PR #51402 完整报告

vllm-project/vllm

[ROCm][CI][Bugfix] Do not microbatch a step that splits a prefix from its writer

合并时间：2026-08-08 02:55

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/51402

执行摘要

- 一句话：阻止 microbatch 拆开前缀写入者，修复 AMD CI 偶发失败
- 推荐动作：建议精读。这是一个以极小侵入修复微妙调度正确性问题的高质量样例：_allow_microbatching 的判断完全基于已有调度数据结构（num_computed_tokens_cpu 与 block table），不引入新契约，并通过“任一 rank 否决即全体不拆分”的语义保证了分布式一致性。值得关注的设计决策是只对“整块”做检查、decode 提前退出、以及不做厂商 gating。若后续要在 model runner v2 或新调度器上复刻同类保护，本函数是现成参考。

功能与动机

PR 目的明确：tests/v1/distributed/test_dbo.py::test_dbo_dp_ep_gsm8k[deepep_high_throughput] 在 AMD CI 上约 50% 概率失败，且并非测量噪声——服务器启动那一刻结果就定了，要么 GSM8K 约 0.65 要么约 0.60，失败时部分生成长度在 4-6 个 token 后截断，如同请求从未看到 few-shot 示例。根因是请求可以在调度阶段对另一请求当前正在计算的块形成前缀缓存命中，整体执行时 step 会先写完所有 KV cache 再读取，而 microbatching 把两个 half 串行化后，写入者在第二个 half 时第一个 half 的读取者会对仍为零的块做 attention。该问题不与后端或厂商绑定，Blackwell 上同样症状在 #29913 中被跟踪且测试标记为 xfail。

实现拆解

整体改动只涉及 vllm/v1/worker/gpu_model_runner.py 一个文件，新增 48 行、无删除，按如下 4 步实现：

1. 新增 _allow_microbatching 判定函数：通过 self.input_batch.num_computed_tokens_cpu 与 num_scheduled_tokens_np 获得各请求已计算 token 数与本轮计划 token 数。未启用 use_ubatching 或 num_reqs < 2 时直接返回 True；decode 型请求（query_lens 很小、computed > 0）因只读旧块也直接返回 True，避免热路径触碰 block table。
2. 块级冲突检测：遍历 self.input_batch.block_table.block_tables，把本轮从 computed // block_size 到 (computed + query_lens + block_size - 1) // block_size 覆盖的块收集为 written 集合，再检查每个 reader 已计算部分中的整块是否出现在 written 中；只查整块，因为 reader 结尾处的部分填充块是私有且自填自读的。命中任一冲突即返回 False。
3. 接入执行入口：execute_model 调用 _determine_batch_execution_and_padding 时传入 allow_microbatching=self._allow_microbatching(num_reqs, num_scheduled_tokens_np)，使危险 step 不再被 maybe_create_ubatch_slices 拆分。

任一 rank 否决即全体不拆分，因为各 rank 的 microbatching 决策是集体一致的。

4. 验证配套：没有新增测试文件；作者用 KV cache update op 审计与 8 次重启实验验证：写入块归属可精确预测成败，修复后 10 次运行全通过且无截断应答。

关键文件：

- vllm/v1/worker/gpu_model_runner.py（模块 模型运行器；类别 source；类型 core-logic；符号 _allow_microbatching）：唯一改动文件：新增 _allow_microbatching 判定函数并在 execute_model 调用 _determine_batch_execution_and_padding 时传入，作为 allow_microbatching 参数。修复同一 step 内前缀写入者与读取者被 micro-batch 拆开导致读取全零 KV 块的问题。

关键符号：_allow_microbatching

关键源码片段

vllm/v1/worker/gpu_model_runner.py

唯一改动文件：新增 _allow_microbatching 判定函数并在 execute_model 调用 _determine_batch_execution_and_padding 时传入，作为 allow_microbatching 参数。修复同一 step 内前缀写入者与读取者被 micro-batch 拆开导致读取全零 KV 块的问题。

```
# 拒绝将一个 step 拆成 microbatch，如果拆分会把"前缀写入者"与读取者分开。
# 场景：同一 batch 中，请求 A 正在计算共享前缀（如 few-shot 示例），
# 请求 B 在调度时就已经命中这个前缀的缓存块。整体执行时，step 会先完成
# 所有 KV cache 写入再做 attention 读取，因此命中有效；一旦拆分，
# 写入者可能落入后半段，前半段的读取者会对仍为全零的块做 attention。
def _allow_microbatching(
    self, num_reqs: int, num_scheduled_tokens_np: np.ndarray
) -> bool:
    # 未启用 microbatching，或 batch 中只有一个请求，无需干预。
    if not self.parallel_config.use_ubatching or num_reqs < 2:
        return True

    computed = self.input_batch.num_computed_tokens_cpu[:num_reqs]
    query_lens = num_scheduled_tokens_np[:num_reqs]

    # decode 步骤读取的块都来自之前 step 的写入，这里直接放行，
    # 避免在热路径上触碰 block table。
    readers = np.flatnonzero(
        (query_lens > (self.reorder_batch_threshold or 1)) & (computed > 0)
    )
    if readers.size == 0:
        return True

    # 统计本 step 会被写入的 KV 块：对每个请求，从当前计算位置到本轮
    # query 结束所覆盖的块区间，都属于"本轮写入范围"。
    for block_table in self.input_batch.block_table.block_tables:
        block_size = block_table.block_size
        table = block_table.get_numpy_array()
```

```
            start = computed // block_size
            stop = (computed + query_lens + block_size - 1) // block_size
            span = int((stop - start).max())
            columns = start[:, None] + np.arange(span)[None, :]
            written = np.unique(
                np.take_along_axis(
                    table[:num_reqs],
                    np.minimum(columns, table.shape[1] - 1),
                    axis=1,
                )[columns < stop[:, None]]
            )

            for reader in readers:
                # 只检查整块：如果 reader 只在本 step 结束时位于一个部分填充块
                # 内部，那个块是私有且由 reader 自己先填充再读取，不算风险。
                whole = computed[reader] // block_size
                if np.isin(table[reader, :whole], written).any():
                    # 命中了：reader 要读的某个整块，正被本 step 的其他请求写入。
                    return False

        return True

    # execute_model 中原先默认 allow_microbatching=True，
    # 现在改为先调用判定函数；任一 rank 返回 False 则所有 rank 都放弃拆分，
    # 因为各 rank 对 microbatching 的决策是集体一致的。
    (
        cudagraph_mode,
        batch_desc,
        should_ubatch,
        num_tokens_across_dp,
        cudagraph_stats,
    ) = self._determine_batch_execution_and_padding(
        num_tokens=num_tokens_unpadded,
        num_reqs=num_reqs,
        num_scheduled_tokens_np=num_scheduled_tokens_np,
        max_num_scheduled_tokens=max_num_scheduled_tokens,
        use_cascade_attn=cascade_attn_prefix_lens is not None,
        num_encoder_reqs=len(scheduler_output.scheduled_encoder_inputs),
        allow_microbatching=self._allow_microbatching(
            num_reqs, num_scheduled_tokens_np
        ),
    )
```

评论区精华

该 PR 没有实质的 review 争论：claude[bot] 因来自 fork 自动跳过，维护者 njhill 直接 approve，没有任何 inline review 评论。最有价值的讨论其实在 PR body 内部的证据链：作者通过审计 KV cache update op，说明在决定性 step 中第一个 half 读取 42 个块，全部是空的、且全部在其后变化；其中哪些块由第二个 half 写入，8 次启动预测结果无一例外。同时作

者澄清该问题不限于 AMD，Blackwell 上有同样症状，故检查不做厂商 gating。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. 性能回退：被判定为危险拆分的 step 将整体执行，可能提高单 step 的峰值显存与延迟；同时 _allow_microbatching 本身在每个计划的 step 都会执行（尽管 decode 路径提前返回），涉及 numpy 数组与 block table 扫描，属于热路径上的新增 CPU 开销。
  2. 正确性边界：np.take_along_axis 配合 np.minimum(columns, table.shape[1] - 1) 会把超出表长的列钳到最后一列，padding 块号可能混入 written 集合，导致保守地多拒绝一些 step——这偏向安全，但可能让一部分本可拆分的情况失去 micro-batch 收益。
  3. 测试缺口：没有新增直接单测或单元测试，回归保护依赖 CI 反复运行；若未来调度语义变化（如 reorder_batch_threshold 含义调整），该判定可能失效而不被发现。
  4. 平台覆盖：虽然作者主张问题与厂商无关，但实际验证只发生在 AMD CI 上；Blackwell 上的同类测试仍为 xfail，此修复对其有效性尚未被验证。 - 影响：影响范围为所有开启 microbatching（use_ubatching）的 vLLM v1 路径。对用户而言是纯 bug 修复：消除 prefix cache 命中共享前缀时可能读到全零 KV 块导致的精度下降与生成截断；只有触发条件的 step 会失去 micro-batch 拆分，通常只在首条 prefill 与长共享前缀（如 few-shot 示例）同时出现时发生，影响面小。对团队而言，AMD CI 上约 50% 概率失败的 flaky 测试被根治，后续性能与回归实验不再依赖运气；同时该修复为其他厂商（Blackwell）提供了同一根因的修复路径。 - 风险标记：核心路径变更，缺少专项测试，热路径新增 CPU 开销，禁用 micro-batch 可能引发性能回退

关联脉络

- PR #29913 Blackwell DBO prefix-split symptom (test marked xfail, referenced in PR body): PR body 明确提到同样的症状在 Blackwell 上被跟踪、对应测试标记为 xfail；本 PR 的检查未做 vendor gating，覆盖该路径。