PR #44938 完整报告

vllm-project/vllm

[Rust Frontend] Support prompt-only completions

合并时间：2026-06-17 14:38

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/44938

执行摘要

此 PR 为 Rust 前端 /v1/completions 路由增加 prompt-only 请求支持，即当 echo=true 且 max_tokens=0 时返回 prompt 本身，不暴露生成 token。内部降级为一个 token 驱动引擎后隐藏，与 Python 行为对齐。

功能与动机

该 PR 是 Rust Frontend Feature Parity (#44280) 的一部分。Python vLLM 已支持 echo=true 时 max_tokens=0 的 prompt-only completions，Rust 前端需补齐此能力以消除功能差距。

实现拆解

1. 验证层放宽约束（validate.rs）：将 max_tokens=0 的拒绝条件改为 !echo，接受 echo=true 组合。
2. 请求转换层降级（convert.rs）：检测 echo=true && max_tokens==0 并设置 prompt_only 标志，强制 max_tokens=1 以驱动引擎；调整 prompt_logprobs 启用条件。
3. 主路由层响应处理（completions.rs）：根据 prompt_only 调整 text、token_ids、logprobs、usage，隐藏内部 token。
4. 辅助函数：新增 prompt_only_logprobs_to_openai 和 prompt_logprobs_to_maps。
5. 测试：在 validate.rs 和 convert.rs 中添加单元测试。

关键源码片段

rust/src/server/src/routes/openai/completions/convert.rs

请求转换层，检测 prompt_only 条件并设置标志，将 max_tokens 替换为 1 以驱动引擎。

```
// detect prompt-only condition: echo=true and max_tokens=0
let prompt_only = request.echo && request.max_tokens == Some(0);

// override max_tokens to 1 so the engine can run
let max_tokens = if prompt_only { Some(1) } else { request.max_tokens };

// enable prompt_logprobs for streaming prompt-only case
let prompt_logprobs = request.prompt_logprobs.or(if request.echo && (!request.stream ||
prompt_only) {
```

```
    logprobs
} else {
    None
});
```

评论区精华

- tahsintunan指出 completion_tokens 计数差异（Rust 初版 0 vs Python 1），作者更新为匹配 Python（completion_tokens=1）。
- tahsintunan建议清理不可达分支，作者重构并共享 logprobs 处理。
- BugenZhao要求为流式 prompt_only 分支加注释，作者已补充。
- chatgpt-codex-connector建议隐藏 token_ids 和 usage，作者对 token_ids 置空，但 usage 保留计数 1 以匹配 Python。

风险与影响

- 兼容性风险：completion_tokens=1 与 Python 行为一致，无破坏性变更。
- 日志风险：启用 enable_log_requests 时可能暴露内部 token 计数，影响有限。
- 回归风险：验证逻辑放宽，但单元测试覆盖充分。
- 影响范围：仅限于 Rust 前端 /v1/completions，功能对齐。

关联脉络

此 PR 是 Rust Frontend Feature Parity 路线图中的关键步骤，与 #44382（abort_requests）、#45848（反序列化修复）等共同完善 Rust 前端。关联 Issue #44280 追踪整体进度。