# PR #25556 完整报告

sgl-project/sglang

[AMD] Fix correctness for AITER MLA backend with `--page-size > 1`

合并时间：2026-06-03 14:01

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25556

## 执行摘要

- 一句话：修复 AITER MLA 后端 page_size>1 时的正确性
- 推荐动作：值得精读。该 PR 展示了如何通过精确理解数据流（allocator -> metadata -> kernel）来修复仅修改元数据就能解决看似复杂的分页问题。PR 描述中关于数据流链的分析具有较高学习价值。

## 功能与动机

修复 AMD 平台上 AITER MLA 后端中 page_size > 1 导致 GSM8K 精度几乎降为零的问题。修复后 page_size=64 的精度从 0.005 恢复到 0.970，同时预填性能提升高达 21%（TP8 并发 64 时）。

## 实现拆解

1. 修正 make_mla_prefill_ps_meta_data 中的 block_size 和 kvlen_granularity（python/sglang/srt/layers/attention/aiter_backend.py 第 491-492 行）：将 kvlen_granularity = max(128, self.page_size) 改为固定值 128；将 block_size = self.page_size 改为固定值 1。因为预填阶段接收的是线性 per-token K/V 张量（mla_prefill_ps_asm_fwd 不读取分页缓存），旧代码导致内核计算错误的工作拆分和步长。
2. 修正 init_cuda_graph_state 中 kv_indices 缓冲区大小：将非统一注意力路径的缓冲区分配从 block 粒度（max_bs * ceil(max_context_len / page_size)）改为 token 粒度（max_bs * max_context_len），避免 create_flashinfer_kv_indices_triton 写入时发生静默越界。
3. 无测试文件变更：PR 仅修改了 1 个文件共 2 行，改动极小。

关键文件：

- python/sglang/srt/layers/attention/aiter_backend.py（模块 注意力后端；类别 source；类型 core-logic；符号 make_mla_prefill_ps_meta_data, init_cuda_graph_state）：核心文件，包含两处关键的元数据修复：make_mla_prefill_ps_meta_data 中固定 block_size=1 和 kvlen_granularity=128；init_cuda_graph_state 中调整 kv_indices 缓冲区大小。

关键符号：make_mla_prefill_ps_meta_data, init_cuda_graph_state

## 关键源码片段

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

核心文件，包含两处关键的元数据修复：make_mla_prefill_ps_meta_data 中固定 block_size=1 和 kvlen_granularity=128；init_cuda_graph_state 中调整 kv_indices 缓冲区大小。

```
defmake_mla_prefill_ps_meta_data(self, ...):     # ...     kvlen_granularity = 128 # 固定为 128，不再依赖 self.page_size     block_size = 1 # 固定为 1，因为预填阶段使用线性 per-token 布局     # ... 剩余代码不变     get_ps_metadata_v1(..., kvlen_granularity=kvlen_granularity, block_size=block_size, ...) # init_cuda_graph_state 中的改动（简化） if not self.use_triton_unified_attention:     # 使用 token 粒度缓冲区，避免越界     kv_indices_buffer_size = max_bs * max_context_len else: max_num_blocks_per_seq = (self.max_context_len + self.page_size - 1) // self.page_size     kv_indices_buffer_size = max_bs * max_num_blocks_per_seq
```

## 评论区精华

AI 审查机器人 (gemini-code-assist[bot]) 指出 init_cuda_graph_state 中的条件也应包含 self.use_mla，因为即使启用了统一注意力，MLA 内核始终以 token 粒度运行；但该评论未在最终提交中得到处理。HaiShaw 已批准该 PR。

- AI 审查建议将 use_mla 加入条件 (correctness): 当前提交未采纳该建议，但审核者 HaiShaw 已批准 PR，可能认为当前条件已足够。

## 风险与影响

- 风险：风险极低：仅修改了两行代码，且 page_size=1 时新值与旧值数学等价，因此不会影响现有使用 page_size=1 的用户。对于 page_size>1 的场景，修复经过了 GSM8K 精度验证（从 0.005 恢复到 0.970）和性能基准测试验证。AI 审查提出的 init_cuda_graph_state 中的潜在问题在当前提交中未被完全解决，但经审核者 HaiShaw 批准，风险可接受。
- 影响：对用户：AMD 平台上使用 AITER 注意力后端和 DeepSeek 模型的用户现在可以安全使用 --page-size > 1，获得更高预填性能而无需担心精度损失。 对系统：改动仅影响 AITER 后端元数据生成，没有改变数据路径或内核，因此对其他后端无影响。 对团队：提供了清晰的文档说明为什么只需修改这两处，有助于未来的维护。
- 风险标记：暂无

## 关联脉络

- PR #27004 fix(disagg): correct DSA/SWA state-page transfer mismatch in PD disaggregation: 都与分页元数据正确性相关，修复了不同后端的 page_size 相关 bug。