PR #33623 完整报告

sgl-project/sglang

[Kimi K3] Fuse MLA gate projection into QKV-A GEMM

合并时间：2026-08-13 00:21

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/33623

执行摘要

- 一句话：Kimi-K3 融合 MLA 门投影进 QKV-A GEMM，减少一次投影 GEMM
- 推荐动作：值得精读，但建议结合后续回滚 PR#34642 一起阅读。本 PR 展示了安全的算子融合方法：通过检查 quant_method 与 dtype 限定未量化场景、用 _merge_weights_as_views 零拷贝拼接权重、以 (gate, None) 表达同流已算好的 gate 避免 CUDA Graph 流等待问题，并用 SimpleNamespace 替身做轻量 CPU 等价性测试。同时，BBuf 的独立 A/B 测试是 review 中值得借鉴的验证方式——它直接戳破了微基准与端到端收益的落差。阅读重点是 prepare_qkv_latent 的 fallback 分级和 pick_tile_m 的 wave 填充策略。

功能与动机

PR body 明确说明：Kimi-K3 MLA computes QKV-A and the TP-local output gate from the same hidden states. They currently run as separate GEMMs. Fuse them to reduce projection cost while keeping the gate output TP-local. 即两份投影共享输入，分别执行浪费一次 GEMM 的权重读取与 kernel 启动开销；同时 gate 必须保持 TP-local（attention 输出在 attn-TP 组内分片，g_proj 需按 attn-TP 分片），因此融合必须维持该分片契约，这也是 _qkv_a_g_proj_sizes 记录 split 尺寸的原因。

实现拆解

实现按以下 4 步推进：

1. 权重合并入口：在 python/sglang/srt/models/kimi_k3.py 的 KimiK3MLAAttention 中新增 _merge_qkv_a_g_proj_weights()，并在 load_weights 逐层处理时对 MLA 层调用（与 KimiK3DeltaAttention 已有的 _merge_bfa_weights() 挂载点并列）。合并条件包括 use_output_gate 开启、两个模块 quant_method 均为 UnquantizedLinearMethod、权重 dtype 相同且为 BF16/FP16；命中后通过 _merge_weights_as_views 拼接权重并记录 split 尺寸，不产生额外拷贝。
2. 前向路径改造：覆写 prepare_qkv_latent()，在合并条件下用一次 dsv3_fused_a_gemm（低延迟路径：M=1~16、deterministic 关闭、权重形状对齐、fused_a_gemm_weight_eligible 通过）或 _k3_bf16_gemm（fallback）完成融合投影，再用 torch.split 分出 qkv_latent 与 gate，并把 (gate, None) 存入 _gate_precomputed。producer stream 为 None 表示 gate 已在当前流算好；_gated_o_proj_forward 的 wait_stream 条件相应改为 precomputed[1] is not None，避免 CUDA Graph 捕获下等待

不存在的流。forward() 在融合启用时跳过 _precompute_output_gate，防止 alt stream重复计算。

3. Kernel 调优：python/sglang/kernels/jit/csrc/gemm/dsv3_fused_a_gemm.cuh 的 pick_tile_m 为 hd_out == 3648 && hd_in == 7168（K3 TP8 融合形状）新增 tile_m = 32 特化，使 114 个 CTA 恰好填满 H200 的一个 wave；作者报告该修复把 fused-A kernel 从 16.8~18.0 us 降到 14.9~15.1 us。
4. 测试与配套验证：新增 test/registered/unit/models/test_kimi_k3_mla_gate_fusion.py，注册 CPU CI（base-a-test-cpu）与 CUDA kernel 单测（base-b-kernel-unit）。CPU 测试用 SimpleNamespace 替身验证合并投影输出与分离 F.linear 输出一致；SM90+ 测试在 K3 TP8 维度下对 M=1/8/16 做 CUDA Graph 捕获与 replay，断言数值接近（rtol=1e-2, atol=1e-3）。作者另提供 H200 单 rank 微基准数据（FlashInfer MLA、CUDA Graph、BF16），但未覆盖 TP8 端到端验证。

关键文件：

- python/sglang/srt/models/kimi_k3.py（模块 模型实现；类别 source；类型 data-contract；符号 _merge_qkv_a_g_proj_weights, prepare_qkv_latent, _gated_o_proj_forward）：核心实现文件：KimiK3MLAAttention 新增权重合并方法，覆写 prepare_qkv_latent，并调整 _gated_o_proj_forward 的流等待逻辑与 forward 的 gate 预计算条件。
- test/registered/unit/models/test_kimi_k3_mla_gate_fusion.py（模块 单元测试；类别 test；类型 test-coverage；符号 TestKimiK3MlaGateFusion, test_merged_projection_matches_separate_projections, test_fused_a_cuda_graph_replay）：新增测试覆盖 CPU 等价性与 SM90+ CUDA Graph replay，验证融合路径输出与分离参考一致，是 PR 正确性的主要保障。
- python/sglang/kernels/jit/csrc/gemm/dsv3_fused_a_gemm.cuh（模块 JIT 内核；类别 source；类型 core-logic）：pick_tile_m 为 K3 合并形状 [3648, 7168] 增加 tile_m=32 特化，使 114 个 CTA 填满 H200 一个 wave，fused-A kernel 从约 17us 降到约 15us。

关键符号：_merge_qkv_a_g_proj_weights, prepare_qkv_latent, _gated_o_proj_forward, _precompute_output_gate, pick_tile_m

关键源码片段

python/sglang/srt/models/kimi_k3.py

核心实现文件：KimiK3MLAAttention 新增权重合并方法，覆写 prepare_qkv_latent，并调整 _gated_o_proj_forward 的流等待逻辑与 forward 的 gate 预计算条件。

```
def _merge_qkv_a_g_proj_weights(self) -> None:
    """合并同输入的 MLA qkv-a 与 TP-local 输出门的权重，减少一次投影 GEMM。"""
    if not self.use_output_gate:
        return
    mods = [self.fused_qkv_a_proj_with_mqa, self.g_proj]
    # K3 的全局 MXFP4 配置会忽略 attention 层，因此不能用 quant_config 判断，
    # 要检查实际解析出来的 quant_method 是否仍是未量化线性层。
    if any(
        not isinstance(mod.quant_method, UnquantizedLinearMethod) for mod in mods
```

```
    ):
        # 量化路径保持原有 fallback 行为，不参与融合。
        return
    dtypes = {mod.weight.dtype for mod in mods}
    if len(dtypes) != 1 or dtypes.pop() not in (torch.bfloat16, torch.float16):
        # 只在 BF16 / FP16 未量化场景下融合，其他 dtype 一律回退。
        return
    # 以视图方式拼接权重，避免额外拷贝；sizes 用于后续 torch.split。
    self._qkv_a_g_proj_weight, self._qkv_a_g_proj_sizes = _merge_weights_as_views(
        mods
    )


def prepare_qkv_latent(self, hidden_states, forward_batch):
    """一次融合 GEMM 同时产出 qkv_latent 与 gate，替代原来的两次独立投影。"""
    weight = self._qkv_a_g_proj_weight
    if (
        weight is None
        or not isinstance(hidden_states, torch.Tensor)
        or getattr(self.fused_qkv_a_proj_with_mqa, "set_lora", False)
        or getattr(self.g_proj, "set_lora", False)
    ):
        # LoRA 或特殊输入（如多模态拼接）时退回父类路径，保证兼容。
        return super().prepare_qkv_latent(hidden_states, forward_batch)

    if self._use_min_latency_fused_a_gemm is None:
        # 低延迟 fused kernel 仅在确定性推理关闭、输出通道数对齐 16 且权重满足约束时启用。
        self._use_min_latency_fused_a_gemm = (
            not get_exec().deterministic.enable_deterministic_inference
            and weight.shape[0] % 16 == 0
            and fused_a_gemm_weight_eligible(self.fused_qkv_a_proj_with_mqa)
        )
    if self._use_min_latency_fused_a_gemm and 1 <= hidden_states.shape[0] <= 16:
        fused = dsv3_fused_a_gemm(
            hidden_states, weight.T, backend=self.fused_a_gemm_backend
        )
    else:
        # 大批量或条件不满足时回落普通 BF16 GEMM，行为与非融合路径数值接近。
        fused = _k3_bf16_gemm(hidden_states, weight)
    qkv_latent, gate = torch.split(fused, self._qkv_a_g_proj_sizes, dim=-1)
    # producer stream 为 None 表示 gate 已在当前流算好，_gated_o_proj_forward
    # 会据此跳过 wait_stream，保证 CUDA Graph 捕获下行为一致。
    self._gate_precomputed = (gate, None)
    return qkv_latent
```

test/registered/unit/models/test_kimi_k3_mla_gate_fusion.py

新增测试覆盖 CPU 等价性与 SM90+ CUDA Graph replay，验证融合路径输出与分离参考一致，是 PR 正确性的主要保障。

```
def test_merged_projection_matches_separate_projections(self):
    # 用 SimpleNamespace 构造轻量替身，避免加载完整模型，CPU 上即可验证融合逻辑。
    qkv_proj = SimpleNamespace(
        weight=torch.nn.Parameter(torch.randn(12, 16, dtype=torch.bfloat16)),
        quant_method=UnquantizedLinearMethod(),
    )
    g_proj = SimpleNamespace(
        weight=torch.nn.Parameter(torch.randn(8, 16, dtype=torch.bfloat16)),
        quant_method=UnquantizedLinearMethod(),
    )
    attn = SimpleNamespace(
        use_output_gate=True,
        quant_config=object(),
        fused_qkv_a_proj_with_mqa=qkv_proj,
        g_proj=g_proj,
        _qkv_a_g_proj_weight=None,
        _qkv_a_g_proj_sizes=None,
        _use_min_latency_fused_a_gemm=False,
        _gate_precomputed=None,
    )
    x = torch.randn(3, 16, dtype=torch.bfloat16)
    # 参考：分别做两次独立的 F.linear 投影。
    expected_qkv = torch.nn.functional.linear(x, qkv_proj.weight)
    expected_gate = torch.nn.functional.linear(x, g_proj.weight)

    KimiK3MLAAttention._merge_qkv_a_g_proj_weights(attn)
    qkv = KimiK3MLAAttention.prepare_qkv_latent(attn, x, None)
    gate, stream = attn._gate_precomputed

    # 融合输出必须与分离投影逐元素一致，且 gate 对应的 producer stream 为 None。
    torch.testing.assert_close(qkv, expected_qkv)
    torch.testing.assert_close(gate, expected_gate)
    self.assertIsNone(stream)
```

评论区精华

本次 review 最有价值的讨论来自 BBuf 在 issue 评论中的独立 A/B 验证（PR 本身无 review 评论）：

- BBuf 在 8×B300（SM103）TP8、trtllm_mla、flashinfer_mxfp4 MoE、CUDA Graph（max bs 64）、release/v0.5.17 基线上对比加本 PR diff 的效果：bs=1 时 mean TPOT 由 8.42/8.41 ms 变为 8.45/8.44 ms（约 −0.4%），bs=64 稳态吞吐从 2206±5 变为 2204±3 tok/s（±0%，噪声内）。
- 关键结论：融合路径确认已启用（JIT cache 出现新的 dsv3_fused_a_gemm_7168_3648_*_arch_10.3a 构建，且 _use_min_latency_fused_a_gemm 条件全部成立），但端到端无稳定收益，说明 H200 单 rank 小批量微基准的收益无法外推到真实 TP8 场景。
- nvpohanh 评论中 cc @YAMY1234 @leejnau，属于征询相关维护者进一步确认；没有后续公开结论，PR 最终由 BBuf approve 并合并。

- B300 TP8 端到端独立 A/B 验证与 H200 微基准的差距 (performance): 融合在端到端 TP8 场景无稳定收益，bs=1 甚至略降；与作者 H200 单 rank 微基准（+1.3% 至 +4.5%）形成鲜明对比。BBuf 最终仍 approve。
- 拉取相关维护者复核 (question): 无后续公开讨论记录；PR 最终由 BBuf approve 并合并。

风险与影响

- 风险：
  1. 长序列性能回归（已被证实）：融合路径中 gate 必须在 QKV-A GEMM 内同流算完，原 _precompute_output_gate 的 alt-stream 与 attention core 重叠机制被绕过，长序列场景引发回归，最终被 PR#34642 完整回滚。这是本 PR 最大的技术风险，且已实际发生。
  2. 端到端收益不明：BBuf 在 B300 TP8 上测到 bs=1 约 −0.4%、bs=64 约 ±0%（噪声内）；作者 H200 的 +1.3% 至 +4.5% 收益集中在低占用小批量场景，无法推广到生产批量。
  3. 数值一致性：fused-a kernel 与分离 GEMM 结果非 bit-exact，测试仅以 rtol=1e-2/atol=1e-3 断言；确定性推理开关开启时会自动回退非融合路径，规避一致性问题。
  4. CUDA Graph 兼容性：融合路径的 (gate, None) 约定简化了流等待逻辑，且新增了 replay 测试，但 breakable CUDA Graph 分段场景未专门覆盖。
  5. 适用面窄：K3 主流生产配置（MXFP4 量化、LoRA 微调）都会自动回退到非融合路径，收益面比标题暗示的更窄。 - 影响：对用户：使用未量化 BF16/FP16 Kimi-K3 的部署在 H200 小批量解码场景可能获得约 1% 至 4% 吞吐提升，但 TP8 端到端无稳定收益，且长序列场景出现回归，最终回滚后行为恢复原状。对系统：改动集中在 KimiK3MLAAttention 与 dsv3_fused_a_gemm kernel，涉及 _gate_precomputed 数据契约扩展为 (gate, stream_or_None)，并新增一条受控的融合前向路径。对团队：融合模式本身的资格检查与 fallback 设计为后续 kernel 融合提供了可复用范式，但其结果也提示：微基准收益必须与端到端基准对比，且不能破坏已有的流重叠调度。 - 风险标记：端到端无稳定收益，长序列回归（已被回滚），核心路径变更，数值非 bit-exact, CUDA Graph 兼容性

关联脉络

- PR #33521 [Kimi K3] Fuse MLA gate projection into QKV-A GEMM: PR body 说明本 PR 是其 rebase 替代品：原 PR 因 base 分支被删除而自动关闭。
- PR #34642 Revert "[Kimi K3] Fuse MLA gate projection into QKV-A GEMM": 本 PR 合并后因引发长序列回归被完整回滚（涉及同一批文件：kimi_k3.py、test_kimi_k3_mla_gate_fusion.py、dsv3_fused_a_gemm.cuh），是理解本 PR 最终结果的关键关联。