# PR #46582 完整报告

vllm-project/vllm

[Rust Frontend] Raise frontend JSON body limit

合并时间：2026-06-24 20:15

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/46582

## 执行摘要

该 PR 将 Rust 前端 Axum 的请求体大小限制从默认的 2 MiB 提升到 32 MiB，并添加回归测试确保大请求可通过。这是对 1M 上下文模型场景的修复。

## 功能与动机

当使用支持 1M 上下文长度的模型时，客户端请求体可能超过 Axum 默认的 2 MiB 限制，导致请求被拒绝并报错 Failed to buffer the request body: length limit exceeded。PR 旨在消除这一人为瓶颈，使大请求能够正常处理。

## 实现拆解

1. 提升限制（rust/src/server/src/routes.rs）：导入 axum::extract::DefaultBodyLimit，定义常量 DEFAULT_JSON_BODY_LIMIT_BYTES = 32 * 1024 * 1024，并在路由器上添加 DefaultBodyLimit::max(DEFAULT_JSON_BODY_LIMIT_BYTES) 层。
2. 回归测试（rust/src/server/src/routes/tests.rs）：新增 chat_completions_accepts_request_body_larger_than_axum_default 测试，构造一个超过 2 MiB 的请求体（通过 chat_template_kwargs 注入 2 MiB 的字符串），验证返回 200 OK。

### rust/src/server/src/routes.rs

核心变更：提升请求体限制。

```
// 增加请求体限制，避免大上下文请求被拒绝
use axum::extract::DefaultBodyLimit;

// 32 MiB，可覆盖 1M 上下文模型的请求
const DEFAULT_JSON_BODY_LIMIT_BYTES: usize = 32 * 1024 * 1024;

// 在构建路由器时添加限制层
let mut router = router
    .with_state(state.clone())
    .layer(DefaultBodyLimit::max(DEFAULT_JSON_BODY_LIMIT_BYTES)) // 关键新增
    .layer(middleware::request_runtime_layer(state.clone()))
    // ... 后续中间件
```

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

回归测试验证大请求体可通过。

```
#[tokio::test(flavor = "multi_thread", worker_threads = 2)]
#[serial]
async fn chat_completions_accepts_request_body_larger_than_axum_default() {
    let (chat, engine_task) = test_chat_with_engine_outputs(
        b"engine-openai-chat-large-body",
        default_stream_output_specs(),
    )
    .await;
    let mut app = build_router(Arc::new(AppState::new(
        vec!["Qwen/Qwen1.5-0.5B-Chat".to_string()],
        chat,
    )));

    // 构造超过 2 MiB 的请求体
    let large_template_arg = "a".repeat(2 * 1024 * 1024); // 正好 2 MiB
    let response = app
        .call(
            Request::builder()
                .method("POST")
                .uri("/v1/chat/completions")
                .header("content-type", "application/json")
                .body(Body::from(
                    json!({
                        "model": "Qwen/Qwen1.5-0.5B-Chat",
                        "stream": false,
                        "messages": [{"role": "user", "content": "hello"}],
                        "chat_template_kwargs": {"large": large_template_arg} // 注入大字段
                    })
                    .to_string(),
                ))
                .expect("build request"),
        )
        .await
        .expect("call app");

    let status = response.status();
    let body = to_bytes(response.into_body(), usize::MAX).await.expect("read body");
    // 断言状态码为 200，否则打印响应体
    assert_eq!(status, StatusCode::OK, "{}", String::from_utf8_lossy(&body));
    engine_task.await.expect("mock engine task");
}
```

## 评论区精华

审查者 BugenZhao 直接批准并评价 "Good catch. Thanks!"，无其他讨论。

## 风险与影响

- 风险：低。仅修改请求体限制，不影响解析逻辑；32 MiB 限制仍可防止无限内存占用。
- 影响：使 1M 上下文模型用户不再因限制丢失请求；仅影响 Rust 前端 OpenAI 路由。

## 关联脉络

该 PR 独立，未发现关联的其他 PR 或 Issue。作为基础设施修复，为后续更大上下文的支持扫清障碍。