PR #23757 完整报告

sgl-project/sglang

[Intel GPU] Fix incorrect KV-cache page table for local attention when page_size > 1

合并时间：2026-05-26 11:02

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/23757

执行摘要

- 一句话：修复 XPU attention page_size > 1 时 KV 缓存页表错误
- 推荐动作：值得精读以理解 page table 粒度转换的重要性，以及局部 attention 的正确前提。关注 reviewer 提出的代码复用建议，可作为后续重构候选。

功能与动机

修复 XPU 后端中 page_size > 1 且启用局部 attention 时 KV 缓存页表传递错误的问题。根因是 make_local_attention_virtual_batches 期望页粒度 block table，但实际传递了 token 粒度的 req_to_token，导致物理页索引错乱，flash_attn_with_kvcache 输出为零或错误。

实现拆解

1. 定位问题函数：在 python/sglang/srt/layers/attention/xpu_backend.py 的 _init_local_attn_metadata 方法中，make_local_attention_virtual_batches 调用前，page_table 变量持有 token 粒度的 req_to_token。
2. 添加页粒度转换：当 self.page_size > 1 时，用 torch.arange(0, page_table.shape[1], self.page_size) 生成步进索引，从 page_table 中每隔 page_size 列取一列，并整数除以 page_size 得到物理页索引，结果覆盖原 page_table。
3. 保留后续逻辑：转换后，page_table 传递给 make_local_attention_virtual_batches，其余计算保持不变，最大化复用现有代码。
4. 无测试配套：本次提交未包含单元测试，但 reviewer 建议后续补充类似 test_triton_attention_backend.py 的测试。

关键文件：

- python/sglang/srt/layers/attention/xpu_backend.py（模块 Attention 后端；类别 source；类型 core-logic）：核心修复文件，在 _init_local_attn_metadata 中添加页粒度转换逻辑，是唯一变更文件。

关键符号：_init_local_attn_metadata

关键源码片段

python/sglang/srt/layers/attention/xpu_backend.py

核心修复文件，在 _init_local_attn_metadata 中添加页粒度转换逻辑，是唯一变更文件。

```
# make_local_attention_virtual_batches expects a page-granularity block table:
# column p is the logical page number, and the value stored at that column is the
# physical page index. The raw req_to_token table is token-granularity (column i =
# the KV slot for token i), so when page_size > 1 we must stride and divide first
# so that block_starts = k_seqstarts_absolute // page_size correctly indexes the table.
if self.page_size > 1:
    strided_indices = torch.arange(
        0, page_table.shape[1], self.page_size, device=page_table.device
    )
    page_table = page_table[:, strided_indices] // self.page_size
```

评论区精华

代码复用讨论：机器人 reviewer 指出页表归一化逻辑在 init_forward_metadata 中已有相同实现（第 371-376 行），建议提取为共享辅助方法或调整执行顺序以避免重复。作者未直接回复，但 PR 最终合入。 测试补充建议：reviewer mingfeima 建议作者参考现有 attention 后端测试文件（如 test_triton_attention_backend.py）添加单元测试，作者承诺后续添加。

- 页表归一化逻辑的代码复用 (design): 作者未回应且未修改，PR 已合并，后续需关注代码冗余。
- 补充单元测试 (testing): 承诺添加，但当前 PR 未包含测试。

风险与影响

- 风险：回归风险低：变更仅影响 page_size>1 且启用局部 attention 的 XPU 路径，其他路径不受影响。无测试覆盖：当前 XPU 局部 attention 测试缺失，回归风险存在但可通过后续测试降低。性能影响可忽略：新增少量张量操作（stride+divide），计算开销极低。
- 影响：影响范围：仅 Intel GPU（XPU）后端，且需满足 page_size>1 且启用局部 attention。影响程度：功能性 Bug 修复，准确率从接近零恢复至正常水平（GSM8K 0.005→0.815），对相关场景至关重要。团队影响：修复由 Intel 团队贡献，需留意后续统一回到其他后端的共同逻辑。
- 风险标记：核心路径变更，缺少测试覆盖，代码重复

关联脉络

- PR #26313 Fix stale forward_metadata leak in DP attn unpadded idle batch: 同为 KV 缓存 metadata 相关 Bug 修复，涉及 page_table 和 attention 元数据正确性。