

PR #49876 完整报告

vllm-project/vllm

[Bugfix][Parser] Confirm reasoning end when an Inkling content block opens

合并时间: 2026-08-09 06:07

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/49876>

执行摘要

- 一句话: 修复 Inkling 无思考块时文本后接工具调用解析失败
- 推荐动作: 值得精读: 变更虽小, 但清楚展示了共享 parser engine 中 phase gating 的设计约束, 以及 maintainer 如何通过 review 把 PR 收窄到单一 bug。关注 `inkling_config()` 的六条 transition 与 `test_visible_text_then_tool_streaming` 的断言方式。后续处理 #50528 或重构 `DelegatingParser` 时可复用此测试模式。

功能与动机

Issue #49865 报告: 使用 `--tool-call-parser inkling --reasoning-parser inkling --tokenizer-mode inkling` 且 `reasoning_effort=none/minimal` 时输出会带上 `<lend_message>`。PR body 复现了更精确的失败: 若响应以可见文本开头且无 thinking block, `DelegatingParser` 的 reasoning pass 从不发出 `REASONING_END`, 于是工具 pass 永远不执行, `<lcontent_invoke_tool_jsonl>{...}<lend_message>` 整段作为 content 返回且解析不出工具调用。PR body 明确: #51391 修复了 reasoning pass 内的 marker 归属, 但未修复这个 phase gating。

实现拆解

实现拆解如下:

1. 定位入口与根因: 入口是 `vllm/parser/inkling.py` 的 `inkling_config()` 状态机配置。`DelegatingParser` 先运行 reasoning pass, 再运行 tool pass; reasoning pass 只在收到 `EventType.REASONING_END` 时离开推理阶段。无 thinking block 的响应没有任何显式结束事件, 因此后续工具块全部留在 reasoning pass 内被当作普通文本输出。
2. 核心状态机修改: 为以下六条“开启可见内容块”的 transition 增加 `EventType.REASONING_END` 事件:
 - (CONTENT, TEXT_START)
 - (CONTENT, TOOL_TEXT)
 - (CONTENT, TOOL_ERROR)
 - (MESSAGE_HEADER, TEXT_START)
 - (MESSAGE_HEADER, TOOL_TEXT)
 - (MESSAGE_HEADER, TOOL_ERROR)

原因是这些 opener 都证明思考阶段已结束，显式确认边界后，后续 block-end 到达时 tool pass 才能接管并剥离工具标记。TOOL_START 不在此列，因为它是真正的工具调用终点，已有独立事件；这与 #51391 的 shared block-closer 处理不冲突。

1. 测试配套：在 tests/parser/engine/test_inkling.py 中扩展 _stream_delegating，使其额外按工具 index 聚合流式参数 fragments 并返回工具参数列表；新增 test_visible_text_then_tool_streaming，参数化 opener (TEXT_START / TOOL_TEXT / TOOL_ERROR) 与 chunk_size (1 / 3 / 64)，共 9 个用例；同时更新两个既有调用点以适配新的四元组返回值。
2. 范围收窄：按 bbrowning 的意见删除 passthrough_terminal_when_skipping_tools 及其 engine/config 管线、通用 _extract_tool_calls fallback；无工具纯文本的流式延迟问题拆到 follow-up；tool-first / no-thinking 场景留给 #50528。最终只保留 vllm/parser/inkling.py 与对应测试文件。

关键文件：

- vllm/parser/inkling.py (模块 解析引擎；类别 source；类型 core-logic；符号 inkling_config)：核心修复文件：在 inkling_config() 状态机中为六条可见内容块开启 transition 补发 REASONING_END 事件，使 DelegatingParser 能正确进入 tool pass。
- tests/parser/engine/test_inkling.py (模块 回归测试；类别 test；类型 test-coverage；符号 test_visible_text_then_tool_streaming, _stream_delegating)：新增回归测试：复现无 thinking block 时文本后接工具调用的流式泄漏，参数化三种 opener 与三种 chunk size，并扩展 _stream_delegating 以断言工具参数。

关键符号：inkling_config, test_visible_text_then_tool_streaming, _stream_delegating

关键源码片段

vllm/parser/inkling.py

核心修复文件：在 inkling_config() 状态机中为六条可见内容块开启 transition 补发 REASONING_END 事件，使 DelegatingParser 能正确进入 tool pass。

vllm/parser/inkling.py — inkling_config() 中可见内容块开启相关 transition 片段

```
terminals = {
    'MSG_MODEL': MESSAGE_MODEL,
    'TEXT_START': CONTENT_TEXT,
    'THINK_START': CONTENT_THINKING,
    'THINK_END': END_MESSAGE,
    'END_SAMPLING': CONTENT_MODEL_END_SAMPLING,
    'TOOL_START': CONTENT_INVOKE_TOOL_JSON,
    'TOOL_TEXT': CONTENT_INVOKE_TOOL_TEXT,
    'TOOL_ERROR': CONTENT_TOOL_ERROR,
}

transitions: dict[tuple[ParserState, str], Transition] = {
    # 从 CONTENT 态打开可见内容块：说明思考阶段必然已经结束。
    # 只有显式发出 `REASONING_END` 事件，`DelegatingParser` 才会在后续
    # block-end 到达时把块交给 tool pass 而不是当作普通文本输出。
    (ParserState.CONTENT, 'TEXT_START'): Transition(
```

```

        ParserState.CONTENT,
        (EventType.REASONING_END,),
    ),
    # 原始 / 错误工具块同样渲染为可见文本，与 TEXT_START 需要相同的边界确认；
    # 否则它们结束后 `END_MESSAGE` 会以文本形式泄漏进 content。
    (ParserState.CONTENT, 'TOOL_TEXT'): Transition(
        ParserState.CONTENT,
        (EventType.REASONING_END,),
    ),
    (ParserState.CONTENT, 'TOOL_ERROR'): Transition(
        ParserState.CONTENT,
        (EventType.REASONING_END,),
    ),
    # MESSAGE_HEADER 态下 (MSG_MODEL 之后的角色 / 函数名元数据段) 同样适用
    (ParserState.MESSAGE_HEADER, 'TEXT_START'): Transition(
        ParserState.CONTENT,
        (EventType.REASONING_END,),
    ),
    (ParserState.MESSAGE_HEADER, 'TOOL_TEXT'): Transition(
        ParserState.CONTENT,
        (EventType.REASONING_END,),
    ),
    (ParserState.MESSAGE_HEADER, 'TOOL_ERROR'): Transition(
        ParserState.CONTENT,
        (EventType.REASONING_END,),
    ),
}

```

tests/parser/engine/test_inkling.py

新增回归测试：复现无 thinking block 时文本后接工具调用的流式泄漏，参数化三种 opener 与三种 chunk size，并扩展 _stream_delegating 以断言工具参数。

```

# tests/parser/engine/test_inkling.py — 新增回归：无 thinking block 时
# 可见文本之后紧跟工具调用，流式 chunk 必须正确切到 tool pass
@pytest.mark.parametrize('opener', [TEXT_START, TOOL_TEXT, TOOL_ERROR])
@pytest.mark.parametrize('chunk_size', [1, 3, 64])
def test_visible_text_then_tool_streaming(
    self, mock_tokenizer, mock_request, chunk_size, opener
):
    """可见内容先于工具调用，且没有 thinking block。

```

旧行为：reasoning pass 始终停在 reasoning 阶段，整个工具块被当作 assistant 文本返回，工具调用完全丢失。三种 opener 都会开启可见内容块，所以必须全部覆盖。

```

"""
tools = [_function_tool()]
mock_request.tools = tools
content, _, names, args = _stream_delegating(
    _delegating(mock_tokenizer, tools),

```

```

mock_request,
f'{opener}let me check{END_MESSAGE}{MSG_MODEL}'
+ _tool_block('get_weather', '{"city": "SF"}'),
chunk_size,
self.GEN_PROMPT,
)
# 可见内容必须精确等于文本本身，工具块与结束标记不得泄漏
assert content == 'let me check'
assert names == ['get_weather']
assert json.loads(args[0]) == {"city": "SF"}
assert TOOL_JSON not in content
assert END_MESSAGE not in content

```

评论区精华

- bbrowning 在 CHANGES_REQUESTED 中要求: “I verified the following still fails on current main… Adding REASONING_END to the visible-content opener transitions fixes it.” 并要求 rebase、只保留六条 transition、新增 DelegatingParser.parse_delta 回归测试，同时删掉 passthrough 逻辑。作者照做后最终获得 approved。
- Codex 在 vllm/parser/inkling.py:199 指出 TOOL_TEXT / TOOL_ERROR 同样打开可见块，若只给 TEXT_START 发 REASONING_END，块结束时 skip_tool_parsing 会把结束标记重发为文本导致泄漏；最终实现覆盖全部六条转换。
- bbrowning 的 pre-commit 反馈: tests/parser/engine/test_inkling.py:173: error: Need type annotation for "args"，作者补充类型注解后通过。
- bbrowning 合入前验证: parser/reasoning/tool 全量单测 4073 passed，并复用 #51391 的 live model 复现脚本确认无回归。
- 早期多轮 Codex review 曾把改动扩展到 numeric reasoning_effort、effective tools、Responses API 等更广范围，最终按 bbrowning 要求全部拆出，只保留 streaming text→tool 这一个 bug。
- 要求 rebase 并收窄范围到 streaming text→tool (design): 作者删除 passthrough_terminal_when_skipping_tools、通用 _extract_tool_calls fallback，只保留六条 transition 与回归测试；bbrowning 最终 approved。
- TOOL_TEXT / TOOL_ERROR 遗漏 REASONING_END (correctness): 最终实现为六条可见块 opener 转换全部增加 REASONING_END，包括 MESSAGE_HEADER 变体。
- 回归测试设计 (testing): 9 个用例在 main 上全部失败、本 PR 全部通过；两个既有调用点同步适配新返回值。
- pre-commit mypy 类型注解 (style): 作者补充 args: dict[int, str] = {} 类型注解后通过。
- 合入前 live model 验证 (testing): 4073 passed, 17 warnings; live model 无回归，bbrowning 给出 approved。

风险与影响

- 风险:

1. 状态机语义风险：REASONING_END 是共享 ParserEngine 的事件，虽然改动只在一个 Inkling 专属配置函数中，但任何未覆盖的 transition 组合（如 REASONING 态下直接出现 TEXT_START）若存在，可能产生重复或缺失事件；当前测试覆盖三种 opener 与三种 chunk size，但未穷举所有状态组合。
2. 回归风险：_stream_delegating 返回值从三元组变为四元组，若其他调用点未同步更新会破坏测试；PR 内两个既有调用点已更新，parser engine 全套 3767/4073 通过。
3. 验证风险：PR 自身说明没有 live Inkling 模型端到端评测；bbrowning 合入前用 Inkling-Small-NVFP4 复现脚本做了 live 验证，但该验证不在 CI 中固化。
4. 上下游依赖：修复依赖 #51391 的共享 block-closer 语义；若后续重构 ParserEngine 或 DelegatingParser 的 phase 切换逻辑，这六条 transition 需要同步回归。

• 影响：

- 用户影响：开启 Inkling reasoning + tool 双解析器的流式请求（尤其 tool_choice=auto）不再把工具块泄漏进正文，能正确解析工具名和参数；同时避免在非 reasoning 模式看到 <lend_message>。
- 系统影响：仅 parser 配置层状态机事件变化，不影响推理、KV cache、调度等路径；无性能开销。
- 团队影响：为共享 parser engine 确立了“可见内容块 opener 需要显式确认 reasoning 边界”的约定；为 #50528 等后续 parser 修复提供了模式。影响范围为 Inkling 模型 + tool-calling 用户，规模中等。
- 风险标记：状态机核心路径变更，无 CI 固化 live 模型验证，依赖 #51391 语义，测试 helper 返回值签名变化

关联脉络

- PR #51391 [Bugfix][Parser] Prevent Inkling block-end leakage with tools: 本 PR 修复其未覆盖的 phase gating: rebase 到 main 后仍存在 streaming text→tool 泄漏；两 PR 共享 inkling_config() 与 DelegatingParser 语义，本 PR 明确在其基础上收窄。
- PR #50528 Tool-first / no-thinking case follow-up (referenced in PR body): PR body 说明 tool-first / no-thinking 场景留给 #50528，由其负责 TOOL_START → REASONING_END 转换，与本 PR 同属 Inkling parser 修复线。