

PR #37339 完整报告

sgl-project/sglang

[Fix] Use real ReqKvInfo in unit-test req mocks

合并时间: 2026-09-01 11:25

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/37339>

执行摘要

- 一句话: 修复 7 个测试的 kv mock, 改用真实 ReqKvInfo
- 推荐动作: 值得快速浏览 (约 5 分钟): 它不是一个复杂 PR, 但体现了“测试 fixture 尽量使用真实数据结构而非鸭子类型 mock”的工程原则。对于后续要编写 mem_cache / KV 相关测试的开发者, 这个 PR 是推荐的 fixture 写法范例, 无需精读实现细节。

功能与动机

PR body 明确指出问题根因: Duck-typed kv=SimpleNamespace(...) mocks miss fields the KV record grew (e.g. the holds_kv property), 导致 test_hisparse_unit.py 在 main 上失败。修复方式为改用真实 ReqKvInfo() dataclass, 让 mock 与生产数据结构保持同步, 避免鸭子类型 mock 与真实字段集漂移。

实现拆解

1. 定位与统一策略: 核心是把测试 fixture 中的 kv 从手写 SimpleNamespace 鸭子类型 mock 换成真实 ReqKvInfo dataclass 实例, 使 mock 与生产数据结构天然同步, 未来任何字段变更都会在构造期暴露。
2. 逐文件替换: | 文件 | 改动 | | --- | --- | | test_hisparse_unit.py | _make_req 中 kv=SimpleNamespace(req_pool_idx=None, kv_allocated_len=0, kv_committed_len=0) 改为 kv=ReqKvInfo(), 用 dataclass 默认值覆盖原字段 | | test_fork.py | _make_group 与 orphan 回收测试中 leader 的 kv 改为 ReqKvInfo(kv_allocated_len=..., kv_committed_len=...) | | test_priority_scheduling_disaggregation.py | _new_decode_req 改为 ReqKvInfo(req_pool_idx=int(priority) % 8, cache_protected_len=0) | | test_disaggregation_wire.py | test_prebuilt_skips_unused_prompt_tensor 中改为 ReqKvInfo(req_pool_idx=0) | | test_lmcache_radix_cache.py | _make_req 与 req_to_token_pool.alloc 的 mock 均改为 ReqKvInfo | | test_radix_cache_unit.py | cache_unfinished_req 测试中 kv 改为 ReqKvInfo(...), 并顺带移除不再使用的 from types import SimpleNamespace 导入 | | test_paged_free_segment.py | SimpleNamespace(kv=ReqKvInfo(req_pool_idx=0)) |
3. 验证: 作者通过 /rerun-test 复跑了 6 个受影响用例 (test_hisparse_unit.py、test_radix_cache_unit.py、test_fork.py、test_disaggregation_wire.py、test_priority_scheduling_disaggregation.py、test_paged_free_segment.py), 在 1-gpu-5090 与 ubuntu-latest 两个 runner 上全部通过, 确认替换后各模块行为不变。

关键文件：

- test/registered/unit/managers/test_hispase_unit.py (模块 稀疏缓存；类别 test；类型 test-coverage；符号 _make_req)：PR body 明确点名此测试在 main 上失败，是本次修复的直接触发点；_make_req 的 kv mock 改为 ReqKvInfo() 默认实例。
- test/registered/unit/beam_search/test_fork.py (模块 束搜索；类别 test；类型 test-coverage；符号 _make_group, ReqKvInfo)：改动最典型的文件之一：_make_group 与孤儿回收测试中的 leader.kv 均替换为 ReqKvInfo，覆盖 beam search fork 的 KV 字段读写路径。
- test/registered/unit/managers/test_priority_scheduling_disaggregation.py (模块 优先调度；类别 test；类型 test-coverage；符号 _new_decode_req)：改动量最大的文件（+6/-2）：_new_decode_req 中 kv 改为 ReqKvInfo(req_pool_idx=int(priority) % 8, cache_protected_len=0)，覆盖优先调度与拆分解码队列。
- test/registered/xpu/test_lmcache_radix_cache.py (模块 远端缓存；类别 test；类型 test-coverage；符号 _make_req)：XPU 集成测试：_make_req 与 req_to_token_pool.alloc 的 mock 均换为 ReqKvInfo，保证 LMCache 远端缓存路径读到真实 KV record。
- test/registered/unit/mem_cache/test_radix_cache_unit.py (模块 前缀缓存；类别 test；类型 test-coverage；符号 ReqKvInfo)：radix cache 核心单测：cache_unfinished_req 测试的 kv mock 改为 ReqKvInfo，并顺带移除不再使用的 SimpleNamespace 导入，是清理最彻底的文件。
- test/registered/unit/disaggregation/test_disaggregation_wire.py (模块 分离部署；类别 test；类型 test-coverage；符号 ReqKvInfo)：disaggregation wire 测试：test_prebuilt_skips_unused_prompt_tensor 中 req.kv 改为 ReqKvInfo(req_pool_idx=0)，覆盖拆分发控路径。
- test/registered/unit/mem_cache/test_paged_free_segment.py (模块 分页释放；类别 test；类型 test-coverage；符号 ReqKvInfo)：paged 分页释放测试：mock Req 的 kv 改为 ReqKvInfo(req_pool_idx=0)，覆盖分页分配器的释放路径。

关键符号：_make_req, _make_group, _new_decode_req

关键源码片段

test/registered/unit/managers/test_hispase_unit.py

PR body 明确点名此测试在 main 上失败，是本次修复的直接触发点；_make_req 的 kv mock 改为 ReqKvInfo() 默认实例。

```
def _make_req(rid="test-req-0", origin_input_ids=None, output_ids=None):
    # 构造 HiSparseCoordinator 所需的最小 mock Req;
    # kv 必须用真实 ReqKvInfo 而不是手写 SimpleNamespace。
    if origin_input_ids is None:
        origin_input_ids = list(range(64))
    if output_ids is None:
        output_ids = []
    req = SimpleNamespace(
```

```

rid=rid,
origin_input_ids=origin_input_ids,
output_ids=output_ids,
fill_ids=origin_input_ids + output_ids,
seqlen=len(origin_input_ids) + len(output_ids),
# ReqKvInfo() 默认值覆盖原 mock 的三个字段 (req_pool_idx 为 None、
# kv_allocated_len / kv_committed_len 为 0)，且自动带上 holds_kv
# 等新增属性，避免鸭子类型 mock 在 main 上缺字段崩溃。
kv=ReqKvInfo(),
finished_reason=None,
hisparse_staging=False,
staging=False,
inflight_middle_chunks=0,
)
# finished 与 set_extend_range 仍按测试需要动态补充。
req.finished = lambda: req.finished_reason is not None
req.set_extend_range = lambda start, end: setattr(
    req, "extend_range", Range(start, end)
)
return req

```

test/registered/unit/beam_search/test_fork.py

改动最典型的文件之一：_make_group 与孤儿回收测试中的 leader.kv 均替换为 ReqKvInfo，覆盖 beam search fork 的 KV 字段读写路径。

```

class TestFreeMemberRows(CustomTestCase):
    def _make_group(self, req_to_token, allocated_len):
        # leader.kv 从 SimpleNamespace 换成 ReqKvInfo 后，
        # free_member_rows 对 kv_allocated_len / kv_committed_len 的读写
        # 走真实字段；断言 leader 回卷到 prompt 长度时与生产语义一致。
        leader = SimpleNamespace(
            kv=ReqKvInfo(
                kv_allocated_len=allocated_len, kv_committed_len=allocated_len
            ),
        )
        return SimpleNamespace(
            leader=leader,
            prompt_len=5,
            member_rows=torch.tensor([1, 2], dtype=torch.int64),
            member_rows_cpu=torch.tensor([1, 2], dtype=torch.int64),
            all_rows=torch.tensor([0, 1, 2], dtype=torch.int64),
        )

```

评论区精华

本 PR 无 review 评论 (review_comments_count=0)。唯一的讨论发生在 issue 侧：作者请求 /rerun-test 复跑受影响测试，github-actions[bot] 返回 1-gpu-5090 (2 个测试) 与 ubuntu-latest (4 个测试) 全部通过的结果，验证了替换 mock 后行为一致性。

- 暂无高价值评论线程

风险与影响

- 风险：风险极低，属于纯测试文件改动，生产代码零变更。需注意两点：一是 ReqKvInfo 构造器默认值必须与原来手写 mock 的字段值一致（如 req_pool_idx=None、kv_allocated_len=0），当前测试通过说明兼容；二是若 ReqKvInfo 未来增加必填位置参数，这 7 个测试文件需同步更新，但这正是本次改造的目的——让测试在构造期暴露数据结构演进，而非隐患。另外 test_radix_cache_unit.py 移除 SimpleNamespace 导入后，后续若复用该写法需重新导入。
- 影响：影响 7 个测试文件，覆盖 HiSparse 稀疏缓存、beam search fork、radix/ 前缀缓存、LMCache 远端缓存（XPU）、disaggregation 线格式、优先调度队列、paged 分页释放等模块。对用户与生产系统零影响；对团队而言统一了 KV mock 的构造方式，减少测试与生产数据结构漂移导致的 CI 断裂，是 mem_cache 系列重构的测试侧配套清理。
- 风险标记：纯测试改动，mock 与真实结构对齐，依赖 ReqKvInfo 构造器兼容性

关联脉络

- PR #37167 [mem_cache] Make release, row-reuse asserts, and presence checks read the KV record: 生产中统一以 KV record (ReqKvInfo) 为唯一事实来源，本 PR 是对应的测试侧清理，消除 mock 字段漂移。
- PR #32710 [Radix Cache] Add Rust TreeCore backend with shared parity tests: 同一 radix cache 测试模块 (test_radix_cache_unit.py) 的 KV 缓存结构演进背景，ReqKvInfo 字段持续变化。