PR #22386 完整报告

sgl-project/sglang

[lora] Speedup triton backend `sgemm` calls with better grid

合并时间：2026-04-16 04:47

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/22386

执行摘要

- 一句话：优化多 LoRA 解码 Triton sgemm 内核网格调度，通过适配器排序减少 GPU 块启动数。
- 推荐动作：该 PR 值得精读，重点关注内核中 _resolve_token_positions 的设计和排序实现，以及性能权衡；建议结合基准测试评估实际收益，并注意测试覆盖的完整性。

功能与动机

根据 PR body 描述，动机是在多 LoRA 解码时，每个序列在 Triton sgemm 网格中都有自己的段，即使许多序列共享相同适配器，导致网格随 batch_size 而非 num_adapters 扩展，启动过多块并浪费 GPU 周期。通过排序令牌并合并每适配器段，可减少网格块数，提高 GPU 利用率。

实现拆解

1. 新增内核工具函数：在 python/sglang/srt/lora/triton_ops/kernel_utils.py 中定义 _resolve_token_positions Triton JIT 函数，用于在排序时通过排列间接访问令牌位置，支持 SORTED_BY_ADAPTER 常量路径。
2. 修改 Triton 后端逻辑：在 python/sglang/srt/lora/backend/triton_backend.py 中，添加 _sgemm_info 方法统一处理合并的批信息，更新 run_lora_a_sgemm、run_lora_b_sgemm、run_qkv_lora、run_gate_up_lora 等方法使用该信息，并添加 compute_sgemm_routing 函数构建每适配器批信息（通过 argsort 和 searchsorted）。
3. 更新所有四个 sgemm 内核：在 sgemm_lora_a.py、sgemm_lora_b.py、qkv_lora_b.py、gate_up_lora_b.py 中，引入 SORTED_BY_ADAPTER 常量和空段早期退出逻辑，使用 _resolve_token_positions 解析令牌位置，调整指针计算以支持排序路径。
4. 添加测试覆盖：新增 test/registered/lora/test_sgemm_sorted_by_adapter.py 测试文件，包含 _make_batch_info、_make_sorted_batch_info 辅助函数和 test_sgemm_lora_a 等测试用例，验证排序前后输出在 bf16 精度下数值等效（atol=1e-4），覆盖混合秩和单适配器边缘情况。
5. CUDA 图缓冲预分配：在 triton_backend.py 的 init_cuda_graph_batch_info 方法中预分配缓冲以支持排序路径，确保 CUDA 图兼容性。

关键文件：

- python/sglang/srt/lora/backend/triton_backend.py（模块 LoRA 后端；类别 source；类型 core-logic；符号 _sgemm_info, compute_sgemm_routing）：核心后端逻辑文件，负责 LoRA sgemm 调用和批信息处理，新增 _sgemm_info 方法统一处理合并段，并更新所有 sgemm 相关方法。
- python/sglang/srt/lora/triton_ops/kernel_utils.py（模块 内核工具；类别 infra；类型 infrastructure；符号 _resolve_token_positions）：新增内核工具函数文件，定义 _resolve_token_positions 用于在排序时通过排列间接访问令牌位置，是内核修改的关键基础。
- test/registered/lora/test_sgemm_sorted_by_adapter.py（模块 测试覆盖；类别 test；类型 test-coverage；符号 _make_batch_info, _make_sorted_batch_info, _check_close, test_sgemm_lora_a）：新增测试文件，验证排序前后 sgemm 内核输出数值等效性，确保功能正确性和覆盖边缘情况。
- python/sglang/srt/lora/triton_ops/sgemm_lora_a.py（模块 内核实现；类别 infra；类型 infrastructure）：sgemm_lora_a 内核文件，修改以支持 SORTED_BY_ADAPTER 路径和空段早期退出，影响核心计算逻辑。
- python/sglang/srt/lora/triton_ops/sgemm_lora_b.py（模块 内核实现；类别 infra；类型 infrastructure）：sgemm_lora_b 内核文件，类似修改以支持排序路径，影响输出计算。

关键符号：_resolve_token_positions, _sgemm_info, compute_sgemm_routing, _make_batch_info, _make_sorted_batch_info

关键源码片段

python/sglang/srt/lora/backend/triton_backend.py

核心后端逻辑文件，负责 LoRA sgemm 调用和批信息处理，新增 _sgemm_info 方法统一处理合并段，并更新所有 sgemm 相关方法。

```
  def _sgemm_info(self, pruned_batch_info=None):
      """返回sgemm批信息（当可用时合并段）。

      如果提供pruned_batch_info则直接返回，否则检查sgemm_batch_info属性，
      回退到self.batch_info。这支持按适配器排序后的合并段处理。
      """
      if pruned_batch_info is not None:
          return pruned_batch_info
      return getattr(self, "sgemm_batch_info", None) or self.batch_info

  def run_lora_a_sgemm(
      self,
      x: torch.Tensor,
      weights: torch.Tensor,
      pruned_batch_info: LoRABatchInfo = None,
      stack_num: int = 1,
      *args,
      **kwargs,
  ) -> torch.Tensor:
```

```
    """运行LoRA A sgemm，使用_sgemm_info获取批信息。

    通过_sgemm_info统一处理排序或非排序路径，确保内核调用正确。
    """
    return sgemm_lora_a_fwd(
        x, weights, self._sgemm_info(pruned_batch_info), stack_num=stack_num
    )
```

python/sglang/srt/lora/triton_ops/kernel_utils.py

新增内核工具函数文件，定义 _resolve_token_positions 用于在排序时通过排列间接访问令牌位置，是内核修改的关键基础。

```
import triton
import triton.language as tl

@triton.jit
def _resolve_token_positions(
    sorted_token_ids, # 排序后的令牌ID数组
    seg_start, # 段起始索引
    s_offset, # 段内偏移
    seg_len, # 段长度
    SORTED_BY_ADAPTER: tl.constexpr # 常量标志，指示是否按适配器排序
):
    """映射逻辑段偏移到物理令牌位置。

    当SORTED_BY_ADAPTER为True时，段按适配器分组，sorted_token_ids提供
    到原始令牌行的间接访问；否则令牌已连续，直接返回seg_start + s_offset。
    """
    if SORTED_BY_ADAPTER:
        # 通过加载sorted_token_ids间接获取物理位置
        return tl.load(
            sorted_token_ids + seg_start + s_offset, mask=s_offset < seg_len
        ).to(tl.int64)
    return (seg_start + s_offset).to(tl.int64) # 直接计算连续位置
```

评论区精华

Review 评论为空，提交历史显示有迭代修复（如“fix tiny bug”），但无实质性技术讨论。

- 暂无高价值评论线程

风险与影响

- 风险：1. 性能风险：排序令牌引入额外开销，若适配器数量接近批大小，性能收益可能不显著；需实测验证。 2. 数值精度风险：测试使用 bf16 和 atol=1e-4，但在边缘场景（如混合秩）可能累积误差。 3. 兼容性风险：新增 permutation 字段和排序逻辑可能影响现有 CUDA 图捕获，需确保向后兼容。 4. 逻辑错误风险：内核中早期退出和间接访问逻辑复杂，可能引入 bug，如空段处理或指针计算错误。

- 影响：1. 用户影响：提升多 LoRA 解码场景下的 GPU 利用率，可能提高推理吞吐量，对批量请求用户有益。 2. 系统影响：减少内核网格块启动数，降低 GPU 资源浪费，优化系统整体性能；但增加排序开销，需权衡净收益。 3. 团队影响：引入新的排序机制和测试套件，增加代码维护复杂度，但提供性能优化范例。
- 风险标记：核心路径变更，数值精度风险，CUDA 图兼容性

关联脉络

- PR #22844 [AMD] Optimize _append_shared_to_topk_output by a single fused Triton kernel for Qwen3.5: 同为 Triton 内核性能优化，涉及融合内核以减少启动开销，与本 PR 的网格优化相关。
- PR #22782 [HiCache]Fix CP support for hybrid model: 涉及 LoRA 相关缓存优化，可能共享类似的多适配器处理逻辑。