

PR #33975 完整报告

sgl-project/sglang

Move SWA chunk-cap hatch tests into the registered suite

合并时间: 2026-08-07 17:49

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/33975>

执行摘要

- 一句话: SWA chunk-cap hatch 测试迁入 CI 注册套件
- 推荐动作: 这是一个纯测试迁移 PR, 代码量小、逻辑清晰, 适合快速浏览。值得关注的点是: `_swa_req_never_fits` 门卫的语义 (什么条件下才允许走 `_swa_chunk_cap hatch`) 以及用真实 `PrefillAdder` fixture 替代手工拼接的测试改进方式。不建议深度精读, 但可将其作为 SWA 调度测试组织的参考。

功能与动机

PR body 明确说明: `TestSwaChunkCapHatch` 覆盖 `_swa_req_never_fits`, 这是让 `_swa_chunk_cap escape hatch` 仅对真正的队首活锁生效的门卫。该测试位于 `test/manual/test_schedule_policy.py`, 没有任何 workflow 会运行它, 因此该门卫目前没有 CI 覆盖, 后续对准入逻辑的改动可能静默放宽 hatch。迁移动机就是让这三个用例进入实际被 CI 执行的注册套件。

实现拆解

1. 删除旧测试载体: 从 `test/manual/test_schedule_policy.py` 移除 `_swa_adder` 帮助函数、`TestSwaChunkCapHatch` 类及其三个用例, 并清理不再使用的 `SimpleNamespace` 与 `PrefillAdder` 导入。
2. 新增 fixture 方法: 在 `test/registered/unit/managers/test_prefill_adder.py` 中新增 `create_swa_adder`, 通过既有 `create_token_allocator(size_swa=...)` 和 `create_adder(...)` 构建真实 `PrefillAdder`, 替代原先用 `__new__` 手工拼接字段的方式, 使测试运行在真实调度对象上。
3. 移植三个用例: 以 `test_swa_never_fits_*` 命名重写原用例, 分别验证瞬时压力下请求应等待 (`False`)、预算超过整个池时触发 hatch (`True`)、以及决策受池容量门控 (同一请求在不同 `size_swa` 下结果不同)。
4. 测试配额变化: `test_prefill_adder.py` 通过数 23 -> 26, `test_schedule_policy.py` 17 -> 14; 无新增配置或部署配套改动。

关键文件:

- `test/manual/test_schedule_policy.py` (模块 调度策略; 类别 test; 类型 test-coverage; 符号 `_swa_adder`, `TestSwaChunkCapHatch`, `test_transient_pressure_request_waits`, `test_request_larger_than_whole_pool_takes_hatch`): 删除了 `_swa_adder` 帮助函数、

TestSwaChunkCapHatch 类及三个用例，并清理 SimpleNamespace / PrefillAdder 导入，是测试迁移的出口文件。

- test/registered/unit/managers/test_prefill_adder.py (模块 预填调度; 类别 test; 类型 test-coverage; 符号 create_swa_adder, test_swa_never_fits_false_under_transient_pressure, test_swa_never_fits_true_when_budget_exceeds_whole_pool, test_swa_never_fits_is_gated_by_pool_capacity) : 新增 create_swa_adder fixture 与三个 _swa_req_never_fits 测试, 使 SWA chunk-cap hatch 门卫进入 CPU CI 覆盖, 是本次迁移的入口文件。

关键符号: create_swa_adder, test_swa_never_fits_false_under_transient_pressure, test_swa_never_fits_true_when_budget_exceeds_whole_pool, test_swa_never_fits_is_gated_by_pool_capacity, _swa_req_never_fits

关键源码片段

test/registered/unit/managers/test_prefill_adder.py

新增 create_swa_adder fixture 与三个 _swa_req_never_fits 测试, 使 SWA chunk-cap hatch 门卫进入 CPU CI 覆盖, 是本次迁移的入口文件。

```
def create_swa_adder(
    self, *, size_swa: int, sliding_window: int, page_size: int = 16
) -> PrefillAdder:
    # 关键改进: 不再用 __new__ 手工拼接字段, 而是复用既有 fixture 构造真实 PrefillAdder
    self.mock_tree_cache.sliding_window_size = sliding_window
    self.mock_token_allocator = self.create_token_allocator(size_swa=size_swa)
    return self.create_adder(
        self.create_running_batch(),
        page_size=page_size,
        rem_chunk_tokens=512,
    )

def test_swa_never_fits_false_under_transient_pressure(self):
    # 小请求 + 充足池: 预算 << 池容量, 等待解码排空即可容纳, 因此不得走 hatch
    adder = self.create_swa_adder(size_swa=1024, sliding_window=128)
    self.assertFalse(
        adder._swa_req_never_fits(extend_input_len=256, max_new_tokens=64)
    )

def test_swa_never_fits_true_when_budget_exceeds_whole_pool(self):
    # 大 host-hit 回载使预算超过整个池, 永远无法容纳 -> 触发 hatch
    adder = self.create_swa_adder(size_swa=1024, sliding_window=128)
    self.assertTrue(
        adder._swa_req_never_fits(
            extend_input_len=256, max_new_tokens=64, swa_host_hit_length=4096
        )
    )

def test_swa_never_fits_is_gated_by_pool_capacity(self):
```

```
# 同样请求只改池大小：证明比较的是预算与 size_swa，防止错误访问器回归
req = dict(extend_input_len=256, max_new_tokens=64, swa_host_hit_length=600)
self.assertTrue(
    self.create_swa_adder(size_swa=512, sliding_window=128)._swa_req_never_fits(
        **req
    )
)
self.assertFalse(
    self.create_swa_adder(
        size_swa=4096, sliding_window=128
    )._swa_req_never_fits(**req)
)
```

评论区精华

本 PR 无 review 评论或实质设计讨论。唯一的交互是作者发出 `/rerun-test test/registered/unit/managers/test_prefill_adder.py` 请求，GitHub Actions bot 返回运行成功。这属于 CI 重跑流程，不涉及技术争议。

- CI 重跑请求 (test): 测试通过，无进一步讨论。

风险与影响

- 风险：无生产代码变更，回归风险极低。
- fixture 语义差异：`create_swa_adder` 依赖 `test_prefill_adder.py` 既有 fixture 的默认值（如 `page_size=16`、`rem_chunk_tokens=512`），若这些 fixture 未来默认值变化，可能影响 `_swa_req_never_fits` 的测试有效性。
- 覆盖范围局限：测试注册在 CPU CI，覆盖的是纯 CPU 调度逻辑；SWA 相关 GPU 路径依赖其他测试补充。
- 误删除风险：从 `test/manual` 删除时若其他手动脚本依赖 `TestSwaChunkCapHatch`，可能造成引用失效；当前仓库内无此类引用迹象。
- 影响：对用户无任何行为影响（无生产代码改动）。对系统：补上了 `_swa_req_never_fits` 门卫的 CI 覆盖，防止后续修改准入逻辑时静默放宽 `_swa_chunk_cap escape hatch`，降低 SWA 重试风暴（retraction storm）回归风险。对团队：测试从 `manual` 目录迁入 `registered` 目录并纳入 CPU CI，符合测试分层治理方向，也为后续 SWA 调度改动提供了安全网。
- 风险标记：测试迁移，无生产代码变更，CI 覆盖补充

关联脉络

- PR #33475 [srt] Batch scheduler cache frees: 同属 SWA 调度与缓存释放优化，涉及 `swa.py` 与 `batch_result_processor.py`，与本 PR 测试的 `_swa_req_never_fits` 门卫同处 SWA 调度路径。
- PR #33794 Fix paged SWA retraction resume accounting: 修复 SWA retraction 恢复记账，与本 PR 测试防护的 SWA 重试风暴风险直接相关，两者共同维护 SWA 调度稳定性。