

# PR #49295 完整报告

vllm-project/vllm

[Rust][Benchmark] Use async HTTP clients

合并时间: 2026-07-21 16:53

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/49295>

## 执行摘要

本 PR 将 Rust 基准测试工具中的 HTTP 客户端从阻塞 (`reqwest::blocking`) 全面迁移到异步 (`reqwest`)，并由此移除了 `blocking` 特性依赖。核心改动包括 `ServerTokenizer` 和 `HubRepo` 两个模块的异步化，以及测试套件的适配。解决了在 Tokio 运行时内丢弃阻塞客户端可能导致 panic 的问题。

## 功能与动机

原始实现中，未缓存的下载创建了 `reqwest::blocking::Client`，而基准测试运行在 Tokio 运行时内部。丢弃阻塞客户端时会关闭其内部运行时，从异步上下文触发 panic。本 PR 通过使用 `async reqwest` 及 `block_on` 桥接同步与异步，消除此风险。

## 实现拆解

1. `ServerTokenizer` 异步化：将 `client` 字段从 `reqwest::blocking::Client` 改为 `reqwest::Client`，新增 `runtime: tokio::runtime::Handle`。原有的同步方法 `encode_inner / decode_inner` 拆分为同步外皮（通过 `block_on` 调用异步核心）和异步核心 `encode_async / decode_async`。
2. `HubRepo` 重写：去除每下载一个文件就启动一个线程并在其中构建 tokio 运行时的复杂模式。直接使用 `hf_hub::api::tokio::ApiRepo`，`get` 方法变为 `async`，构造函数完成一次 API 初始化。
3. 调用链适配：所有依赖 `HubRepo::get` 的代码（`tiktoken.rs`、`hf_dataset.rs`、`speed_bench.rs`、`sharegpt.rs` 等）均改为 `async`，使用 `.await` 调用下载。涉及函数签名如 `try_load_tiktoken`、`download_hf_dataset`、`get_with_retry` 等。
4. 测试适配：`#[test]` 改为 `#[tokio::test]`，测试中调用 `load_tokenizer` 等异步函数时使用 `.await`。部分测试（如 `random.rs` 中的 `test_generate_random_dataset_token_ids`）改用内置 `Tiktoken` 避免网络依赖。
5. 依赖清理：从 `Cargo.toml` 中移除 `reqwest/blocking` 特性。

## HubRepo 异步化 (rust/src/bench/src/hub.rs)

```
use hf_hub::api::tokio::{ApiBuilder, ApiRepo};
```

```
pub struct HubRepo {  
    repo: ApiRepo,  
}
```

```
impl HubRepo {
    fn new(repo: Repo) -> Result<Self, String> {
        let mut builder = ApiBuilder::from_env();
        if let Ok(token) = std::env::var("HF_TOKEN") {
            builder = builder.with_token(Some(token));
        }
        let api = builder.build().map_err(|e| format!("Failed to init HF API: {e}"))?;
        Ok(Self { repo: api.repo(repo) })
    }

    pub async fn get(&self, filename: &str) -> Result<PathBuf, String> {
        self.repo.get(filename).await.map_err(|e| format!("{e}"))
    }
}
```

## 评论区精华

无实质性讨论。

## 风险与影响

- 风险：block\_on 在 current\_thread 运行时如果同步接口被异步上下文调用可能导致死锁，但仅发生在 server tokenizer 回退路径，风险可控。异步客户端可能改变错误处理语义（如超时抛出不同错误）。
- 影响：解决了一个间歇性 panic，提升基准测试稳定性。移除 blocking 特性使得依赖更加简洁，为进一步全异步化铺平道路。对用户无感知。

## 关联脉络

本 PR 是 Rust 基准测试模块持续优化的延续。上一阶段（PR #48937）改进了日志系统（tracing），本次将网络调用异步化。后续可能继续消除 block\_on 桥接，实现完全异步的 Tokenizer 接口。