PR #34443 完整报告

sgl-project/sglang

[Bugfix][DSA] Fix num_splits "(b+1)" crash on prefill-CP speculative decode

合并时间：2026-08-12 10:13

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/34443

执行摘要

- 一句话：修复 DSA prefill-CP 推测解码 num_splits 崩溃
- 推荐动作：这是一个小而精的 bugfix，值得所有涉及 DSA 注意力、推测解码或 CP-v2 相关工作的工程师精读。核心看点是 cal_padded_tokens 与 prepare_mlp_sync_batch 的镜像关系，以及 enable_cp_v2() / is_cp_v2_active() 两个开关在不同前向模式下的组合语义。建议在后续 PR 中补充自动化回归测试，并把该对齐关系抽取为可复用的共享函数，避免再次漂移。

功能与动机

Fixes #30296。DSA 模型（GLM-5.x / DeepSeek-V3.2）在 --dsa-prefill-backend flashmla_kv --enable-prefill-cp --cp-strategy interleave 与 EAGLE/MTP 推测解码组合下，warmup 阶段崩溃：RuntimeError: num_splits must have shape (b+1)，这是 torch.ops.sgl_kernel.fwd_kvcache_mla 的 TORCH_CHECK 断言 num_splits.shape[0] == q.shape[0] + 1 失败。issue #30296 的复现矩阵显示，attn_tp_size == 1 或关闭推测解码时均正常，问题特定于 DSA attention TP 下推测路径的元数据不一致。

实现拆解

1. 根因定位：python/sglang/srt/layers/attention/dsa/utils.py 中的 cal_padded_tokens 通过 pad_dsa_cache_seqlens → _compute_flashmla_metadata 决定 FlashMLA num_splits 元数据长度；而 q 的长度由 ForwardBatch.prepare_mlp_sync_batch 决定，两者在 CP-v2 配置下出现分歧。
2. 策略分析：prepare_mlp_sync_batch 仅在 enable_cp_v2() 为 False 时应用 cp_align_size；而在推测前向 TARGET_VERIFY / DRAFT_EXTEND_V2 中，enable_cp_v2() 为 True 但 is_cp_v2_active(forward_batch) 为 False（这些模式不是 context_parallel_extend），因此 q 只按 attn_tp_size 填充，不会再按 attn_cp_size 对齐。
3. 修复实施：给 cal_padded_tokens 的 cp_align 循环增加 if not enable_cp_v2(): 守卫，并从 sglang.srt.layers.cp.utils 导入 enable_cp_v2；同时更新注释，明确说明与 prepare_mlp_sync_batch 的镜像关系以及该修复是 #30642 的 attn_cp 对应版。这样元数据长度与 q.shape[0] 完全一致，不再产生没有 q 对应的幻影填充行。
4. 验证方式：在 GLM-5-FP8（tp8 / ep8）上使用 --attention-backend dsa --dsa-prefill-backend flashmla_kv --enable-prefill-cp --cp-strategy interleave --moe-a2a-backend megamoe 加 EAGLE 推测解码，修复前 warmup 约 175s 确定性崩

溃，修复后可正常服务请求。

5. 测试配套：本 PR 未新增自动化测试文件，仅手工验证目标场景；建议后续补充覆盖 speculative + prefill-CP + DSA 的回归测试。

关键文件：

- python/sglang/srt/layers/attention/dsa/utils.py（模块 注意力层；类别 source；类型 core-logic；符号 cal_padded_tokens）：核心修改文件：cal_padded_tokens 的 cp_align 循环增加 enable_cp_v2() 守卫，修复 num_splits 元数据与 q 长度不一致导致的 (b+1) 断言崩溃。

关键符号：cal_padded_tokens

关键源码片段

python/sglang/srt/layers/attention/dsa/utils.py

核心修改文件：cal_padded_tokens 的 cp_align 循环增加 enable_cp_v2() 守卫，修复 num_splits 元数据与 q 长度不一致导致的 (b+1) 断言崩溃。

```
def cal_padded_tokens(forward_batch: "ForwardBatch"):
    # 与 `ForwardBatch.prepare_mlp_sync_batch` 的 `padding` 计算逻辑保持一致，
    # 计算 `attn_tp_size > 1` 或 `MAX_LEN` 模式下的实际 `token` 长度。
    from sglang.srt.layers.cp.padding import get_cp_padding_align_size
    from sglang.srt.layers.cp.utils import enable_cp_v2, is_cp_v2_active

    # CP-v2 已把每个 rank 的本地 shard 填充到物理大小，直接返回实际 token 数。
    if is_cp_v2_active(forward_batch):
        return forward_batch.attn_cp_metadata.per_rank_actual_token[
            get_parallel().attn_cp_rank
        ]

    global_num_tokens = forward_batch.global_num_tokens_cpu.copy()
    sync_group_size = len(global_num_tokens)
    attn_cp_size = get_parallel().attn_cp_size

    # 必须镜像 `ForwardBatch.prepare_mlp_sync_batch` 的填充策略：它只在 CP-v2 禁用时
    # 应用 `cp_align_size`。在推测前向（`TARGET_VERIFY` / `DRAFT_EXTEND_V2`）中，
    # `enable_cp_v2()` 为 True 但 `is_cp_v2_active()` 为 False，此时 `q` 仅按
    # `attn_tp_size` 填充；若这里无条件 `cp_align`，`num_splits` 会被填充得比 `q` 更长，
    # 触发 `fwd_kvcache_mla` 的 "num_splits must have shape (b+1)" 断言。
    # 该修复是 `PR #30642`（`attn_tp` 对齐）的 `attn_cp` 对应版本。
    if not enable_cp_v2():
        cp_align_size = get_cp_padding_align_size()
        for i in range(sync_group_size):
            global_num_tokens[i] = ceil_align(global_num_tokens[i], cp_align_size)

    # 复用 DP buffer 准备时选中的 padding 模式，避免重复计算 `dp_padding_mode`。
    dp_padding_mode = forward_batch.dp_padding_mode
    if dp_padding_mode is None:
```

```
        dp_padding_mode = DpPaddingMode.get_dp_padding_mode(
            forward_batch.is_extend_in_batch, global_num_tokens
        )
    if dp_padding_mode.is_max_len():
        tokens = max(global_num_tokens)
    elif len(global_num_tokens) > 1:
        tokens = global_num_tokens[get_parallel().attn_dp_rank]
    else:
        tokens = global_num_tokens[0]

    # prefill-CP round-robin 切分时，token 数需按 `attn_cp_size` 做 ceil 除法。
    if can_dsa_prefill_cp_round_robin_split(forward_batch):
        tokens = ceil_div(tokens, attn_cp_size)
    return tokens
```

评论区精华

该 PR 没有实质性的 review 评论，评审人 Fridge003 直接 APPROVE，唯一相关的交互是机器人触发的 /tag-and-rerun-ci CI 重跑命令。核心"讨论"体现在 PR body 中：作者明确指出 enable_cp_v2() 与 is_cp_v2_active(forward_batch) 在推测路径上的不同语义，并说明本次修复是 #30642（attn_tp 对齐）的 attn_cp 对应版；该分析解释了为什么修复只影响 TARGET_VERIFY / DRAFT_EXTEND_V2 两种前向模式，而 Legacy CP 与普通 EXTEND 路径保持不变。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. 覆盖范围窄：改动只影响 TARGET_VERIFY / DRAFT_EXTEND_V2 推测前向，Legacy CP 与普通 EXTEND 路径行为不变，回归风险较低。
  2. 潜在漂移风险：修复依赖 ForwardBatch.prepare_mlp_sync_batch 的填充策略，若该函数后续改变对齐规则，cal_padded_tokens 需要同步维护，否则可能重新出现 num_splits 与 q 的不一致。
  3. 测试缺口：没有新增自动化回归测试，enable_cp_v2() 与 is_cp_v2_active() 的组合语义未来可能被其他新前向模式破坏。
  4. 性能影响：基本无，修复只移除多余的元数据填充，不改变前向计算。风险等级：低 - 中。 - 影响：对用户：修复了 DSA 注意力 + --enable-prefill-cp --cp-strategy interleave + EAGLE/MTP 推测解码组合在 warmup 阶段确定性崩溃的问题，让 GLM-5.x / DeepSeek-V3.2 用户可以在该配置下正常使用推测解码和 prefill-CP。对系统：仅一个函数内的条件执行路径变化，无数值影响；对非 DSA 后端、非 speculative 场景无影响。对团队：这是 issue #30296 链条上继 #30642（attn_tp）之后的第二个修复，提示 DSA 元数据编排与 MLP 同步填充之间需要系统性的一致化测试。 - 风险标记：缺少自动化测试覆盖，核心注意力路径变更，依赖 prepare_mlp_sync_batch 对齐策略

关联脉络

- PR #30642 [Bugfix][DSA] Fix num_splits (b+1) crash with DP-attention + speculative decode: 同一 issue #30296 的 attn_tp 对齐修复；本 PR 是其 attn_cp 对应版，两者共同解决 DSA + speculative 下 num_splits 元数据与 q 不一致的问题。