PR #43982 完整报告

vllm-project/vllm

[Bugfix] Fix Gemma4 MTP block_table batch_size mismatch under concurrent load

合并时间：2026-06-04 08:11

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/43982

执行摘要

- 一句话：修复 Gemma4 MTP 并发下 block table batch_size 不匹配
- 推荐动作：建议尽快合并此修复，因为它直接解决了 Gemma4 MTP 在 FlashAttention 后端下的生产阻塞 bug。虽然改动极小，但 root cause 分析清晰，值得其他 speculative decoder 开发者在实现类似 per-group block table 时注意 batch 维度对齐。

功能与动机

Gemma4 + MTP + FlashAttention 在并发负载下（batch 部分占用时）会触发 RuntimeError: batch_size must be equal to batch_size_k，导致服务崩溃。该问题由作者在测试中发现并修复。

实现拆解

1. 定位根因：在 vllm/v1/spec_decode/gemma4.py 的 build_per_group_and_layer_attn_metadata 方法中，per-group block table 通过 set_per_group_block_table() 存储，其维度为 (num_reqs_padded, max_blocks)（CUDA graph 填充后的 padded 维度）。而 common_attn_metadata 在进入此方法前已通过 .unpadded() 裁剪到实际 num_reqs，导致二者 batch 维度不一致。
2. 修复单行代码：在 build_per_group_and_layer_attn_metadata 中，从 common_attn_metadata 获取 batch_size（实际请求数 num_reqs），在赋值 cm.block_table_tensor 时进行 self._per_group_block_tables[gid][:batch_size] 切片，使其与 cu_seqlens_q 维度对齐。
3. 测试验证：作者提供了详细的复现脚本和测试结果：在 8 卡环境、并发 8 请求、32 请求的负载下，修复前在 wave 1 就有 5/8 请求失败，随后服务崩溃；修复后所有请求成功。

关键文件：

- vllm/v1/spec_decode/gemma4.py（模块 推测解码；类别 source；类型 core-logic；符号 build_per_group_and_layer_attn_metadata）：本 PR 唯一修改文件，核心修复在 build_per_group_and_layer_attn_metadata 方法中增加 [:batch_size] 切片操作，解决 padded block table 维度不匹配问题。

关键符号：build_per_group_and_layer_attn_metadata

关键源码片段

vllm/v1/spec_decode/gemma4.py

本 PR 唯一修改文件，核心修复在 build_per_group_and_layer_attn_metadata 方法中增加 [:batch_size] 切片操作，解决 padded block table 维度不匹配问题。

```
# vllm/v1/spec_decode/gemma4.py

def build_per_group_and_layer_attn_metadata(
    self,
    common_attn_metadata: CommonAttentionMetadata,
    draft_index: int = 0,
) -> tuple[list[object], dict[str, object]]:
    """Build attention metadata using the correct block table per group.

    Gemma4 has multiple KV cache groups (sliding vs full attention)
    with different block tables.  The base class receives a single
    common_attn_metadata whose block_table belongs to one group.
    We swap in the correct block table for each draft attention group.
    """
    per_group_attn_metadata: list[object] = []
    per_layer_attn_metadata: dict[str, object] = {}
    batch_size = common_attn_metadata.batch_size() # 获取实际 batch 大小
    for attn_group in self.draft_attn_groups:
        gid = attn_group.kv_cache_group_id
        if gid in self._per_group_block_tables:
            cm = copy(common_attn_metadata)
            # 关键修复：切片到实际 batch 大小，对齐 cu_seqlens_q 维度
            # 原始 block table 可能带有 CUDA graph 填充的 padded 维度
            cm.block_table_tensor = self._per_group_block_tables[gid][:batch_size]
        else:
            cm = common_attn_metadata
        attn_metadata = attn_group.get_metadata_builder().build_for_drafting(
            common_attn_metadata=cm, draft_index=draft_index
        )
        per_group_attn_metadata.append(attn_metadata)
        for layer_name in attn_group.layer_names:
            per_layer_attn_metadata[layer_name] = attn_metadata
    return per_group_attn_metadata, per_layer_attn_metadata
```

评论区精华

reviewer benchislett批准了该PR，并询问为何该bug这么久才暴露出来。没有其他深层讨论。

- 暂无高价值评论线程

风险与影响

- 风险：风险极低：修改仅涉及一行代码（在 block_table_tensor 赋值时加 [:batch_size] 切片），且逻辑与 common_attn_metadata 的 padding 处理保持一致。不会影响其他模型或非 MTP 模式。注意需要确保 common_attn_metadata.batch_size() 返回的值是正确的实

际 batch 大小，当前逻辑中该值来自 unpadded 的 metadata，是可靠的。

- 影响：
  - 用户：修复了 Gemma4 MTP 在并发负载下的崩溃问题，使得该模型可以稳定用于生产。
  - 系统：仅影响 Gemma4 MTP 的 speculative decoding 路径，其他模型或解码方式无影响。
  - 团队：合并后无需额外配置或迁移。
  - 风险标记：并发路径变更，缺少测试覆盖

关联脉络

- PR #44253 [Bug Fix][Model Runner V2][Spec Decode] Warmup & capture with different attention states for speculator prefill: 同样关注 speculative decoding 下 CUDA graph 的 attention state 处理，与本 PR 的 batch 填充问题有相似的上下文。
- PR #44429 [Model] Add Gemma4 Unified (encoder-free) support: 本次修复针对 Gemma4 模型，该 PR 新增 Gemma4 Unified 支持，属于同一模型家族。