

PR #37015 完整报告

sgl-project/sglang

[Test] Add unit test for muse_glimmer_format

合并时间: 2026-09-13 18:40

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/37015>

执行摘要

- 一句话: 为 muse_glimmer_format 流式助手新增 31 个 CPU 单元测试
- 推荐动作: 值得快速浏览, 重点是 TestPartialMarkerLen 与 TestCouldStartHeader 的边界用例设计——它们清晰呈现了流式解析“暂缓输出 vs 立即输出”的判定规则, 对后续编写其他流式解析器的单元测试有直接借鉴价值。PR 本身技术含量不高, 无需精读完整实现。

功能与动机

PR body 声明: 为 #20865 增加 CPU 单元覆盖 muse_glimmer_format helpers (has_atem_markers、partial_marker_len、could_start_header) ——检测器已有测试, 但这三个流式辅助函数没有。这些函数在 Muse Glimmer 流式输出过程中决定是否暂缓输出文本, 误判会导致标记泄漏进 assistant 内容或吞掉真实文本, 属于需要回归保护的流式解析核心路径。

实现拆解

1. 新增测试文件 test/registered/unit/function_call/test_muse_glimmer_format.py, 从 sglang.srt.function_call.muse_glimmer_format 导入 EOM、EOT、MESSAGE、START、FUNCTION_CALLS_OPEN、INVOKE_OPEN、MAX_CHANNEL_MARKER、MAX_MARKER 等常量, 以及 has_atem_markers、partial_marker_len、could_start_header 三个被测函数。
2. 通过 register_cpu_ci(est_time=1, suite="base-a-test-cpu") 注册到 CPU CI 基线套件, 确保测试以秒级运行, 不启动服务器、不加载权重。
3. 按功能拆分为四个测试类: TestHasAtemMarkers 验证 ATEM 开标签检测以及关闭标签、相似标签不误报; TestPartialMarkerLen 验证流式缓冲尾部部分 marker 的 hold-back 长度, 覆盖空串、完整 marker、共享前缀 (EOM/EOT)、lookalike 后缀拒绝、max_len 边界; TestCouldStartHeader 验证 header 起始判断的接受与拒绝边界 (空串、to= 前缀、recipient 含空白、错误 marker 等); TestMarkerLengthInvariants 验证 MAX_CHANNEL_MARKER、MAX_MARKER 常量与最长的 framing 标记长度一致。
4. 本地 31 个用例全部通过; 提交历史包含 3 次 merge main 和一次 lint 修复, 维护者 ispobock 通过 /rerun-test 指令单独重跑该文件后 APPROVE。
5. 无配置、schema、部署等配套改动, 纯新增测试文件。

关键文件:

- test/registered/unit/function_call/test_muse_glimmer_format.py (模块 格式解析; 类别 test; 类型 test-coverage; 符号 TestHasAtemMarkers, TestPartialMarkerLen, TestCouldStartHeader, TestMarkerLengthInvariants) : 唯一的变更文件, 新增 4 个测试类共 31 个用例, 系统覆盖 muse_glimmer_format 流式辅助函数的正常、边界与反例路径, 并注册到 CPU CI 套件。

关键符号: has_atem_markers, partial_marker_len, could_start_header, test_invoke_open_is_a_marker, test_holds_back_the_longest_matching_suffix, test_shared_prefix_of_eom_and_eot_is_held, test_complete_header_is_accepted

关键源码片段

test/registered/unit/function_call/test_muse_glimmer_format.py

唯一的变更文件, 新增 4 个测试类共 31 个用例, 系统覆盖 muse_glimmer_format 流式辅助函数的正常、边界与反例路径, 并注册到 CPU CI 套件。

```
"""muse_glimmer_format 流式辅助函数的 CPU 单元测试。
```

```
这些函数在 Muse Glimmer 流尚未结束时决定: 立即输出文本, 还是暂时
按住 (hold back), 因为它可能是 channel header 或 ATEM marker 的前缀。
误判会让标记泄漏进 assistant 内容, 或吞掉真实文本直到下一个 chunk。
"""
```

```
from sglang.srt.function_call.muse_glimmer_format import (
    EOM, EOT, FUNCTION_CALLS_CLOSE, FUNCTION_CALLS_OPEN,
    INVOKE_CLOSE, INVOKE_OPEN, MAX_CHANNEL_MARKER, MAX_MARKER,
    MESSAGE, START, could_start_header, has_atem_markers, partial_marker_len,
)
```

```
from sglang.test.ci.ci_register import register_cpu_ci
from sglang.test.test_utils import CustomTestCase
```

```
# 注册到 CPU CI 基线套件: 不启动服务器、不加载模型权重
register_cpu_ci(est_time=1, suite="base-a-test-cpu")
```

```
# 会被按住暂缓输出的 marker 集合, 只有开标签才触发一次调用开始
_CHANNEL_MARKERS = (MESSAGE, EOM, EOT, START)
_HELD_BACK_MARKERS = (MESSAGE, EOM, EOT, START, FUNCTION_CALLS_OPEN, INVOKE_
OPEN)
```

```
class TestHasAtemMarkers(CustomTestCase):
    def test_invoke_open_is_a_marker(self):
        # <atem:invoke> 是函数调用开标签, 必须被识别为 marker
        self.assertTrue(has_atem_markers('<atem:invoke name="get_weather">'))

    def test_close_tags_alone_are_not_open_markers(self):
        # 关闭标签不触发调用, 只有开标签才启动一次调用
        self.assertFalse(has_atem_markers(INVOKE_CLOSE))
```

```

self.assertFalse(has_atem_markers(FUNCTION_CALLS_CLOSE))

def test_similar_atem_tag_is_not_an_open_marker(self):
    # <atem:parameter> 形似 ATEM 标签但并非开标签，不应误报
    self.assertFalse(
        has_atem_markers('<atem:parameter name="city">SF</atem:parameter>')
    )

class TestPartialMarkerLen(CustomTestCase):
    def test_complete_max_len_marker_is_not_held_back(self):
        # k 只遍历到 max_len-1，长度恰为 max_len 的完整 marker
        # 不可能再被当作“部分后缀”按住等待更多字节
        self.assertEqual(len(MESSAGE), MAX_CHANNEL_MARKER)
        self.assertEqual(
            partial_marker_len(MESSAGE, _CHANNEL_MARKERS, MAX_CHANNEL_MARKER),
            0,
        )

    def test_shared_prefix_of_eom_and_eot_is_held(self):
        # "<leo" 同时是 EOM 与 EOT 的前缀，必须按住等待后续字节
        text = "done<leo"
        held = partial_marker_len(text, _CHANNEL_MARKERS, MAX_CHANNEL_MARKER)
        self.assertEqual(held, len("<leo"))
        self.assertTrue(EOM.startswith(text[-held:]))
        self.assertTrue(EOT.startswith(text[-held:]))

    def test_lookalike_suffix_is_not_held(self):
        # "<leox" 无法长成任何已知 marker，继续等待只会吞文本，因此不按住
        self.assertEqual(
            partial_marker_len("done<leox", _CHANNEL_MARKERS, MAX_CHANNEL_MARKER),
            0,
        )

```

评论区精华

该 PR 没有代码 review 评论，维护者 ispobock 直接 APPROVE。主要交互是 CI 重跑指令：ispobock 在 PR 中执行 `/rerun-test test/registered/unit/function_call/test_muse_glimmer_format.py`，github-actions bot 回复 ">  ubuntu-latest (1 test):  [View workflow run]"，确认该测试文件在 ubuntu-latest 上单独运行通过。

- CI 重跑指定的测试文件 (other): 该测试文件在 ubuntu-latest 上通过，PR 随后由 ispobock APPROVE。

风险与影响

- 风险：
 1. 纯新增测试文件，无任何生产代码改动，运行时回归风险为零。

2. 测试与实现细节强耦合：多个用例直接断言常量关系（如 `len(MESSAGE) == MAX_CHANNEL_MARKER`、`MAX_MARKER` 覆盖 `FUNCTION_CALLS_OPEN`），未来若调整 `marker` 格式或常量定义，这些用例需要同步更新，但也因此能在第一时间暴露不变量被破坏。
3. 测试直接 `import` 源码内部常量，若被测函数签名变化会立即引起 `import` 错误，这实际是可接受的回归信号。
4. CI 影响：注册到 `base-a-test-cpu` 套件，`est_time=1` 秒，31 个纯逻辑用例对整体 CI 时长影响可忽略。
 - 影响：对用户和运行时无任何影响。对 CI 而言，CPU 基线套件新增一个约 1 秒的测试文件。对团队而言，为 `function_call/muse_glimmer` 流式解析提供了系统的回归保护，尤其是 ATEM 标记泄漏和真实文本被吞这两类高风险行为；同时为 #20865 的单元测试覆盖目标贡献了 `function_call` 方向的完整覆盖样例。
 - 风险标记：纯测试新增，实现细节强耦合，CI 新增 CPU 用例

关联脉络

- PR #39136 [Fix] Preserve GLM tool argument types across JSON Schema unions: 同属 `function_call` 检测器测试完善（新增 `unit/function_call` 测试），与本次为 `muse_glimmer_format` 补测试共同推进 #20865 的单测覆盖目标，且共享同一测试目录评审惯例。