PR #52703 完整报告

vllm-project/vllm

[Rust Frontend][RL] add routed expert prompt offset

合并时间：2026-08-19 04:04

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/52703

执行摘要

本 PR 为 Rust EngineCore 采样协议新增 routed_experts_prompt_start 字段，并在 Rust 前端参数映射中固定发送默认值 0，以对齐 Python SamplingParams.routed_experts_prompt_start 的语义。同步更新了 Rust 客户端与 Python 兼容 fixture 测试，覆盖默认值与非零值两种编码场景。当前变更不改变任何用户可见行为，是为后续 RL routed-expert 特性铺路的协议基础。

功能与动机

PR body 明确说明："This PR lets Rust send routed_experts_prompt_start into Python EngineCore. This tells Python how much of the prompt's routing information has already been consumed and therefore where capture should start." 即 Rust 前端需要把 prompt 中已消费的 routing 信息量告知 Python EngineCore，以便路由专家数据捕获从正确位置开始。

实现拆解

1. 协议层扩展：在 rust/src/engine-core-client/src/protocol/sampling.rs 的 EngineCoreSamplingParams 结构体中新增 routed_experts_prompt_start: u32 字段，并在 for_test() 构造函数中初始化为 0，同时补充文档注释说明"0 表示返回整个 prompt 的 routing 数据"。
2. 前端参数映射：在 rust/src/text/src/lower.rs 的 lower_sampling_params 函数中构造 EngineCoreSamplingParams 时显式填入 routed_experts_prompt_start: 0。由于 Rust 前端尚未暴露该参数的用户入口，这里暂时硬编码默认值；同时更新了多处 expect_test 快照，确保字段不会因遗漏而丢失。
3. 双向兼容测试：在 rust/src/engine-core-client/src/tests/client.rs 的 sample_request_with_id 中设置非零偏移量 1，验证 Rust 编码能携带非默认值；在 python_compat.py 的 Python 镜像 struct 中同步新增该字段并在请求样例中设置 1，确保 Rust 与 Python 的 msgpack 编码 fixture 完全一致，同时保留全默认请求的解码回归测试。

rust/src/engine-core-client/src/protocol/sampling.rs

协议定义主文件，新增 routed_experts_prompt_start 字段并初始化默认值，是整个变更的核心入口。

```
// EngineCoreSamplingParams 是 Rust 前端发送给 Python EngineCore 的采样参数集合。
// 本次新增 `routed_experts_prompt_start` 字段，用于告知引擎端 prompt 中已有多少
```

```
// routing 信息被消费，从而决定 routed-expert 数据 capture 的起始位置。
pub struct EngineCoreSamplingParams {
    // ... 既有字段（temperature、top_p、stop_token_ids 等）省略 ...

    /// Number of prompt tokens to skip from returned routed-expert data.
    /// A value of zero returns routing data for the entire prompt.
    pub routed_experts_prompt_start: u32,
}

impl EngineCoreSamplingParams {
    /// 仅用于测试的默认构造：新字段默认 0，表示返回整个 prompt 的 routing 数据。
    pub fn for_test() -> Self {
        Self {
            // ... 其他字段填充默认值 ...
            // 协议要求该字段始终存在，默认 0 与 Python 侧行为一致。
            routed_experts_prompt_start: 0,
        }
    }
}
```

rust/src/text/src/lower.rs

前端参数降级映射所在文件，lower_sampling_params 构造 EngineCoreSamplingParams 时固定填充 0，并同步更新大量 expect 快照。

```
// `lower_sampling_params` 负责将高层的 SamplingParams 转换为 EngineCore 协议参数。
// 当前 Rust 前端尚未暴露 `routed_experts_prompt_start` 的用户入口，
// 因此这里固定发送 0（等价于“对整个 prompt 返回 routing 数据”），
// 先让协议字段对齐 Python 的 `SamplingParams.routed_experts_prompt_start`。
let params = EngineCoreSamplingParams {
    // ... 其他字段保持不变 ...
    logprob_token_ids,
    skip_reading_prefix_cache,
    extra_args: vllm_xargs,
    // TODO: 后续 PR 应从用户请求中解析真实偏移量并透传，
    // 以支持 RL 场景下跳过已消费的 routing 信息。
    routed_experts_prompt_start: 0,
};

// 同步更新的 expect_test 快照会包含 `routed_experts_prompt_start: 0`，
// 用于防止字段丢失或默认值被意外改动。
validate_resolved_sampling_params(&params)?;
validate_vocab_range(&params, &sampling_limits)?;
Ok(params)
```

评论区精华

- claude[bot] 提示："This pull request is from a fork — automated review is disabled."，即 fork 来源 PR 无法自动审查。
- njhill 直接批准并回复："Thanks @biswapanda"，无进一步技术讨论。

整体而言，该 PR 没有产生设计交锋，评审过程顺利。

风险与影响

- 协议兼容风险：新增字段为必填 u32，所有 Rust 侧构造点都必须显式赋值，编译期可强制发现遗漏；但若未来新增其他协议字段，需要同步维护 Rust 结构、Python fixture 和 *expect* 快照三处，容易遗漏。
- 功能不完整风险：lower.rs 中硬编码 0，Rust 前端用户目前无法真正设置偏移量，RL *routed-expert capture* 起点仍固定为 0，实际能力尚未落地。
- 验证不充分风险：测试只验证了编码一致性和默认值回填，没有验证 Python EngineCore 真实消费该字段后的行为是否符合预期。

对用户的影响为无行为变化；对团队的影响是确立了一个协议字段同步的参考模板。

关联脉络

本 PR 是 Rust Frontend 系列对齐 Python 行为工作的一部分。历史 PR 如 #52671（等待 utility 调用完成）、#52575（简化 *data-parallel size* 归属）以及 #51426（GLM-5.2 模板渲染对齐）都涉及 *engine-core-client* 协议结构与测试同步，体现了 Rust 前端逐步补齐与 Python EngineCore 功能对等性的演进方向。本 PR 为该系列中面向 RL 路由专家特性的协议基础。