddr>(),
    );
    // ...
}

```

评论区精华

本 PR 没有实质 review 评论。两条评论中，[gemini-code-assist\[bot\]](#) 是服务下线通知，[sherlockwu](#) 只是触发了 extra CI 重跑，均不涉及技术讨论。回滚原因未在公开渠道说明。

风险与影响

- 功能回退风险：#33026 声称修复的 TCP 层 TTFT 卡顿将重现。大请求体 (> 约 90 KB) 首包可能因内核 rcvbuf 不足产生约 200 ms 延迟；keep-alive 连接因 Nagle 与 delayed-ACK 叠加增加约 13 ms 延迟。
- 稳定性收益：恢复的是此前长期运行的绑定流程，无新增风险，对依赖稳定 socket 行为的用户更安全。
- 影响范围：仅影响 rust-server 的 HTTP 前端路径，Python server 无变化。

关联脉络

本 PR 直接回滚 #33026。该回滚说明 #33026 的优化未被团队接受，后续大概率会出现以更保守方式（例如可配置开关、分阶段放量）重新提交的修复。关注 rust-server 性能的团队应留意后续 PR。