PR #42879 完整报告

vllm-project/vllm

[Bugfix] Stream DeepSeek DSML tool-call argument deltas incrementally

合并时间：2026-05-28 17:50

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/42879

执行摘要

- 一句话：修复 DeepSeek DSML 工具调用参数非增量流式问题
- 推荐动作：建议精读，因为展示了如何从缓冲式流式解析迁移到增量状态机，对实现其他 tool parser 的增量流式有借鉴意义。同时 schema 兼容性处理方式（find_tool_properties 统一处理多种工具类型）值得关注。测试用例设计良好，覆盖了核心边界。

功能与动机

Issue #42878 报告 DeepSeek V4 DSML 工具调用流式并非真正的增量流式，而是缓冲整个 invoke 块，仅在闭合后才发出 arguments，导致客户端无法获得参数增量。期望行为是在 invoke 开始标签识别时立即发出工具元数据，并在参数内容到达时增量发射 arguments 片段。本 PR 实现了该行为，同时保留了先前 #41801 引入的 string=true/false 行为。引用 PR description: 'Fixes #42878'。

实现拆解

1. 状态扩展与正则新增：在 DeepSeekV32ToolParser.__init__ 中添加 _buffer, _in_tool_calls, _active_tool_index, _active_tool_name, _active_param_name, _active_param_string_attr, _active_param_mode, _active_param_parts, _args_started 等状态变量，并新增 invoke_start_regex 和 parameter_start_regex 用于识别标签开始。涉及文件：vllm/tool_parsers/deepseekv32_tool_parser.py。
2. 流式解析重写：将原来的 _extract_delta_tool_calls 方法替换为基于状态机的增量解析。新方法 _add_tool_call_delta, _begin_streaming_tool_call, _append_param_prefix, _append_json_param_value 负责在识别到标签开始 / 内容 / 结束时逐步构建 DeltaToolCall 并立即发射。当 invoke 开始标签被解析时，立即生成包含 id/type/name 的 delta；当参数内容逐片到达时，累积并增量发射 arguments 片段。
3. 参数配置与类型转换：新增 _get_param_config 方法，利用 find_tool_properties 统一支持 ChatCompletionToolsParam 和 Responses API FunctionTool 类型，确保 schema 兼容性。新增 _json_escape_string_content 正确处理 string=true 时的 JSON 转义。新增 _param_types_for_name 获取参数类型映射，以便在增量流式时进行类型转换。完善了 _convert_params_with_schema 的调用。
4. 状态重置与测试配套：扩展 _reset_streaming_state 以重置所有新增状态。测试文件 tests/tool_parsers/test_deepseekv32_tool_parser.py 新增 4 个测试用例，覆盖 Responses API FunctionTool schema 兼容性、流式与非流式转换回退一致性、未完整

invoke 不发射、以及 invoke 完成前即开始发射 arguments。tests/tool_parsers/test_deepseekv4_tool_parser.py 新增增量参数分块测试，确保每块参数内容碎片多于 2 次且拼接后结果正确。

关键文件：

- vllm/tool_parsers/deepseekv32_tool_parser.py（模块 解析器核心；类别 source；类型 core-logic；符号 _get_param_config, _json_escape_string_content, _extract_delta_tool_calls, _add_tool_call_delta）：核心变更，实现了 DSML 流式状态机，将完整 invoke 缓冲改为增量发射；改动量最大 (+313/-59)。
- tests/tool_parsers/test_deepseekv32_tool_parser.py（模块 V3.2 测试；类别 test；类型 test-coverage；符号 test_responses_function_tool_schema_in_streaming, test_streaming_matches_non_streaming_conversion_fallbacks, test_no_emission_while_incomplete, test_emits_arguments_before_invoke_completes）：新增 4 个测试用例，验证 Responses API FunctionTool schema 兼容性、非流式回退一致性、未完整 invoke 不发射、以及 invoke 完成前开始发射 arguments。
- tests/tool_parsers/test_deepseekv4_tool_parser.py（模块 V4 测试；类别 test；类型 test-coverage；符号 test_streaming_emits_incremental_argument_chunks）：新增 test_streaming_emits_incremental_argument_chunks，验证 V4 流式增量参数分块发射，确保参数碎片多于 2 块且拼接后结果正确。

关键符号：_get_param_config, _json_escape_string_content, _add_tool_call_delta, _begin_streaming_tool_call, _append_param_prefix, _append_json_param_value, _param_types_for_name, test_responses_function_tool_schema_in_streaming, test_streaming_matches_non_streaming_conversion_fallbacks, test_no_emission_while_incomplete, test_emits_arguments_before_invoke_completes, test_streaming_emits_incremental_argument_chunks

关键源码片段

vllm/tool_parsers/deepseekv32_tool_parser.py

核心变更，实现了 DSML 流式状态机，将完整 invoke 缓冲改为增量发射；改动量最大 (+313/-59)。

```
# 在 __init__ 中新增的流式状态变量
self._buffer: str = "" # 累积未处理的文本缓冲区
self._in_tool_calls: bool = False # 是否在 <｜DSML｜function_calls> 或 <｜DSML｜tool_calls> 内部
self._active_tool_index: int | None = None # 当前正在流式的工具索引
self._active_tool_name: str | None = None # 当前工具名称
self._active_param_name: str | None = None # 当前正在处理的参数名
self._active_param_string_attr: str | None = None # 当前参数的 string 属性 ("true"|"false")
self._active_param_mode: str | None = None # 参数模式：None / "prefix" / "value"
self._active_param_parts: list[str] = [] # 参数值的累积片段
self._args_started: list[bool] = [] # 每个工具是否已开始发射 arguments

# 新增正则用于识别标签开始
```

```
self.invoke_start_regex = re.compile(r'< | DSML | invoke\s+name="([^"]+)"\s*>')
self.parameter_start_regex = re.compile(
    r'< | DSML | parameter\s+name="([^"]+)"\s+string="(true|false)"\s*>'
)
```

评论区精华

- gemini-code-assist指出 _get_param_config 手动只处理 ChatCompletionToolsParam，会遗漏 FunctionTool schema，建议使用 find_tool_properties 工具函数。作者接受并修改，同时添加了回归测试 test_responses_function_tool_schema_in_streaming 验证 FunctionTool 的类型转换。
- chaunceyjiang（合并者）请求添加测试验证流式与非流式在 type 转换回退上一致，特别是 string=false 时的转换行为。作者添加了 test_streaming_matches_non_streaming_conversion_fallbacks 测试并通过。最终 chaunceyjiang 批准该 PR。
- _get_param_config 兼容 FunctionTool (correctness): 作者接受并修改为使用 find_tool_properties，同时添加了回归测试 test_responses_function_tool_schema_in_streaming 验证 FunctionTool schema 的类型转换。
- 增加流式与非流式回退一致性的测试 (testing): 作者添加了 test_streaming_matches_non_streaming_conversion_fallbacks 测试，覆盖多种类型转换情况，并通过。

风险与影响

- 风险：状态机新增状态变量较多，若 _reset_streaming_state 漏重置某个变量可能导致状态污染（已通过测试覆盖）。JSON arguments 片段拼接若中间状态不合法，但客户端应能容忍（符合 OpenAI 增量规范）。FunctionTool schema 兼容性已通过 find_tool_properties 修复，但若未来引入其他工具类型可能遗漏（当前 parser 仅支持这两种）。性能方面状态机每次 token 都需要正则匹配，但相比模型推理开销极小。非流式路径未经修改，但需确保不影响。
- 影响：影响 DeepSeek V3.2 和 V4 模型的流式工具调用功能。非流式路径未受影响，向前兼容。客户端将收到真正的增量 arguments，从假流式变为真流式，用户体验显著提升。所有工具 schema 转换逻辑与 #41801 保持一致。团队应关注流式工具调用的兼容性测试。
- 风险标记：状态机状态重置覆盖不全，新增正则匹配可能影响性能，FunctionTool 兼容性依赖工具函数修复

关联脉络

- PR #41801 Fix string=true/false behavior for DeepSeek tool call parameters: 本 PR 预设了 string=true/false 语义和 wrapper 修复，并在此基础上实现增量流式，依赖此前的参数类型转换逻辑。