PR #24420 完整报告

sgl-project/sglang

[LoRA] Fix qkv_proj LoRA buffer sizing when tp_size > num_key_value_heads

合并时间：2026-05-07 05:51

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/24420

执行摘要

- 一句话：修复 qkv_proj LoRA 在 KV 头复制时的缓冲区尺寸错误
- 推荐动作：此 PR 值得精读，尤其是有 LoRA 和 TP 相关开发需求的工程师。它展示了一个典型的分布式训练 / 推理中因张量布局理解错误导致的 bug 及其修复模式，对理解 QKVParallelLinear 的 KV 头复制机制和 LoRA 权重切片非常有帮助。设计决策（在缓冲分配和切片两端保持一致的每 rank 维度计算）是可靠的。

功能与动机

在 TP 数大于 KV 头数时，QKVParallelLinear 的 KV 头会被复制到多个 rank 上，而 LoRA 的缓冲区分配和权重切片未考虑到这种复制，导致加载任何含有 qkv_proj 的 LoRA adapter 都会因形状断言失败而崩溃。PR 提供了具体复现命令和错误信息。

实现拆解

1. mem_pool.py：新增 _column_parallel_lora_b_per_rank_dim 方法在 LoRAMemoryPool 中，用于计算 column-parallel 模块（尤其是 qkv_proj）的每 rank LoRA B 输出维度。对于非 qkv_proj 模块或 tp_size ≤ num_kv_heads 场景，直接均匀分割；对于 tp_size > num_kv_heads 场景，从总输出中减去 KV 的总维度后，再按 rank 均匀分割 Q 部分，最后加上每个 rank 固定的 head_dim * 2（K 和 V 各一个 head）。同时处理了多模态模型的配置解析（get_text_config）。
2. layers.py：修复 slice_lora_b_weights 的索引逻辑在 QKVParallelLinearWithLoRA 中，原先使用 base_layer.output_sizes[1] 作为 K 的完整维度，但该值在 KV 复制场景下是未复制的原始大小，导致 V 切片偏移错误。改为使用 output_sizes[1] // num_kv_head_replicas 得到每个 rank 实际拥有的 K 维度，使索引与 PEFT 格式的 B 张量布局一致。
3. 测试文件（后经 review 移除）：曾包含一个 386 行的单元测试 test_qkv_lora_kv_replication.py，覆盖了 _column_parallel_lora_b_per_rank_dim 和 slice_lora_b_weights 的各种场景，但 reviewer 认为不需要单独测试文件，最终被移除。

关键文件：

- python/sglang/srt/lora/mem_pool.py（模块 LoRA 内存池；类别 source；类型 core-logic；符号 _column_parallel_lora_b_per_rank_dim）：新增 _column_parallel_lora_b_per_rank_dim 方法，是修复的核心：正确计算 qkv_proj 在 KV

头复制场景下的每 rank LoRA B 输出维度。

- python/sglang/srt/lora/layers.py（模块 LoRA 层；类别 source；类型 core-logic；符号 slice_lora_b_weights）：修复 slice_lora_b_weights 中 K 和 V 的偏移计算，使用 output_sizes[1] // num_kv_head_replicas 取代原始的 output_sizes[1]，保证索引与 PEFT 格式的 B 张量布局一致。

关键符号：_column_parallel_lora_b_per_rank_dim, slice_lora_b_weights, get_lora_B_shape

关键源码片段

python/sglang/srt/lora/mem_pool.py

新增 _column_parallel_lora_b_per_rank_dim 方法，是修复的核心：正确计算 qkv_proj 在 KV 头复制场景下的每 rank LoRA B 输出维度。

```
def _column_parallel_lora_b_per_rank_dim(
    self,
    module_name: str,
    total_output_dim: int,
    effective_tp_size: int,
) -> int:
    """Per-rank LoRA B output dim for column-parallel modules.

    For most modules this is just an even split. For ``qkv_proj`` when
    ``effective_tp_size > num_key_value_heads``, the underlying
    :class:`QKVParallelLinear` *replicates* each KV head across
    ``tp_size // num_kv_heads`` ranks instead of dividing further, so
    each rank owns ``head_dim`` of K/V (not ``head_dim * num_kv_heads
    / tp_size``). A naive ``divide(total, tp_size)`` undersizes the
    buffer and produces a shape mismatch when the
    :meth:`QKVParallelLinearWithLoRA.slice_lora_b_weights` slice runs.
    """
    # 对于非 qkv_proj 模块，仍然使用均匀分割
    if module_name != "qkv_proj":
        return divide(total_output_dim, effective_tp_size)

    # 解析配置，处理多模态模型（例如 Qwen2-VL）
    cfg = self.base_hf_config
    if hasattr(cfg, "get_text_config"):
        cfg = cfg.get_text_config()
    num_kv_heads = getattr(cfg, "num_key_value_heads", None)
    # 如果不存在 num_kv_heads 或 tp_size 不大于 kv_heads，则无需特殊处理
    if num_kv_heads is None or num_kv_heads >= effective_tp_size:
        return divide(total_output_dim, effective_tp_size)

    # 计算 head_dim，优先使用显式配置，否则从 hidden_size 和 num_attention_heads 推导
    head_dim = getattr(cfg, "head_dim", None) or (
        cfg.hidden_size // cfg.num_attention_heads
```

```
    )
    # KV 部分总维度 = 2 * num_kv_heads * head_dim
    kv_dim_total = 2 * num_kv_heads * head_dim
    # Q 部分总维度 = 总输出维度 - KV 部分
    q_dim_total = total_output_dim - kv_dim_total
    # 每个 rank 的 Q 维度 = Q 总维度 / effective_tp_size
    q_per_rank = divide(q_dim_total, effective_tp_size)
    # 每个 rank 的最终维度 = Q_per_rank + 2 * head_dim（每个 rank 拥有完整的 1 个 K 和 1 个 V head）
    return q_per_rank + 2 * head_dim
```

python/sglang/srt/lora/layers.py

修复 slice_lora_b_weights 中 K 和 V 的偏移计算，使用 output_sizes[1] // num_kv_head_replicas 取代原始的 output_sizes[1]，保证索引与 PEFT 格式的 B 张量布局一致。

```
def slice_lora_b_weights(self, B: torch.Tensor, tp_rank: int) -> torch.Tensor:
    base_layer = self.base_layer
    q_proj_shard_size = base_layer.q_proj_shard_size
    kv_proj_shard_size = base_layer.kv_proj_shard_size
    num_kv_head_replicas = base_layer.num_kv_head_replicas

    q_start_idx = q_proj_shard_size * tp_rank
    q_end_idx = q_start_idx + q_proj_shard_size

    kv_shard_id = tp_rank // num_kv_head_replicas
    kv_start_idx = kv_proj_shard_size * kv_shard_id
    kv_end_idx = kv_start_idx + kv_proj_shard_size

    # 重要修复：`base_layer.output_sizes[1]` 是未复制的完整 K 维度，
    # 需要除以 num_kv_head_replicas 才能得到每个 rank 实际拥有的 K 维度。
    q_size = base_layer.output_sizes[0]
    k_size = base_layer.output_sizes[1] // num_kv_head_replicas
    B_q_shard = B[q_start_idx:q_end_idx, :]
    B_k_shard = B[q_size + kv_start_idx : q_size + kv_end_idx, :]
    B_v_shard = B[q_size + k_size + kv_start_idx : q_size + k_size + kv_end_idx, :]

    return torch.concat(
        (
            B_q_shard,
            B_k_shard,
            B_v_shard,
        ),
        dim=0,
    )
```

评论区精华

1. _text_config 未初始化问题：机器人 reviewer 指出 _column_parallel_lora_b_per_rank_dim 中使用了未初始化的 self._text_config，会导致 AttributeError，建议改用 self.base_hf_config 并调用 get_text_config()。作者在第二个提交中采纳并修复。
2. 测试文件移除：reviewer Fridge003 明确要求删除新增的测试文件 test_qkv_lora_kv_replication.py，认为不需要。最终提交中已删除。
3. 注释精简：Fridge003 建议删除代码中重复的解释性注释，因为 docstring 已经足够。作者在第三个 commit 中处理了相关注释。

- 未初始化属性 _text_config 导致 AttributeError (correctness): 作者在第二个 commit 中采用建议，将 self._text_config 替换为 self.base_hf_config 并添加 get_text_config() 的多模态兼容处理。
- 测试文件是否必要 (testing): 测试文件被移除，最终提交中不包含该测试。
- 冗余注释清理 (style): 作者在第三个 commit 中清理了相关注释。

风险与影响

- 风险：
  1. 回归风险：改动集中在 qkv_proj 的 LoRA 路径，非 qkv_proj 模块和 tp_size ≤ num_kv_heads 场景走原均匀分割逻辑，与之前行为一致。PR 在提交历史中测试通过，CI 通过。
  2. 配置兼容性：_column_parallel_lora_b_per_rank_dim 依赖 head_dim 属性，对于没有显式 head_dim 的模型，通过 hidden_size // num_attention_heads 计算，存在潜在精度问题（但通常 head_dim 会整除）。
  3. 多模态模型：代码中已处理 get_text_config，但未覆盖所有可能的配置结构，若模型 config 中没有 num_key_value_heads 或 head_dim 会走 fallback，行为不变。
- 影响：
  1. 用户影响：修复了在 tp_size > num_kv_heads 时（如 Qwen3.5-35B-A3B 的 tp=4、kv_heads=2）加载 qkv_proj LoRA 的崩溃问题，使得此类模型可以正常使用 LoRA。
  2. 系统影响：仅修改了 LoRA 初始化路径，推理性能无影响（辅助函数仅在初始化时调用一次）。
  3. 团队影响：变更范围小（2 个源文件，40 行新增），但涉及到对 Tensor Parallelism 中 KV 头复制语义的正确理解，设计文档清晰。 - 风险标记：核心路径变更，多模态配置兼容性

关联脉络

- 暂无明显关联 PR