

PR #33179 完整报告

sgl-project/sglang

[CI] Fix runtime context setup in flat logprob tests

合并时间: 2026-08-01 13:46

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/33179>

执行摘要

- 一句话: 修复 flat logprob 测试在 runtime context 迁移后的失败
- 推荐动作: 值得快速浏览, 重点看两处: 一是 TestSchedulerFlatAssembly 的 setUp/tearDown 如何使用 override_server_args() 的 scoped install/restore 模式对齐 namespace accessor 迁移, 这是配置迁移后测试写法的样板; 二是 SchedulerLogprobResultProcessor 死字段清理与调度器构造点的同步修改, 展示了跨 PR 栈合并顺序冲突的收尾手法。无需精读, 改动体量小且无核心逻辑变化。

功能与动机

PR body 明确指出: "Fixes the test failure exposed when #32223 landed after #33013 by publishing the runtime config expected by the migrated namespace accessor"。即 #33013 将配置读取从 `self.server_args.FIELD` 迁到 `get_exec()` / `get_memory()` 等 namespace accessor 之后, #32223 的单元测试仍以旧方式传入 `server_args`, 导致 TestSchedulerFlatAssembly 在 CI 上失败; 本 PR 的目标是让测试与迁移后的配置发布机制对齐。

实现拆解

按以下步骤完成修复:

1. 清理处理器死字段 (python/sglang/srt/managers/scheduler_components/logprob_result_processor.py): 删除 SchedulerLogprobResultProcessor 的 server_args: ServerArgs 字段, 并将 `from sglang.srt.server_args import (MIS_DELIMITER_TOKEN_ID, ServerArgs)` 收敛为 `from sglang.srt.server_args import MIS_DELIMITER_TOKEN_ID`。enable_mis 不再由构造参数注入, 而是由处理器内部通过 runtime context (`get_exec()` 等 accessor) 读取; `model_config` 字段保留。
2. 同步调度器构造点 (python/sglang/srt/managers/scheduler.py): 在 `init_batch_result_processor()` 中构造 SchedulerLogprobResultProcessor 时去掉 `server_args=self.server_args`, 只传 `model_config=self.model_config`, 与新的 dataclass 定义保持一致。
3. 修复测试 fixture (test/registered/unit/managers/test_flat_raw_top_logprobs.py): 新增 `from sglang.srt import runtime_context as rc; _make_logprob_processor()` 不再伪造 `server_args=SimpleNamespace(enable_mis=False)`; 为

TestSchedulerFlatAssembly 增加 setUp() / tearDown(), 通过 rc.get_context().override_server_args().install() / .restore() 以 scoped 方式发布 exec bag, 避免污染其他用例。

4. CI 验证: 通过 /rerun-test 重跑 test_flat_raw_top_logprobs.py、test_runtime_context.py、test_runtime_context_override.py、test_runtime_context_config_bags.py 四个套件, 全部通过。

关键文件:

- python/sglang/srt/managers/scheduler_components/logprob_result_processor.py (模块 结果处理; 类别 source; 类型 dependency-wiring) : 核心接线清理: 移除 server_args 字段, 使 enable_mis 改从 runtime context 的 namespace accessor 读取, 这是 #33013 迁移在 logprob 处理链路上的收尾。
- python/sglang/srt/managers/scheduler.py (模块 调度器; 类别 source; 类型 core-logic) : 调度器构造点同步: init_batch_result_processor() 中不再传 server_args, 与处理器新定义对齐, 是生产路径上唯一的调用方修正。
- test/registered/unit/managers/test_flat_raw_top_logprobs.py (模块 测试配套; 类别 test; 类型 test-coverage; 符号 setUp, tearDown) : 测试修复主体: TestSchedulerFlatAssembly 增加 setUp/tearDown 以 scoped override 发布 runtime config, _make_logprob_processor() 不再伪造 enable_mis, 是本次 CI 修复的关键配套。

关键符号: SchedulerLogprobResultProcessor, init_batch_result_processor, TestSchedulerFlatAssembly.setUp, TestSchedulerFlatAssembly.tearDown, _make_logprob_processor

关键源码片段

python/sglang/srt/managers/scheduler_components/logprob_result_processor.py

核心接线清理: 移除 server_args 字段, 使 enable_mis 改从 runtime context 的 namespace accessor 读取, 这是 #33013 迁移在 logprob 处理链路上的收尾。

```
from sglang.srt.runtime_context import get_exec # namespace accessor (#33013 迁移)
from sglang.srt.server_args import MIS_DELIMITER_TOKEN_ID
```

```
@dataclass(kw_only=True, slots=True, frozen=True)
```

```
class SchedulerLogprobResultProcessor:
```

```
    # 删除 server_args 字段: enable_mis 等配置不再由构造参数注入,
    # 而是通过 runtime context 的 accessor 读取 (与 #33013 保持一致),
    # 处理器内部仍可经 get_exec() 拿到已发布的配置 bag。
    model_config: ModelConfig
```

```
    def _process_input_token_logprobs(
        self, req: Req, input_token_logprobs: List
    ) -> None:
```

```
        """Process input token logprobs values and indices."""
```

```
# multi-item scoring 判定依赖 enable_mis, 该值在运行期由
# 调度器启动流程发布到 runtime context, 测试中则通过 setUp 安装。
is_multi_item_scoring = self._is_multi_item_scoring(req)
...
```

test/registered/unit/managers/test_flat_raw_top_logprobs.py

测试修复主体: `TestSchedulerFlatAssembly` 增加 `setUp/tearDown` 以 `scoped override` 发布 runtime config, `_make_logprob_processor()` 不再伪造 `enable_mis`, 是本次 CI 修复的关键配套。

```
from sglang.srt import runtime_context as rc
```

```
class TestSchedulerFlatAssembly(CustomTestCase):
    """Scheduler-side flat assembly in the logprob result processor."""

    def setUp(self):
        # enable_mis 已改从 runtime context 的 exec bag 读取 (#33013 迁移),
        # 因此测试前必须发布配置; install() 将 override 注入当前上下文。
        super().setUp()
        self._server_args_override = rc.get_context().override_server_args()
        self._server_args_override.install()

    def tearDown(self):
        # restore() 撤销本用例的发布, 避免配置泄漏到其他测试用例。
        self._server_args_override.restore()

    def _make_req(self, flat: bool, num_tokens: int = 5) -> Req:
        return Req(
            "r0",
            "",
            array("q", range(1, num_tokens + 1)),
            SamplingParams(),
            return_logprob=True,
            top_logprobs_num=2,
            return_flat_raw_top_logprobs=flat,
        )
```

评论区精华

该 PR 没有正式的 review 评论 (review_comments_count 为 0), 核心讨论隐含在两次提交和 CI 操作中:

- 第一个提交 (mmangkad) 先以 "直接发布 runtime config" 的方式修复测试; 第二个提交 (hnyls2002, 合入前) 改为 "use scoped context override; drop dead server_args field", 即用 `override_server_args()` 的 `install/restore` 作用域发布替换全局发布, 同时删除处理器中已无用的 `server_args` 字段。这体现了对测试隔离和配置生命周期一致性的取舍。
- 维护者通过 `/rerun-test` 指令重跑 4 个相关测试套件确认修复, CI 全绿。

- Issue 中出现一条 Gemini Code Assist bot 的停运说明 ("The consumer version of Gemini Code Assist on GitHub has been sunset")，与本次变更无实质关系。
- scoped override 替代全局发布 (design): 采用 scoped override 模式，测试间配置隔离更安全，同时清理死字段避免误导后续维护者。
- CI 重跑验证 (testing): 四个套件全部通过，CI 恢复绿色。

风险与影响

- 风险：风险整体较低，但仍有几个值得关注的点：
- 配置读取路径变更：SchedulerLogprobResultProcessor 内部的 enable_mis 读取从构造函数注入改为依赖 runtime context 发布。若某条路径在配置发布前调用处理器（如某些单测或边缘启动流程），get_exec() 可能读到未发布的 bag 而失败或走错分支。生产路径由 #33012/#33013 的 publish 流程保证，但其他构造点若仍传 server_args= 会直接因类型不匹配报错，需靠全量 CI 兜底。
- 测试状态泄漏：TestSchedulerFlatAssembly.setUp() 安装的 override 依赖 tearDown() 正确 restore；若用例中途异常且 tearDown 未被调用，可能污染同进程后续用例。当前实现是标准的 install/restore 模式，风险可控。
- 跨 PR 合并顺序耦合：本次失败是 #32223 与 #33013 合并顺序造成的，说明该功能线对 runtime-context 迁移存在时序依赖，后续类似栈需在合并前排查测试兼容性。
- 影响：影响范围很小：
- 用户与运行时：无任何行为变化。server_args 字段删除与调度器构造点调整均为内部接线清理，flat raw top logprobs 的 wire 格式和语义不变。
- 系统：仅影响调度器启动时 SchedulerLogprobResultProcessor 的实例化方式（少传一个参数），以及该处理器的配置来源（从 runtime context bag 读取）。
- 团队：恢复 CI 绿色，为后续 runtime-context 迁移收尾提供测试范式（scoped override 发布配置），减少同类测试在其他模块重蹈覆辙的概率。
- 风险标记：配置读取路径变更，跨 PR 合并顺序耦合，测试依赖 runtime context

关联脉络

- PR #32223 [perf] Assemble flat prompt top logprobs scheduler-side as numpy arrays: 本 PR 修复的目标对象：#32223 的测试在 #33013 迁移后暴露失败。
- PR #33013 config: read resolved config via namespace accessors: 配置读取迁移的源头 PR，导致 SchedulerLogprobResultProcessor 不再持有 server_args，本 PR 是其测试侧收尾。
- PR #33012 runtime_context: record the publishing process role: configuration 发布栈的基础 PR（#33013 的 base），定义了 runtime context 的进程角色发布机制，本 PR 依赖该机制。
- PR #33011 config: preserve resolved config across nested publishes + mutation ratchets: 同一 runtime-context 配置栈的前置 PR，保障了配置生命周期快照恢复，是测试中 scoped override 可用的前提。