

PR #43255 完整报告

vllm-project/vllm

[CI] Add composed-schema regression tests for DeepSeek V3.2/V4 parsers

合并时间: 2026-05-21 08:36

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/43255>

执行摘要

- 一句话: DeepSeek V3.2/V4 解析器 composed schema 测试
- 推荐动作: 建议阅读, 该 PR 展示了如何为工具调用解析器编写针对复杂 JSON Schema 类型的回归测试, 对理解 DSML 解析器的类型强制转换逻辑有参考价值。

功能与动机

如果解析器仅读取直接顶层的 `schema['type']`, 通过 `anyOf`、`oneOf`、`allOf` 等组合 schema 定义类型的参数可能回退为 `string` 类型, 导致客户端接收到 JSON 编码的字符串而非结构化对象。PR #43019 通过共享 `coerce_to_schema_type` 修复了该行为, 但缺少针对 composed schema 的回归测试。

实现拆解

1. V3.2 解析器测试: 在 `tests/tool_parsers/test_deepseekv32_tool_parser.py` 中新增三个测试方法:
 - `test_composed_schema_converts_object_and_array_params`: 验证 `anyOf` 对象参数转换为 dict, `oneOf` 数组参数转换为 list。
 - `test_string_attr_true_preserves_literal_for_composed_schema`: 验证 `string="true"` 时强制保留原始字符串。
 - `test_composed_schema_conversion_in_streaming`: 验证流式场景下 composed schema 仍能正确转换。
2. V4 解析器测试: 在 `tests/tool_parsers/test_deepseekv4_tool_parser.py` 中新增一个测试方法 `test_composed_schema_converts_object_and_array_params`, 验证 `anyOf` 对象和 `allOf` 数组参数的转换。
3. 所有测试均使用 `@pytest.mark.skip_global_cleanup` 标记, 避免全局自动清理对测试状态的影响。

关键文件:

- `tests/tool_parsers/test_deepseekv32_tool_parser.py` (模块 V3.2 解析器; 类别 test; 类型 test-coverage; 符号 `test_composed_schema_converts_object_and_array_params`, `test_string_attr_true_preserves_literal_for_composed_schema`, `test_composed_schema_conversion_in_streaming`): 新增三个 composed schema 回归测试, 覆盖 V3.2 解析器的主要场景 (非流式对象 / 数组转换、`string=true` 保留、流式转

换)。

- tests/tool_parsers/test_deepseekv4_tool_parser.py (模块 V4 解析器; 类别 test; 类型 test-coverage; 符号 test_composed_schema_converts_object_and_array_params) : 新增一个针对 V4 解析器的 composed schema 回归测试, 覆盖 anyOf 对象和 allOf 数组参数。

关键符号: test_composed_schema_converts_object_and_array_params,
test_string_attr_true_preserves_literal_for_composed_schema,
test_composed_schema_conversion_in_streaming

关键源码片段

tests/tool_parsers/test_deepseekv32_tool_parser.py

新增三个 composed schema 回归测试, 覆盖 V3.2 解析器的主要场景 (非流式对象 / 数组转换、string=true 保留、流式转换)。

```
# 测试包含 anyOf/oneOf 组合 schema 的参数能否正确转换为 dict 和 list
# 该测试位于 DeepSeekV32ToolParserTest 类中
@pytest.mark.skip_global_cleanup
def test_composed_schema_converts_object_and_array_params(self):
    # 构造工具定义: wait 属性使用 anyOf 中的两个对象 schema, patches 属性使用 oneOf (数组或 null)
    tool = ChatCompletionToolsParam(
        function=FunctionDefinition(
            name='set_timer',
            parameters={
                'type': 'object',
                'properties': {
                    'wait': {
                        'anyOf': [
                            {'type': 'object', 'properties': {'type': {'const': 'until'}, 'date': {'type': 'string'}}},
                            {'type': 'object', 'properties': {'type': {'const': 'for'}, 'minutes': {'type': 'number'}}},
                        ],
                    },
                },
                'patches': {
                    'oneOf': [
                        {'type': 'array', 'items': {'type': 'object'}},
                        {'type': 'null'},
                    ],
                },
            },
        ),
    )
    parser = make_parser(tools=[tool])
    # 构造 DSML 输入, string='false' 表示解析器应根据 schema 进行类型转换
```

```

model_output = (
    f'{FC_START}\n'
    f'{INV_START}set_timer">\n'
    f'{PARAM_START}wait" string=\'false\'>[{{\'type\':\'for\',\'minutes\':2880}}]{PARAM_END}\n'
    f'{PARAM_START}patches" string=\'false\'>[{{\'op\':\'replace\',\'path\':\'/schedule\',\'value\':\'quiet\'}}]{PARAM_END}\n'
    f'{INV_END}\n'
    f'{FC_END}'
)
result = parser.extract_tool_calls(model_output, None)
assert result.tools_called
args = json.loads(result.tool_calls[0].function.arguments)
# 验证对象参数解码为 dict，而非 JSON 字符串
assert isinstance(args['wait'], dict)
# 验证数组参数解码为 list，而非 JSON 字符串
assert isinstance(args['patches'], list)
assert args == {
    'wait': {'type': 'for', 'minutes': 2880},
    'patches': [{\'op\': 'replace', 'path': '/schedule', 'value': 'quiet'}],
}

```

评论区精华

该 PR 仅包含测试代码，没有实质性讨论。自动审查后由 sfeng33 快速批准合并。

- 暂无高价值评论线程

风险与影响

- 风险：风险极低，仅增加测试代码。但需注意：测试用例依赖具体的 schema 结构，若未来 shared coercion 逻辑修改可能需同步更新测试用例。另外 @pytest.mark.skip_global_cleanup 标记若与全局 cleanup 配置冲突可能导致测试跳过或失败。
- 影响：对用户透明，不涉及生产代码。对团队增加了针对 composed schema 的回归保障，防止后续重构破坏由 PR #43019 引入的修复。
- 风险标记：暂无

关联脉络

- PR #43019 [Bugfix] Use shared coerce_to_schema_type in DeepSeekV32 tool parser: 本 PR 为该修复添加回归测试，确保 composed schema 场景不被退化。