PR #45553 完整报告

vllm-project/vllm

[Bugfix][Gemma4] Fix offline parser truncation, adjust_request token leak, and chat template sync

合并时间：2026-06-16 12:31

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/45553

执行摘要

- 一句话：修复 Gemma4 离线解析截断和 thinking 禁用时的 token 泄漏
- 推荐动作：建议阅读此 PR 了解 Gemma4 工具调用问题的修复模式，特别是 adjust_request 和 is_reasoning_end 的逻辑。强烈关注后续 PR #45832 以解决 adjust_request 回归。对于 Gemma4 用户，建议升级并监控工具调用行为。

功能与动机

修复用户报告的三个 Gemma4 工具调用 bug：Issue #39069 指出离线解析截断含引号字符串；Issue #39130 指出 --reasoning-parser gemma4 结合 enable_thinking=false 时结构化输出（xgrammar）被绕过；以及 adjust_request 在 thinking 禁用时导致 token 泄漏。PR 旨在统一修复已知问题并与上游模板保持一致。

实现拆解

1. 修复离线解析截断：在 vllm/tool_parsers/gemma4_utils.py 中，删除原有基于正则的 _parse_tool_arguments 实现，改为从 vllm.parser.gemma4 导入 _parse_gemma4_args 并委托给它，从而正确处理内部引号。
2. 修复 is_reasoning_end 当 thinking 禁用时的行为：在 vllm/parser/gemma4.py 中，Gemma4Parser.__init__ 从 chat_template_kwargs 读取 enable_thinking 存储为 _thinking_enabled。修改 is_reasoning_end：当遇到 <|turn> 或 <|tool_response> 时，返回 not self._thinking_enabled（索引为 thinking 禁用时这些 token 表示 reasoning 已结束）；并将最后的 fallthrough 从 self._reasoning_ended 改为 True，因为空 input_ids 且没有 reasoning token 即视为 reasoning 从未开始或已结束。
3. 修复 adjust_request 绕过父类当 thinking 禁用时：在 Gemma4Parser 中新增 adjust_request 方法，检查请求的 chat_template_kwargs 中 enable_thinking 是否为假，若为假则直接返回 request（保持默认 skip_special_tokens=True），避免父类强制设为 False 导致特殊 token 泄漏到内容中。
4. 聊天模板同步：更新 examples/tool_chat_template_gemma4.jinja：修复 format_argument 对 None 输出 null；默认 preserve_thinking 设为 false；添加 add_generation_prompt 守卫，确保在 tool 链中不重复添加 <|turn>model；优化 O(1) 的 continuation 检测；添加 enable_thinking=false 时在 assistant 首个消息前注入空的 thought channel 等。

5. 测试适配：修正测试中 arguments 字段格式（从字符串改为 dict）；修正 NO_REASONING 测试中 is_reasoning_end 预期从 False 改为 True 以匹配新逻辑。

关键文件：

- vllm/parser/gemma4.py（模块 解析器；类别 source；类型 core-logic；符号 init, adjust_request, is_reasoning_end）：核心解析器，包含 adjust_request 覆盖和 is_reasoning_end 修复，直接影响 thinking 禁用时的行为。
- vllm/tool_parsers/gemma4_utils.py（模块 工具解析；类别 source；类型 dependency-wiring；符号 _parse_tool_arguments）：离线工具调用解析器，修复含引号字符串截断问题，改为委托给原生 <|"|> 解析器。
- examples/tool_chat_template_gemma4.jinja（模块 聊天模板；类别 other；类型 configuration）：聊天模板与 HuggingFace 上游同步，修复 null 渲染、preserve_thinking 默认值、turn 边界等问题。
- tests/renderers/test_gemma4_chat_template.py（模块 模板测试；类别 test；类型 test-coverage）：更新测试用例中的参数格式以匹配新模板行为。
- tests/reasoning/test_gemma4_reasoning_parser.py（模块 推理测试；类别 test；类型 test-coverage）：修正 NO_REASONING 测试中 is_reasoning_end 预期值。

关键符号：adjust_request, is_reasoning_end, _parse_tool_arguments

评论区精华

Review 中核心讨论：

- Isotr0py 建议删除 gemma4_utils.py：由于 Transformers v5 已提供类似功能，作者同意但作为单独变更处理（创建 Issue #45605）。
- yzong-rh 报告 BFCL multi_turn_miss_func 下降 6.4%：提供 Before/After 数据，怀疑是模板变更引入的回归，有待进一步调查。
- yzong-rh 质疑 preserve_thinking 默认值变更：作者解释这是根据 Gemma4 官方文档建议，默认 false 以符合上下文管理最佳实践。
- bbrowning 指出 adjust_request 逻辑可能导致 thinking 禁用时解析完全失效：引擎需要特殊 token 才能工作，设置 skip_special_tokens=True 会破坏解析。该问题将在 #45832 中修复。
  - 删除 gemma4_utils.py 的可能性 (design): 作者同意，但作为单独变更，创建 Issue #45605 跟踪。
  - BFCL multi_turn_miss_func 指标下降 (performance): 评论表示可能不是本 PR 引起，但需进一步观察。
  - preserve_thinking 默认值变更是否故意 (design): 作者说明这是根据 Gemma4 文档建议更改，文档明确建议管理 thought 上下文。
  - adjust_request 逻辑导致 thinking 禁用时 parsing 失效 (correctness): 需要 #45832 修复，本 PR 的逻辑有误。

风险与影响

- 风险：
  1. adjust_request 回归（高）：bbrowning 指出当 thinking 禁用时，跳过父类会导致 skip_special_tokens=True，而引擎依赖特殊 token 进行解析，可能导致工具调用完全失效。已由 #45832 跟进修复。
  2. BFCL 多轮指标下降（中）：yzong-rh 报告 multi_turn_miss_func 下降 6.4%，虽然可能属于随机波动，但需持续监控。
  3. preserve_thinking 默认值变更兼容性（中）：用户可能依赖旧行为（保留思考内容），升级后需要显式设置 preserve_thinking=true。
  4. is_reasoning_end fallthrough 改为 True 的边缘情况（低）：某些边界场景（如窗口包含部分 reasoning token）可能误报结束，但测试覆盖了正常情况。 - 影响：主要影响使用 Gemma4 模型并启用 tool calling或 reasoning parser的用户。修复了关键 bug，恢复了 thinking 禁用时结构化输出的功能，并同步聊天模板到上游版本。需注意 preserve_thinking 默认改为 false，以及 adjust_request 逻辑中未解决的风险。总体影响范围中等，但属于必要修复。 - 风险标记：adjust_request 回归风险，BFCL 指标下降，preserve_thinking 默认变更兼容性

关联脉络

- PR #45588 [Frontend] Replace legacy Gemma4 parsers with engine-based implementation: 本 PR 基于 #45588 的重写框架，修复其未完全解决的遗留问题（离线解析截断、is_reasoning_end fallback、adjust_request token 泄漏）。