

PR #45465 完整报告

vllm-project/vllm

[Bugfix][Rust Frontend] Make metrics respect --served-model-name

合并时间: 2026-06-15 16:05

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/45465>

执行摘要

修复 Rust 前端指标忽略 `--served-model-name` 的 bug。通过在 `build_state` 中提前解析 served model name 并传递给 engine client, 确保 `model_name` 标签使用公共名称。新增测试覆盖验证正确性。

功能与动机

Rust 前端接受 `--served-model-name` 参数, 但 Prometheus 指标仍使用后端模型路径 (如 `Qwen/Qwen3-0.6B`) 作为 label, 导致监控混乱。本 PR 要求指标标签使用用户指定的公共名称 (如 `my-model`) 。

实现拆解

1. 提取公共函数: 新增 `effective_served_model_names`, 逻辑同原来内联代码, 但作为独立函数便于测试。
2. 调整构建状态顺序: 在 `build_state` 最前解析 served model names, 并提取第一个作为 `metrics_model_name`。
3. 修改核心配置: 将 `EngineCoreClientConfig::model_name` 从 `config.model` 改为 `metrics_model_name`。
4. 添加测试: 在 `tests.rs` 中新增集成测试, 模拟请求并验证 `vllm:request_success_total` 的 `model_name` 标签为 served name。

rust/src/server/src/lib.rs

核心逻辑变更: 新增 `effective_served_model_names` 函数, 修改 `build_state` 中模型名称解析和传递, 直接影响指标标签。

```
/// Resolve the public model names accepted by the frontend.
fn effective_served_model_names(model: &str, served_model_name: &[String]) -> Vec<String> {
    // 如果用户没有指定 served model name, 则回退到后端模型路径
    if served_model_name.is_empty() {
        vec![model.to_string()]
    } else {
        // 使用用户指定的公共名称作为指标标签
        served_model_name.to_vec()
    }
}
```

```

/// Build the shared application state for one configured model and one engine client.
async fn build_state(config: &Config) -> Result<Arc<AppState>> {
    // 先解析有效的 served model names, 并取第一个作为 metrics 标签
    let served_model_names = effective_served_model_names(&config.model, &config.served_model_name);
    let metrics_model_name = served_model_names[0].clone();

    // ... 其他初始化逻辑 ...

    // 关键变更: 将 EngineCoreClientConfig 的 model_name 从 config.model 改为 metrics_model_name
    let client = EngineCoreClient::connect(EngineCoreClientConfig {
        transport_mode: config.transport_mode.clone(),
        coordinator_mode,
        model_name: metrics_model_name, // 之前是 config.model.clone()
        client_index: 0,
    })
    .await
    .context("failed to connect to engine core")?;

    // ...
}

```

rust/src/server/src/routes/tests.rs

新增集成测试, 验证指标 label 正确性, 确保变更有效且防止回归。

```

#[tokio::test(flavor = "multi_thread", worker_threads = 2)]
#[serial]
async fn request_metrics_use_served_model_name_label() {
    // 建立 mock engine, 模拟请求响应
    let ipc = IpcNamespace::new().expect("create ipc namespace");
    // ... 初始化 ...
    let client = EngineCoreClient::connect(
        EngineCoreClientConfig::new_single(handshake_address)
            .with_model_name("served-model-metrics") // 模型标签为 served name
            // ...
    )
    .await
    .expect("connect client");
    // 创建 AppState 时使用两个 served model names, 第一个作为指标标签
    let mut app = build_router(Arc::new(AppState::new(
        vec![
            "served-model-metrics".to_string(),
            "served-model-alias".to_string(),
        ],
        chat,
    )));
    let before = METRICS.render().unwrap();

    // 发送请求时使用第二个名称 (别名), 但指标应记录第一个作为 model_name
}

```

```

let response = app
  .call(
    Request::builder()
      .method("POST")
      .uri("/v1/chat/completions")
      .header("content-type", "application/json")
      .body(Body::from(
        json!({
          "model": "served-model-alias", // 请求使用别名
          // ...
        })
      ).to_string(),
    ))
    .expect("build request"),
  )
  .await
  .expect("call app");

assert_eq!(response.status(), StatusCode::OK);
// 渲染指标并断言指标包含正确的 model_name 标签
let after = METRICS.render().unwrap();
assert_eq!(
  metric_delta(
    &before,
    &after,
    "vllm:request_success_total",
    Some("model_name=\"served-model-metrics\",engine=\"0\",finished_reason=\"stop\""),
  ),
  1.0
);
// ...
}

```

评论区精华

无讨论，直接 LGTM。

风险与影响

- 风险：低，不涉及核心推理路径，有测试覆盖。
- 影响：只影响 Rust 前端且使用 `--served-model-name` 的用户，指标标签变得正确。

关联脉络

该 PR 为独立 bugfix，无直接关联的其他 PR 或 Issue。