PR #42128 完整报告

vllm-project/vllm

[Bugfix] Fix Gemma4ToolParser streaming float corruption

合并时间：2026-05-14 09:03

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/42128

执行摘要

- 一句话：修复 Gemma4 流式浮点数损坏
- 推荐动作：值得精读，展示了流式 diff 场景下防御性保留的典型处理模式。设计上只改动了最必要的部分，避免了过度工程。

功能与动机

关联 Issue #42047 报告 Gemma4ToolParser 流式输出浮点数错误（如 108.2 变 108.02）。根因是浮点值末尾带 '.'（如 "108."）被过早解析为 108.0，造成流式 diff 前缀不一致。

实现拆解

1. 在 _parse_gemma4_args() 中增加 trailing-dot 保留检查（第 214-221 行）：在 bare value 分支中，当 partial=True 且原始值以 '.' 结尾时，直接 break 保留该值，避免输出不完整的浮点 JSON。
2. 在 _parse_gemma4_array() 中增加对称检查（第 305-308 行）：数组元素同样处理，防止数组中的浮点数损坏。
3. 性能优化（第二次提交）：将 raw_val = args_str[val_start:i].strip() 和条件检查移到 if partial: 块内，避免非流式路径产生额外开销。
4. 回归测试：在 tests/tool_parsers/test_gemma4_tool_parser.py 中新增 test_trailing_dot_float_partial_withheld 测试方法（dict 和 array 各一个），验证拖尾点号在 partial 模式被保留、非 partial 模式正常解析，以及多个参数中受影响值不会影响已稳定的前面参数。

关键文件：

- vllm/tool_parsers/gemma4_tool_parser.py（模块 工具解析器；类别 source；类型 core-logic；符号 _parse_gemma4_args, _parse_gemma4_array）：核心源码改动：在 _parse_gemma4_args 和 _parse_gemma4_array 中加入 trailing-dot 保留检查，防止流式解析输出错误浮点数。
- tests/tool_parsers/test_gemma4_tool_parser.py（模块 测试；类别 test；类型 test-coverage；符号 test_trailing_dot_float_partial_withheld）：新增回归测试，验证 trailing dot 保留行为正确，覆盖 dict 和 array 两种情况。

关键符号：_parse_gemma4_args, _parse_gemma4_array

关键源码片段

vllm/tool_parsers/gemma4_tool_parser.py

核心源码改动：在 _parse_gemma4_args 和 _parse_gemma4_array 中加入 trailing-dot 保留检查，防止流式解析输出错误浮点数。

```
# vllm/tool_parsers/gemma4_tool_parser.py (partial excerpt)

def _parse_gemma4_args(args_str: str, *, partial: bool = False) -> dict:
    # ... 其他解析逻辑 ...
    while i < n:
        # ... 键解析 ...
        # Bare value (number, boolean, etc.)
        else:
            val_start = i
            while i < n and args_str[i] not in (",", "}", "]"):
                i += 1
            if partial and i >= n:
                # 值可能不完整（如部分布尔值）—— 保留避免类型不稳定
                break
            if i == val_start:
                logger.warning(
                    "Gemma4 args parser made no progress at position %d; "
                    "aborting on malformed input.",
                    i,
                )
                break
            # 新增：partial 模式下，如果值以 '.' 结尾则保留
            # 避免 float("108.") → 108.0 产生的 JSON rep "108.0" 造成流式 diff 损坏
            if partial:
                raw_val = args_str[val_start:i].strip()
                if raw_val.endswith("."):
                    break
            result[key] = _parse_gemma4_value(args_str[val_start:i])
    return result

def _parse_gemma4_array(arr_str: str, *, partial: bool = False) -> list:
    # ... 类似逻辑 ...
    # Bare value
        else:
            val_start = i
            while i < n and arr_str[i] not in (",", "]"):
                i += 1
            if partial and i >= n:
                break
            if i == val_start:
                # ... 警告 ...
                break
            if partial:
```

```
                raw_val = arr_str[val_start:i].strip()
                if raw_val.endswith("."):
                    break
            items.append(_parse_gemma4_value(arr_str[val_start:i]))
    return items
```

tests/tool_parsers/test_gemma4_tool_parser.py

新增回归测试，验证 trailing dot 保留行为正确，覆盖 dict 和 array 两种情况。

```
# tests/tool_parsers/test_gemma4_tool_parser.py

def test_trailing_dot_float_partial_withheld(self):
    """Bare float ending with '.' is withheld in partial mode.

    Regression test for #42047: float("108.") → 108.0 causes
    streaming diff corruption (108.0 → 108.2 becomes 108.02).
    """
    # Single key with trailing dot — withheld entirely
    result = _parse_gemma4_args("left:108.,right:22.8", partial=True)
    assert result == {}

    # Stable key before trailing-dot key — stable key is kept
    result = _parse_gemma4_args(
        'name:<|"|>test<|"|>,score:3.,count:1', partial=True
    )
    assert result == {"name": "test"}

    # Non-partial mode parses trailing dot normally
    result = _parse_gemma4_args("left:108.,right:22.8", partial=False)
    assert result == {"left": 108.0, "right": 22.8}
```

评论区精华

gemini-code-assist[bot] 提出两点核心讨论：

- 性能优化：建议将 strip() 调用移到 if partial: 内部，避免非流式路径的额外开销（已被作者采纳并在第二 commit 修复）。
- 语义争议：检查仅在值已被分隔符终止时触发（i < n），理论上值已完整，但作者 abinggo 解释为“防御性做法”，因为 Gemma4 并不会将末尾带点的浮点作为完整值输出；核心的不完整保护仍是 i >= n 的 break。

bbrowning 批准时评价“改动很小，回归风险低，最坏情况只是流式保 - 持时间稍长，但远比输出错误浮点数好。”

- 性能：strip() 应在 partial 内部调用 (performance): 作者 abinggo 采纳建议并在第二次提交中修复。
- 语义：值已经被分隔符终止，是否还需要保留？ (correctness): 保持防御性保留，不影响功能。

风险与影响

- 风险：低风险。核心逻辑改动仅 12 行，在 partial 模式下增加了一个字符串前缀检查；非 partial 路径不受影响。最坏情况是流式保 - 持可能略慢，但对用户体验影响极小。测试覆盖了 dict 和 array 两种场景。
- 影响：
  - 用户影响：Gemma4 模型流式工具调用场景浮点数错误修复，提升准确性。
  - 系统影响：无额外性能开销（除非 partial 模式，但 strip 仅在 partial 模式下执行）。
  - 团队影响：小范围增量修复，便于后续更全面的 parser 改进。
  - 风险标记：边界条件防御性修改

关联脉络

- PR #42300 [Bugfix] Broader Gemma4ToolParser streaming fix (unstable bare values + refactor): 本 PR 最初被关闭以支持 #42300 的更广泛重构，后因 reviewer 倾向小增量修复重新开启。#42300 包含更全面不稳定值检查和 parser 重构。