PR #35446 完整报告

sgl-project/sglang

Fix HiCache PP sync test fixture

合并时间：2026-08-19 12:25

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/35446

执行摘要

- 一句话：修复 HiCache PP 同步测试夹具缺 buffer_pipeline 属性
- 推荐动作：值得速览而不必精读。它提供了一个典型样例：测试夹具绕过构造函数手工造对象时，会与源码新增属性之间形成“隐式契约”，源码一旦演进就会静默破裂。建议团队后续将 _make_cache() 改为基于真实 __init__ 参数构造，或引入字段清单式校验，减少这类逐字段修补的维护负担。

功能与动机

TestUnifiedPPSyncBatching._make_cache() 刻意绕过 UnifiedRadixCache.__init__() 改用 object.__new__() 手工搭建测试对象。加入 buffer-only HiCache 支持后，check_hicache_events() 新增了 buffer_pipeline 属性访问，而该夹具从未提供此字段，导致 AttributeError: 'UnifiedRadixCache' object has no attribute 'buffer_pipeline'，该问题在 PR #35306 的 base-a-test-cpu (2) 任务中暴露，需要恢复被破坏的单元测试。

实现拆解

变更入口是 test/registered/unit/mem_cache/test_hiradix_pp_sync_drain.py 中的 TestUnifiedPPSyncBatching._make_cache() 方法，整个 PR 只新增 1 行。实现过程拆解如下：

1. 定位根因：该夹具用 object.__new__(UnifiedRadixCache) 绕过 __init__ 手工构造被测对象，只补齐了 check_hicache_events() 当时需要的字段。PR #33473 引入 PP 批量写 / 读同步逻辑后，该方法依赖 _all_reduce、writing_check、loading_check 等字段完成跨 rank 计数聚合；PR #34798 追加 buffer-only 模式后，同一方法又新增读取 buffer_pipeline 属性，而夹具未同步更新，于是单测在 leader.check_hicache_events() 处抛 AttributeError。
2. 最小修复：在 _make_cache() 中、storage_metrics_collector 赋值之后补上 cache.buffer_pipeline = None，让手工构造对象与 check_hicache_events() 的读路径对齐。该测试不涉及 buffer-only 管道，置 None 即可满足当前取值语义，不引入额外 mock。
3. 验证：作者本地跑通 pre-commit hooks（Python AST、isort、ruff、codespell、CI 注册校验）以及 python3 -m py_compile 和 git diff --check；因 macOS 本地缺少 SGLang 运行时依赖，完整单测交由 PR CI 执行，随后通过 /rerun-test 在 ubuntu-latest 上确认 test_hiradix_pp_sync_drain.py 通过。

4. 配套情况：无配置、文档、schema 或部署改动；改动面严格限定在单测夹具，未触及生产代码。

关键文件：

- test/registered/unit/mem_cache/test_hiradix_pp_sync_drain.py（模块 缓存层；类别 test；类型 test-coverage；符号 TestUnifiedPPSyncBatching, _make_cache）：该文件是本次唯一变更文件。_make_cache() 用 object.__new__() 绕过 UnifiedRadixCache.__init__() 手工构造对象，在 PR #33473 和 PR #34798 合入后缺少 buffer_pipeline 字段，导致 check_hicache_events() 抛 AttributeError。补上一行 cache.buffer_pipeline = None 即可恢复 HiCache PP 同步单测。

关键符号：_make_cache

关键源码片段

test/registered/unit/mem_cache/test_hiradix_pp_sync_drain.py

该文件是本次唯一变更文件。_make_cache() 用 object.__new__() 绕过 UnifiedRadixCache.__init__() 手工构造对象，在 PR #33473 和 PR #34798 合入后缺少 buffer_pipeline 字段，导致 check_hicache_events() 抛 AttributeError。补上一行 cache.buffer_pipeline = None 即可恢复 HiCache PP 同步单测。

```
# 测试夹具：刻意用 `object.__new__` 绕过 `UnifiedRadixCache.__init__` 手工构造对象，
# 因此必须手动补齐 `check_hicache_events()` 会读取的所有字段，
# 否则该单测会因属性缺失直接抛 `AttributeError`。
def _make_cache(self, pp_rank, write_ready, load_ready):
    cache = object.__new__(UnifiedRadixCache)
    cache.tree_core = SimpleNamespace(enable_storage=False)
    cache.pp_rank = pp_rank
    cache.pp_size = 2
    cache.enable_storage_metrics = False
    cache.storage_metrics_collector = None
    # PR #34798 引入 buffer-only 模式后，`check_hicache_events()` 会读取 `buffer_pipeline`；
    # 此前 fixture 未提供该属性导致回归，这里显式置 `None` 恢复测试。
    # 注意：此测试不覆盖 buffer-only 真实管道，若未来读路径改为调用其方法，需同步更新。
    cache.buffer_pipeline = None
    cache._drain_async_work = MagicMock()
    cache._all_reduce = MagicMock()
    cache.writing_check = MagicMock()
    cache.loading_check = MagicMock()
    cache.drain_storage_control_queues = MagicMock()
    # 用 `ack_write_queue` 与 `ack_load_queue` 模拟各 PP rank 的完成事件，
    # `finish_event.query()` 的返回值由 `write_ready` / `load_ready` 参数决定，
    # 用于验证批量写 / 读计数只经一次 `_all_reduce` 聚合（见测试中的调用次数断言）。
    cache.cache_controller = SimpleNamespace(
        ack_write_queue=[
            SimpleNamespace(
                finish_event=SimpleNamespace(query=MagicMock(return_value=ready))
            )
```

```
            for ready in write_ready
        ],
        ack_load_queue=[
            SimpleNamespace(
                finish_event=SimpleNamespace(query=MagicMock(return_value=ready))
            )
            for ready in load_ready
        ],
    )
    return cache
```

评论区精华

该 PR 的 code review 评论数为 0，reviewer ispobock 直接 APPROVED 且未附评论。唯一的交互发生在 issue 评论：作者发布 /rerun-test test_hiradix_pp_sync_drain.py 触发单测复跑，github-actions bot 回报 ubuntu-latest 环境下 1 个测试通过（workflow run 32214675732）。讨论结论即该 fixture 修复在 CI 目标环境中验证有效，无需进一步改动。

- CI 复跑验证单测通过 (testing): ubuntu-latest 环境下 test_hiradix_pp_sync_drain.py 通过，验证 fixture 修复有效。

风险与影响

- 风险：风险集中在测试夹具的长期维护上：
- 同类回归风险：_make_cache() 仍采用 object.__new__() 绕过构造函数的模式，未来 check_hicache_events() 若再新增字段，该夹具仍可能再次失配；本次只补了 buffer_pipeline，没有引入“按源码字段自动补齐”的机制。
- 覆盖盲区：buffer_pipeline = None 意味着该单测不会覆盖 buffer-only 模式下的真实管道行为，check_hicache_events() 对 buffer_pipeline 非空路径（如事件统计、反向传播）没有测试保障。
- 验证环境限制：作者本地 macOS 未安装 SGLang 运行时，无法本机跑完整单测，验证依赖 CI；不过 rerun-test 已给出通过结果，当前风险已解除。
- 影响：对最终用户无任何影响（零生产代码变更）；对工程团队的影响是修复了 main 分支 CI 回归，让 PR #35306 之后一度失败的 test_hiradix_pp_sync_drain.py 重新可用，间接保障 HiCache PP 同步功能（批量写 / 读完成事件经一次 all_reduce 聚合）的回归防线。影响面小，但补上了跨 PR 演进造成的测试空窗。
- 风险标记：测试夹具脆弱性，object.__new__绕过初始化，覆盖盲区：buffer-only 模式，验证依赖 CI

关联脉络

- PR #33473 [HiCache] Batch PP write and load completion sync: 本测试正是覆盖该 PR 引入的 PP 批量写 / 读同步功能（check_hicache_events、_all_reduce 计数断言），该 PR 合入后 check_hicache_events() 依赖的字段变多，为本次 fixture 失配埋下伏笔。

- PR #34798 [HiCache] Buffer-only mode for HiCache host memory layer: 该 PR 为 check_hicache_events() 增加 buffer_pipeline 属性访问，是本次 AttributeError 的直接根因来源。