PR #46114 完整报告

vllm-project/vllm

[ROCm][Bugfix] Fix chunk alignment when using context parallelism with TRITON_MLA

合并时间：2026-06-25 12:07

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/46114

执行摘要

- 一句话：修复 ROCm MLA context parallelism 的 chunk 对齐问题
- 推荐动作：值得精读，特别是对于涉及跨平台条件编译的项目。该 PR 演示了如何通过移除不必要的平台特定门控来修复 bug，并展示了测试重构以提升可靠性的典型手法。

功能与动机

There is a bug in mla_attention when using context parallelism on ROCm. max_context_chunk is being aligned properly on CUDA because of the self.aot_schedule path (which is CUDA-only). The chunk misalignment causes zero accuracy in test_context_parallel.py on ROCm with dcp_size=4 using TRITON_MLA.

实现拆解

1. 移除平台条件门控：在 MLACommonAttention.__init__ 中，原代码仅当 self.aot_schedule（CUDA）时才设置 self.page_size，导致 ROCm 上 page_size 未初始化。现改为无条件设置 self.page_size = self.kv_cache_spec.block_size。
2. 统一 chunk 对齐逻辑：在 MLACommonAttention.build 方法中，原代码的 round_down 对齐操作仅在 self.aot_schedule 条件下执行，现移除该条件，使 max_context_chunk 始终按 page_size 对齐，确保所有 GPU 平台一致行为。
3. 测试配套重构：在 tests/distributed/test_context_parallel.py 中，根据 current_platform.is_rocm() 分流模型配置，ROCm 下仅测试 dcp_multipliers=[1]（避免 CUDA 专属后端）。同时将评估方法从自定义 evaluate_gsm8k 切换为 lm_eval.simple_evaluate，使用 local-completions 接口，提高结果稳定性。

关键文件：

- vllm/model_executor/layers/attention/mla_attention.py（模块 注意力层；类别 source；类型 core-logic；符号 MLACommonAttention.init, MLACommonAttention.build）：核心修复文件：移除平台条件门控，使 page_size 初始化和 chunk 对齐逻辑在所有 GPU 平台上统一执行。
- tests/distributed/test_context_parallel.py（模块 并行测试；类别 test；类型 test-coverage；符号 _test_cp_gsm8k, CP_TEXT_GENERATION_MODELS）：测试增强：根据平台分流模型配置，改用 lm_eval 提高评估稳定性。

关键符号：MLACommonAttention.init, MLACommonAttention.build

关键源码片段

vllm/model_executor/layers/attention/mla_attention.py

核心修复文件：移除平台条件门控，使 page_size 初始化和 chunk 对齐逻辑在所有 GPU 平台上统一执行。

```
# 在 __init__ 中：移除平台条件，统一设置 page_size
self.page_size = self.kv_cache_spec.block_size

# 在 build() 方法中处理 chunked context 对齐：
if max_context_len_cpu > 0:
    max_context_chunk = (
        self.chunked_prefill_workspace_size // num_prefills_with_context_cpu
    )
    # 之前只在 aot_schedule (CUDA) 下对齐，现在无条件执行 round_down
    max_context_chunk = round_down(max_context_chunk, self.page_size)
    assert max_context_chunk > 0
    num_chunks = cdiv(max_context_len_cpu, max_context_chunk)
```

评论区精华

Review 中 LucasWilkinson 提出质疑：为什么不直接使用 round_down(max_context_chunk, self.kv_cache_spec.block_size) 而是引入 lcm？作者 micah-wil 反问能否移除 self.aot_schedule 门控。Lucas 指出该门控已是历史遗留（最初因为 self.kv_cache_spec.block_size 需要从 runner 获取，现已可直接使用）。最终共识：移除 aot_schedule 条件，让对齐逻辑无条件执行，简化代码。

- 简化 chunk 对齐逻辑，移除平台条件门控 (design): 采用直接无条件 round_down 方案，删除平台门控，简化代码。

风险与影响

- 风险：该 PR 改动集中在注意力层核心，但改动量小且逻辑清晰：将原本 CUDA 专属的对齐行为推广到所有平台。潜在风险是如果未来某些平台需要不同的对齐策略，该统一逻辑可能不适用，但当前所有 GPU 平台均使用相同的 block_size 语义，风险较低。测试已覆盖 ROCm 下的 context parallelism 场景，回归风险较小。
- 影响：直接影响使用 ROCm + TRITON_MLA + context parallelism (dcp > 1) 的用户，修复了此前零准确率的问题。对 CUDA 平台无功能变化（对齐逻辑与之前一致）。测试改用 lm_eval 带来更稳定的 CI 结果，但引入了新的依赖 (lm_eval)。
- 风险标记：核心路径变更，平台条件依赖移除，测试增强覆盖

关联脉络

- 暂无明显关联 PR