PR #47173 完整报告

vllm-project/vllm

[Frontend] Add /abort_requests to the RLHF dev API router

合并时间：2026-07-12 14:21

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/47173

执行摘要

- 一句话：为 RLHF dev API 路由添加 /abort_requests 端点
- 推荐动作：值得精读，特别是跨语言实现相同语义的设计（Python 直接收集内部 ID，Rust 通过空向量表示中止所有）。注意 Rust 端跟踪时机窗口问题，未来需关注是否引入锁或调整注册时机。

功能与动机

RL rollout 框架需要在步骤边界丢弃过采样 / 长尾生成，同时保留部分输出。现有的 /pause?mode=abort 会暂停调度器需要 /resume，使用不便。disagg 路由器已有等效 /abort_requests，现在为 RLHF 开发路由器添加相同能力。

实现拆解

1. Python API 路由层：在 vllm/entrypoints/serve/dev/rlhf/api_router.py 中添加 POST /abort_requests 异步处理函数。解析请求体，若 request_ids 存在则直接调用 engine.abort(request_ids)；若缺失则从 AsyncLLM.output_processor 收集所有内部请求 ID（包括父请求 ID），并调用 engine.abort(request_ids, internal=True) 以中止所有请求。同时处理 JSONDecodeError 返回 400。
2. Rust 前端路由层：在 rust/src/server/src/routes/abort_requests.rs 中将 request_ids 从必填改为可选，使用 unwrap_or_default() 替代 ok_or_else 错误返回，空 body 时传递空向量给后端。
3. Rust LLM 内核：在 rust/src/llm/src/lib.rs 中扩展 Llm::abort 方法：当 external_ids 为空时调用 self.inflight.all_internal_ids() 收集所有跟踪的内部 ID，否则使用原 resolve 逻辑。
4. 跟踪机制：在 rust/src/llm/src/inflight.rs 中新增 all_internal_ids 方法，从锁保护的 HashMap 中收集所有内部 ID。
5. 测试与文档：在 rust/src/server/src/routes/tests.rs 中将空 body 预期从 400 改为 200 并重命名测试函数；同步更新三份文档文件添加端点说明。

关键文件：

- vllm/entrypoints/serve/dev/rlhf/api_router.py（模块 API 路由；类别 source；类型 entrypoint；符号 abort_requests）：核心入口，添加 POST /abort_requests 端点，处理指定或全部请求中止，并包含关键修复（internal=True、parent IDs、JSON 拒绝）。

- rust/src/server/src/routes/abort_requests.rs（模块 Rust 路由；类别 source；类型 entrypoint）：Rust 前端路由，修改 request_ids 从必填变为可选，实现空 body 转发。
- rust/src/llm/src/lib.rs（模块 Rust 内核；类别 source；类型 core-logic）：Rust LLM 核心，扩展 abort 方法以支持空 external_ids 时中止所有请求。
- rust/src/llm/src/inflight.rs（模块 请求跟踪；类别 source；类型 core-logic）：新增 all_internal_ids 方法，支持获取所有在途请求 ID。
- rust/src/server/src/routes/tests.rs（模块 测试套件；类别 source；类型 test；符号 abort_requests_route_rejects_missing_request_ids, abort_requests_route_aborts_all_when_request_ids_missing）：更新测试用例，反映空 body 行为从 400 变为 200。
- docs/serving/online_serving/README.md（模块 文档；类别 docs；类型 documentation）：新增端点文档说明。
- docs/training/async_rl.md（模块 文档；类别 docs；类型 documentation）：同步更新文档。
- docs/usage/security.md（模块 文档；类别 docs；类型 documentation）：同步更新文档。

关键符号：abort_requests, abort_requests (Rust route), abort (Llm), all_internal_ids (InflightRequests), abort_requests_route_rejects_missing_request_ids (test), abort_requests_route_aborts_all_when_request_ids_missing (test)

关键源码片段

vllm/entrypoints/serve/dev/rlhf/api_router.py

核心入口，添加 POST /abort_requests 端点，处理指定或全部请求中止，并包含关键修复（internal=True、parent IDs、JSON 拒绝）。

```
@router.post("/abort_requests")
async def abort_requests(raw_request: Request) -> JSONResponse:
    """Abort in-flight requests without pausing the scheduler.

    Empty/missing `request_ids` aborts all in-flight requests.
    """
    engine = engine_client(raw_request)

    # Parse JSON body; reject malformed JSON explicitly
    try:
        body = await raw_request.json()
    except json.JSONDecodeError as e:
        raise HTTPException(status_code=400, detail="Invalid JSON format") from e

    request_ids = body.get("request_ids")

    try:
        if request_ids:
            # User-supplied external IDs
            await engine.abort(request_ids)
```

```
        else:
            # Dev RL server uses AsyncLLM; gather all internal IDs
            from vllm.v1.engine.async_llm import AsyncLLM
            assert isinstance(engine, AsyncLLM)
            op = engine.output_processor
            # Include both child (request_states) and parent (parallel-sampling) IDs
            request_ids = [
                *op.request_states.keys(),
                *op.parent_requests.keys(),
            ]
            # Internal flag is required because these are internal suffixed IDs
            await engine.abort(request_ids, internal=True)
        return JSONResponse(
            content={"status": "aborted", "aborted": len(request_ids)},
            status_code=HTTPStatus.OK.value,
        )
    except Exception as err: # pragma: no cover - defensive
        logger.exception("Failed to abort requests")
        return JSONResponse(
            content={"error": f"Failed to abort requests: {err}"},
            status_code=HTTPStatus.INTERNAL_SERVER_ERROR.value,
        )
```

rust/src/server/src/routes/abort_requests.rs

Rust 前端路由，修改 request_ids 从必填变为可选，实现空 body 转发。

```
use std::sync::Arc;

use axum::Json;
use axum::extract::State;
use axum::extract::rejection::JsonRejection;
use axum::http::StatusCode;
use serde::Deserialize;

use crate::error::ApiError;
use crate::state::AppState;
use crate::utils::utility_call_error;

#[derive(Debug, Deserialize)]
pub(crate) struct AbortRequestsRequest {
    // `request_ids` is now optional; missing/unwrapped defaults to empty vec
    request_ids: Option<Vec<String>>,
}

pub async fn abort_requests(
    State(state): State<Arc<AppState>>,
    body: Result<Json<AbortRequestsRequest>, JsonRejection>,
) -> Result<StatusCode, ApiError> {
    let Json(body) = body.map_err(|error| ApiError::json_parse_error(error.body_text()))?;
```

```
    // Empty/missing `request_ids` aborts all in-flight requests.
    let request_ids = body.request_ids.unwrap_or_default();

    state
        .chat
        .abort(&request_ids)
        .await
        .map_err(|error| utility_call_error("abort_requests", error))?;

    Ok(StatusCode::OK)
}
```

rust/src/llm/src/lib.rs

Rust LLM 核心，扩展 abort 方法以支持空 external_ids 时中止所有请求。

```
    /// Abort in-flight requests by their external (user-supplied) request ids.
    ///
    /// External ids are resolved to the internal engine ids actually known to
    /// engine-core (one external id may map to several internal ids). Unknown
    /// or already-finished ids resolve to nothing and are a safe no-op. The
    /// tracking entries themselves are removed when the corresponding output
    /// streams are dropped, not here.
    pub async fn abort(&self, external_ids: &[String]) -> Result<()> {
        // Empty `external_ids` means abort every in-flight request.
        let internal_ids = if external_ids.is_empty() {
            self.inflight.all_internal_ids()
        } else {
            self.inflight.resolve(external_ids)
        };
        if internal_ids.is_empty() {
            return Ok(());
        }
        self.client.abort(&internal_ids).await?;
        Ok(())
    }
```

评论区精华

- 内部 ID 标记问题：Codex 评论指出空 body 路径需要设置 internal=True 否则 abort 无效，作者后续 commit 修复。
- Malformed JSON 处理：Codex 建议区分空 body 与 malformed JSON 避免误终止所有请求，作者通过捕获 JSONDecodeError 返回 400 修复。
- 并行采样父 ID 泄漏：Codex 指出中止所有时只收集子请求 ID 导致父条目残留，作者通过添加 parent_requests.keys() 修复。
- Rust 端跟踪时机：Codex 提出 abort-all 可能遗漏尚未跟踪的请求（因为 Llm::generate() 在 await 后才添加跟踪），未在 PR 中解决。
- 计数不准确：Codex 指出返回的 aborted 计数可能包含未实际中止的 ID，未修复。

- 空 body 时未使用 internal=True 导致 abort 无效 (correctness): 作者在后续 commit 中添加了 internal=True。
- Malformed JSON 被当作空 body 可能误终止所有请求 (correctness): 作者通过捕获 JSONDecodeError 并返回 400 修复。
- 并行采样父 ID 泄漏 (correctness): 作者通过添加 parent_requests.keys() 修复。
- Rust 端 abort-all 遗漏尚未跟踪的请求 (correctness): 未在 PR 中修复，可能认为影响有限或后续处理。

风险与影响

- 风险:
  1. 空 body 语义可能误终止所有请求，尤其在 Rust 端跟踪时机窗口内请求未完全注册时。
  2. 并行采样父 ID 泄漏问题已修复，但返回的计数仍可能不准确。
  3. 依赖 AsyncLLM 内部结构（output_processor、parent_requests），未来重构可能引发兼容性问题。
  4. Rust 端 all_internal_ids 仅覆盖已注册请求，generate() 中 client.call 之前发生的 abort-all 会遗漏。
- 影响:
  - 用户：RL 训练框架开发者获得更精细的请求控制，无需暂停 / 恢复调度器，简化 rollout 流程。
  - 系统：新增一个端点，对整体性能无影响。
  - 团队：需要维护 Python 和 Rust 两个实现，但逻辑一致，降低长期维护成本。
  - 风险标记：空 body 误终止，并行父 ID 泄漏（已修复），Rust 跟踪窗口遗漏，aborted 计数不准确

关联脉络

- PR #46725 Runtime Draft Weight Update for Speculative Decoding: 在同一个 RLHF 开发路由器上添加了 weight transfer 相关端点，本 PR 是同一系列的延续。