PR #49113 完整报告

vllm-project/vllm

[Bugfix][Rust Frontend] Handle zero-column logprobs payloads without panicking

合并时间：2026-07-21 12:30

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/49113

执行摘要

本 PR 修复了 Rust engine-core-client 中当 logprobs 载荷为零行或零列时 slice::chunks 因 chunk size 为零而 panic 的问题。变更在解码器中添加了两个守卫分支：零行时返回空 Logprobs，零列时返回结构化错误。新增两个单元测试覆盖这些异常路径。该修复对 Python runner 用户无影响，但增强了引擎核心协议边界的健壮性。

功能与动机

engine-core 协议可能携带零个评分位置的 logprobs 载荷，例如 [0,0] 或 [0,k+1] 张量形状。解码器直接将列数传入 slice::chunks，当块大小为零时会无条件 panic。虽然 Python runner 不会发送此类形状，但 engine-core 协议是公开边界，panic 是错误的失败模式。该问题在审查 openinfer 相关 PR 时被发现。

实现拆解

1. 在 WireLogprobs::decode 中添加前置守卫（rust/src/engine-core-client/src/protocol/logprobs.rs）：
   - 在形状校验后、chunks 循环前插入两个守卫。
   - 若 token_ids.rows == 0，直接返回空的 Logprobs。
   - 若 token_ids.cols == 0 且 rows > 0，通过 bail_ext_value_decode! 返回 ExtValueDecode 错误。
2. 新增单元测试（rust/src/engine-core-client/src/protocol/logprobs/tests.rs）：
   - decodes_zero_row_logprobs_as_empty：验证 [0,0] 和 [0,3] 形状均解码为空。
   - rejects_zero_column_logprobs_with_rows：验证 [2,0] 形状返回错误而非 panic。

rust/src/engine-core-client/src/protocol/logprobs.rs

核心修复逻辑所在：在 decode 方法中添加了零行和零列的前置检查，避免 panic。

```
// rust/src/engine-core-client/src/protocol/logprobs.rs

// 在形状校验完成后，chunks 循环之前插入以下守卫分支

// Empty position lists may be encoded as either [0, 0] or [0, k + 1].
// 零行载荷：直接返回空 positions，避免后续 `slice::chunks` 因 chunk size 为零而 panic
if token_ids.rows == 0 {
    return Ok(Logprobs {
```

```
            positions: Vec::new(),
        });
    }

    // 零列但非零行的载荷：不可能有意义，报出结构化错误而非 panic
    if token_ids.cols == 0 {
        bail_ext_value_decode!(
            "{field_prefix}: zero-column logprobs payload with {} rows",
            token_ids.rows
        );
    }
```

rust/src/engine-core-client/src/protocol/logprobs/tests.rs

新增了两个单元测试，覆盖零行和零列两种异常场景。

```
// rust/src/engine-core-client/src/protocol/logprobs/tests.rs

#[test]
// 验证零行 logprobs 形状 [0,0] 和 [0,3] 均解码为空，不会 panic
fn decodes_zero_row_logprobs_as_empty() {
    for shape in [[0usize, 0], [0, 3]] {
        let frames = vec![Bytes::from(encode_value(&output_wire_with_custom_fields(
            None,
            Some(Value::Array(vec![
                ndarray_value("<i8", &shape, Value::Ext(3, Vec::new())),
                ndarray_value("<f4", &shape, Value::Ext(3, Vec::new())),
                ndarray_value("<i8", &[0], Value::Ext(3, Vec::new())),
                Value::Nil,
            ])),
        )))];
        let decoded = decode_engine_core_outputs(&frames)
            .unwrap()
            .into_request_batch()
            .unwrap();
        let logprobs = decoded.outputs[0]
            .new_prompt_logprobs_tensors
            .clone()
            .unwrap()
            .into_direct()
            .unwrap();
        assert!(logprobs.is_empty());
    }
}

#[test]
// 验证零列但非零行的 payload 导致 ExtValueDecode 错误而非 panic
fn rejects_zero_column_logprobs_with_rows() {
    let ranks = Value::Ext(3, vec![1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]);
    let frames = vec![Bytes::from(encode_value(&output_wire_with_custom_fields(
```

```
            Some(Value::Array(vec![
                ndarray_value("<i8", &[2, 0], Value::Ext(3, Vec::new())),
                ndarray_value("<f4", &[2, 0], Value::Ext(3, Vec::new())),
                ndarray_value("<i8", &[2], ranks),
                Value::Nil,
            ])),
            None,
        ))];

        let error = decode_engine_core_outputs(&frames).unwrap_err();
        let crate::error::Error::ExtValueDecode { message } = &error else {
            panic!("expected ExtValueDecode");
        };
        assert_eq!(
            message,
            "new_logprobs: zero-column logprobs payload with 2 rows"
        );
    }
```

评论区精华

- BugenZhao 询问上下文是否来自 openinfer-project/openinfer#721，作者 FeathBow 澄清是在审查 openinfer-project/openinfer#722 时发现的，并非直接关联。
- BugenZhao 以 LGTM 批准。

风险与影响

- 风险：低。变更为新增前置返回分支，不改变现有逻辑路径；测试覆盖了新增的两种异常情况。但缺少集成测试验证与其他引擎核心的兼容性。
- 影响：直接影响自定义 engine-core 实现的用户；对使用原始 Python runner 的用户无影响。

关联脉络

该 PR 与 PR #48992（Rust Frontend 引擎感知健康报告）同属于 Rust engine-core-client 模块的改进，增强了协议层的健壮性。