

PR #34728 完整报告

sgl-project/sglang

[XPU][test] Add cache_salt=None to _make_req in test_lmcache_radix_cache.py

合并时间: 2026-08-14 10:27

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/34728>

执行摘要

- 一句话: XPU radix 缓存测试补 cache_salt 字段, 修复 CI 崩溃
- 推荐动作: 值得花 2 分钟快速阅读, 作为「API 迁移导致测试桩漂移」的最小修复范例:
PR body 的根因提交定位、波及面证据 (运行 ID 列表)、修复对齐方式三步论证很规范, 适合作为测试桩维护的参考样例; 但对推理引擎架构学习价值有限, 无需精读。建议顺手排查仓库内是否仍有其他未补 cache_salt 的 SimpleNamespace 测试桩。

功能与动机

30827 (feat: add cache salt support to KV cache events)
在 `python/sglang/srt/mem_cache/radix_cache.py:485` 的 `cache_finished_req` 中新增了 `cache_salt=req.cache_salt` 读取, 目的是让外部 KV-aware 路由器的缓存命名空间与 `extra_key` 解耦。但 XPU 测试 `test/registered/xpu/test_lmcache_radix_cache.py` 用 `SimpleNamespace` 手工构造假请求, 未同步补上该字段, 导致 `AttributeError: 'types.SimpleNamespace' object has no attribute 'cache_salt'` 沿 `lmcache_radix_cache.py:446` -> `radix_cache.py:485` 调用链抛出。PR body 明确指出:
#30827 当时修补了同构的 `test/registered/unit/mem_cache/test_swap_eviction_boundary.py` (本文件 docstring 也声明与之互为镜像), 却遗漏了 XPU 兄弟测试, 本 PR 是一次一行对齐修复。

实现拆解

1. 问题定位: 基于失败堆栈, 确认崩溃点在 `RadixCache.cache_finished_req` 读取 `req.cache_salt`, 而测试桩 `_make_req` (`test/registered/xpu/test_lmcache_radix_cache.py:64`) 的 `SimpleNamespace` 缺少该属性。

2. 一行修复：在 `_make_req` 的 `SimpleNamespace` 字面量中、`extra_key=None` 之后插入 `cache_salt=None`，以未加盐语义匹配真实请求默认值，并保持与 `test_swa_eviction_boundary.py` 的既有写法完全一致。
3. 验证与合入：本次改动不涉及生产代码、配置或 schema；Extra CI 运行 #31758802269 通过 ([:white_check_mark:])，Base 运行 #31697254447 失败，经 mingfeima 确认「failures unrelated」。第二个 commit 为 merge main，无实质变更。

关键文件：

- `test/registered/xpu/test_lmcache_radix_cache.py`（模块 缓存测试；类别 test；类型 test-coverage；符号 `_make_req`）：唯一变更文件：在 `_make_req` 的 `SimpleNamespace` 请求桩中补充 `cache_salt=None`，修复 #30827 引入 `cache_salt=req.cache_salt` 读取后 XPU CI 的 `AttributeError`，并与 `test_swa_eviction_boundary.py` 的既有修复对齐。

关键符号： `_make_req`

关键源码片段

`test/registered/xpu/test_lmcache_radix_cache.py`

唯一变更文件：在 `_make_req` 的 `SimpleNamespace` 请求桩中补充 `cache_salt=None`，修复 #30827 引入 `cache_salt=req.cache_salt` 读取后 XPU CI 的 `AttributeError`，并与 `test_swa_eviction_boundary.py` 的既有修复对齐。

```
def _make_req(rid, req_pool_idx, token_ids, tree):  
    """构造一个假的 Req 对象，字段集对齐 RadixCache / LMCRadixCache 实际读取的属性。
```

```
  
    这里用 SimpleNamespace 手工拼字段，避免依赖完整调度器；也因此存在  
    与真实 Req 字段漂移的风险——每次缓存路径新增读取属性时都要同步补桩。  
    本文件的桩与 test_swa_eviction_boundary.py 互为镜像，修复需保持一致。  
    """
```

```
    req = SimpleNamespace(  
        rid=rid,  
        req_pool_idx=req_pool_idx,  
        origin_input_ids=token_ids,  
        output_ids=[],  
        extra_key=None,  
        # 自 #30827 起 cache_finished_req 会直接读取 req.cache_salt 拼接  
        # radix key，缺失会触发 AttributeError；None 表示未加盐  
        cache_salt=None,  
        last_node=tree.root_node,  
        cache_protected_len=0,  
        priority=0,  
        kv_committed_freed=False,  
        kv_committed_len=len(token_ids),  
    )  
    # 缓存写入路径还会调用 pop_committed_kv_cache() 获取已提交长度  
    req.pop_committed_kv_cache = lambda: len(token_ids)  
    return req
```

评论区精华

仅有 1 条 Issue 评论：审核人 mingfeima 对 Base CI 失败给出结论「failures unrelated.」。结合 PR body 的 CI 插槽（Base 运行 #31697254447 失败、Extra 运行 #31758802269 通过），该评论澄清了失败并非本 PR 引入，而是 XPU 套件在 #30827 合并后的存量问题。无 review comment，审核直接 APPROVED；这也间接说明一行修复与仓库既有惯例（镜像 `test_swa_eviction_boundary.py` 的写法）一致，没有设计分歧或未解决疑虑。

- Base CI 运行失败的归因 (question): Base 运行失败属于存量问题（#30827 合并后 XPU 套件的既有污染），与本 PR 一行修改无关；改动本身经 Extra 运行验证通过，mingfeima APPROVED 并合并。

风险与影响

- 风险：改动本身零回归面（仅测试文件 +1 行）。真正值得警惕的是测试桩的「字段漂移」模式：test_swa_eviction_boundary.py 与本文件这套 SimpleNamespace 桩依赖手工维护与 Req 的属性对齐，后续缓存路径再新增字段时，仓库内其他用桩测试（不止 XPU 这一个）都可能再次被漏掉。本 PR 仅覆盖 XPU 注册套件，是否有其他测试文件仍缺 cache_salt 未在材料中确认，建议后续全局排查。
- 影响：修复直接恢复 XPU stage-b-test-1-gpu-xpu 套件，终结 #30827 合并以来 11+ 次无关 PR 的 XPU CI 失败噪音（PR body 列出了 11 个受影响运行 ID）；对 intel/xpu 开发者和所有跑该套件的 contributor 都有正向收益。对生产代码、模型推理、性能无任何影响。
- 风险标记：测试桩字段漂移，跨平台同步遗漏

关联脉络

- PR #30827 feat: add cache salt support to KV cache events: 根因来源：#30827 在 RadixCache.cache_finished_req 中新增 cache_salt=req.cache_salt 读取，本 PR 补齐其遗漏的 XPU 测试桩字段；#30827 当时只修补了同构的 test_swa_eviction_boundary.py，未覆盖 XPU 兄弟测试。