PR #24230 完整报告

sgl-project/sglang

[pd]: (Bug Fix) Incorrect out_cache_loc slicing in prepare_for_prebuilt

合并时间：2026-05-03 18:35

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/24230

执行摘要

- 一句话：修复 PD prefilling 中 out_cache_loc 切片偏移错误
- 推荐动作：是否值得精读：否。改动很小，但修复逻辑清晰，适合作为 radix cache 与 token pool 交互布局的快速了解案例。值得关注的设计决策：reviewer 建议的“在修复同时添加精度回归测试”是很好的实践，推荐团队在类似 bugfix 中推广。

功能与动机

PR body 明确指出：https://github.com/sgl-project/sglang/pull/19746 在 PD decode 端引入了 radix tree；在启用 radix cache 时，_pre_alloc() 将 req_to_token_pool 写入为 [prefix_locs(0..pre_len) | delta_locs(pre_len..seq_len)]。当存在前缀命中时（pre_len > 0），prepare_for_prebuilt() 从索引 0 切片（[:extend_input_len]）会读取 prefix 位置而非 delta 区域，导致 token 定位错误。

实现拆解

1. 核心修复：在 python/sglang/srt/disaggregation/decode_schedule_batch_mixin.py 的 prepare_for_prebuilt() 中，将 pre_len 的计算提前到切片之前，并将切片范围从 [:req.extend_input_len] 改为 [pre_len : pre_len + req.extend_input_len]，确保读取的是 delta 区域而非前缀区域。
2. 测试增强：在 test/registered/distributed/test_disaggregation_decode_radix_cache.py 中新增 test_gsm8k_accuracy_two_passes 测试，运行两轮 GSM8K 精度评估：第一轮无缓存，第二轮利用 radix cache，要求两轮精度均 > 0.80，且第二轮精度下降不超过 3%，以验证缓存不会显著降低输出质量。
3. 评审建议落地：新增测试正是响应 reviewer ShangmingCai 的建议“最好添加一个 GSM8K 精度测试”。

关键文件：

- python/sglang/srt/disaggregation/decode_schedule_batch_mixin.py（模块 调度器；类别 source；类型 core-logic；符号 prepare_for_prebuilt）：核心修复文件：修复 prepare_for_prebuilt() 中 out_cache_loc 切片错误，确保在 radix cache 场景下读取正确的 delta 区域。
- test/registered/distributed/test_disaggregation_decode_radix_cache.py（模块 测试；类别 test；类型 test-coverage；符号 test_gsm8k_accuracy_two_passes）：新增 GSM8K

精度回归测试 test_gsm8k_accuracy_two_passes，验证 radix cache 启用后不显著降低输出质量，响应 reviewer 建议。

关键符号：prepare_for_prebuilt, test_gsm8k_accuracy_two_passes

关键源码片段

python/sglang/srt/disaggregation/decode_schedule_batch_mixin.py

核心修复文件：修复 prepare_for_prebuilt() 中 out_cache_loc 切片错误，确保在 radix cache 场景下读取正确的 delta 区域。

```
# 修复前：从索引 0 开始切片，在 pre_len > 0 时会误读前缀区域
# chunk = self.req_to_token_pool.req_to_token[req.req_pool_idx][
# : req.extend_input_len
# ]

# 修复后：从 pre_len 偏移开始切片，正确读取 delta 区域
pre_len = len(req.prefix_indices) # 提前计算，供切片和后续逻辑复用
chunk = self.req_to_token_pool.req_to_token[req.req_pool_idx][
    pre_len : pre_len + req.extend_input_len # 切片偏移 pre_len
]
```

test/registered/distributed/test_disaggregation_decode_radix_cache.py

新增 GSM8K 精度回归测试 test_gsm8k_accuracy_two_passes，验证 radix cache 启用后不显著降低输出质量，响应 reviewer 建议。

```
def test_gsm8k_accuracy_two_passes(self):
    """Run GSM8K twice to verify decode radix cache does not degrade accuracy."""
    args = SimpleNamespace(
        base_url=self.base_url,
        model=self.model,
        eval_name="gsm8k",
        api="completion",
        max_tokens=512,
        num_examples=500, # 使用 500 个 GSM8K 样本进行评估
        num_threads=100, # 高并发以暴露潜在并发问题
        num_shots=6, # 6-shot prompting
    )

    metrics_first = run_eval(args) # 第一轮：通常无缓存命中
    print(f"First run metrics: {metrics_first}")

    metrics_second = run_eval(args) # 第二轮：期望利用 radix cache 加速
    print(f"Second run metrics: {metrics_second}")

    # 两轮精度都必须大于 80%，保证基本准确性
    self.assertGreater(metrics_first["score"], 0.80)
    self.assertGreater(metrics_second["score"], 0.80)

    # 第二轮精度不得比第一轮下降超过 3%
```

```
    accuracy_drop = metrics_first["score"] - metrics_second["score"]
    self.assertLessEqual(
        accuracy_drop,
        0.03,
        f"Second run accuracy dropped by {accuracy_drop:.4f} "
        f"(first={metrics_first['score']:.4f}, second={metrics_second['score']:.4f}), "
        f"exceeds 3% threshold",
    )
```

评论区精华

评审摘要：

- ShangmingCai 快速审批通过，并主动建议“It would be better to add a gsm8k accuracy test.”——该建议被采纳，在第二个 commit 中完成。
- ishandhanani 表示感谢发现并请教排查方法：“Thanks for the find. Can you explain how you found this error. Will help me in the future” —— 表明该 bug 定位难度较高，属于启发式发现的隐蔽问题。
- GSM8K 精度测试建议 (testing): 作者采纳建议，在第二个 commit 中新增了 test_gsm8k_accuracy_two_passes 测试。

风险与影响

- 风险：回归风险（低）：核心改动仅 2 行（切片范围表达式变更），逻辑正确。预计算 pre_len 后复用，不影响 seq_lens、cached_tokens 等下游计算。 精度风险（低）：新增 GSM8K 精度测试覆盖了缓存启用的场景，两层断言（绝对值 > 0.80、相对下降 <= 3%）能有效捕捉精度退化。 性能风险（低）：无影响。 兼容性风险（低）：仅在 --disaggregation-decode-enable-radix-cache 启用时生效，不影响默认行为。
- 影响：用户影响：修复了 PD 分离式架构中 decode radix cache 场景下的 token 定位错误。受影响的用户为开启 decode radix cache 的用户（通过参数 --disaggregation-decode-enable-radix-cache）。 系统影响：无明显影响。 团队影响：测试用例为后续 regresssion 提供了保障，减少同类 bug 逃逸的概率。
- 风险标记：核心路径变更，测试覆盖新增

关联脉络

- PR #19746 introduce radix tree on pd-decode side: 该 PR 在 PD decode 端引入了 radix tree，是当前 bug 的引入源头。本 PR 修复了该改动引入的切片偏移问题。
- PR #24257 [PD]: Support incremental transfer for mooncake transfer engine: 同为 PD 分离式架构的改进，涉及 KV cache 传输，与本 PR 在调度和缓存逻辑上有间接关联。