

PR #47965 完整报告

vllm-project/vllm

[Rust Frontend] Wait for mock engine endpoints before ZMQ connect

合并时间: 2026-07-16 17:20

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/47965>

执行摘要

修复 Rust mock engine 在端点就绪前的竞态等待问题: 通过原始 TCP/IPC 连接探测替代仅检查文件存在性, 消除了 ZMQ 内部重试延迟, 将测试总耗时从约 1.4s 降至约 0.03s (50x 加速), 提升测试确定性。

功能与动机

mock engine 启动时可能因前端端点尚未绑定就执行 ZMQ connect, 导致连接被拒绝。ZMQ 库对拒绝连接进行退避重试 (backoff), 即使端点很快准备就绪, 某些测试仍要等待约 1.4s。原逻辑的问题在于:

- TCP 端点: 完全跳过预检, 直接 ZMQ connect。
- IPC 端点: 仅检查 socket 文件是否存在, 但文件创建到 listener 就绪之间仍有窗口。

PR 作者在描述中明确指出了这一竞态及其测试开销。

实现拆解

1. 重构等待函数: 将 `wait_for_ipc_endpoint` 重写为 `wait_for_endpoint`, 内部根据 endpoint 前缀分派不同连接策略。
 - `ipc://`: 使用 `tokio::net::UnixStream::connect` 尝试连接, 而非仅检查文件存在。
 - `tcp://`: 新增分支, 使用 `tokio::net::TcpStream::connect` 探测 (原无任何等待)。
 - 其他 scheme: 跳过等待 (向后兼容)。
2. 更新所有调用点: `connect_to_frontend` 中原来对 `handshake` 端点和后续 `input/output` 地址的三次 `wait_for_ipc_endpoint` 调用全部替换为 `wait_for_endpoint`。
3. 调整导入与文档: 移除不再需要的 `std::path::Path`, 合并 `tokio::time` 导入; 更新 `MockEngineConfig::connect_timeout` 的注释以反映新语义。

变更仅涉及单个文件 `rust/src/engine-core-client/src/mock_engine.rs`, +34/-24 行。

`rust/src/engine-core-client/src/mock_engine.rs`

唯一变更文件; 重写了端点等待逻辑, 新增 TCP 端点支持, 替换了 IPC 路径存在性检查为真实连接探测。

```
/// Wait for an endpoint to accept connections before attempting the ZMQ connect.
async fn wait_for_endpoint(endpoint: &str, connect_timeout: Duration) -> Result<> {
    // 处理 IPC 端点: 尝试连接 UnixStream, 而非仅检查路径存在
    if let Some(socket_path) = endpoint.strip_prefix("ipc://") {
        timeout(connect_timeout, async {
```

```

        while tokio::net::UnixStream::connect(socket_path).await.is_err() {
            sleep(Duration::from_millis(20)).await;
        }
    })
    .await
    .map_err(|_| Error::HandshakeTimeout {
        stage: "mock engine IPC endpoint",
        timeout: connect_timeout,
    })
} else if let Some(address) = endpoint.strip_prefix("tcp://") {
    // 新增 TCP 端点支持: 尝试原始 TCP 连接直到成功或超时
    timeout(connect_timeout, async {
        while tokio::net::TcpStream::connect(address).await.is_err() {
            sleep(Duration::from_millis(20)).await;
        }
    })
    .await
    .map_err(|_| Error::HandshakeTimeout {
        stage: "mock engine TCP endpoint",
        timeout: connect_timeout,
    })
} else {
    // 未知 scheme 直接返回, 向后兼容
    Ok(())
}
}

// 在 connect_to_frontend 入口处替换调用
// 原: wait_for_ipc_endpoint(engine_handshake, config.connect_timeout).await?;
// 新:
wait_for_endpoint(engine_handshake, config.connect_timeout).await?;

// 后续对 input/output 地址的等待也改为 wait_for_endpoint
// 原: wait_for_ipc_endpoint(input_address, config.connect_timeout).await?;
// 新:
wait_for_endpoint(input_address, config.connect_timeout).await?;
wait_for_endpoint(output_addre...

```

评论区精华

- BugenZhao 触发 @codex review, codex 未发现 issues。
- BugenZhao 直接 APPROVED 并留言 "Thanks!"。

风险与影响

风险: 低。新逻辑仅在 mock engine 测试中使用, 超时沿用已有配置 (默认 5s), 无新增依赖, 未改变连接后流程。

影响:

- 所有 mock engine 测试（7 个用例）运行时间从约 1.4s 骤降至约 0.03s，提升 50x。
- 测试结果更稳定，消除因端点竞争导致的偶发失败。
- 仅影响测试基础设施，生产环境无任何变更。

关联脉络

- 本 PR 与后续 PR #48738 (Fix mock engine test shutdown race) 由同一作者和 reviewer 维护，共同改进 mock engine 测试的稳定性。
- PR #48738 的讨论中作者提及本 PR，建议 reviewer 优先审查。
- 本 PR 是 Rust 前端测试基础设施演进中的一环，与近期多个 Rust 前端质量改进 PR 形成清晰脉络。