PR #27583 完整报告

sgl-project/sglang

[AMD] Enable fused GDN QKV split Triton kernel on HIP

合并时间：2026-06-11 17:01

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/27583

执行摘要

- 一句话：AMD HIP 启用 GDN QKV 融合拆分核
- 推荐动作：建议合入。改动简洁、风险低、收益明确。可作为 AMD GPU 上 Triton kernel 启用策略的参考案例。

功能与动机

GDN 预填充阶段中，经 causal conv1d 输出的 packed QKV 张量内存布局是跨步的（strided），使用 torch.split() 后接 .view() 会触发三次独立的逐元素拷贝 kernel，造成额外开销。而 fused_qkv_split_gdn_prefill 是 @triton.jit kernel，原生兼容 ROCm/HIP，但之前被 is_cuda() 条件错误拦截，导致 AMD GPU 无法使用。

实现拆解

1. 文件：python/sglang/srt/layers/attention/linear/gdn_backend.py - 在导入中添加 is_hip（来自 sglang.srt.utils）。 - 将 fused_qkv_split_gdn_prefill 的导入条件从 if is_cuda(): 改为 if is_cuda() or is_hip():。 - 将运行时的分派条件从 if is_cuda() and qkv_dim <= MAX_FUSED_QKV_SPLIT_DIM: 改为 if (is_cuda() or is_hip()) and qkv_dim <= MAX_FUSED_QKV_SPLIT_DIM:。
2. 无测试或配置变更：该 PR 仅涉及 3 行代码修改，无新增测试或配置。CI 中 AMD 的相关测试 test_qwen3_coder_next_8gpu.py 未运行，但经过精度测试（GSM8K）和端到端 benchmark 验证。

关键文件：

- python/sglang/srt/layers/attention/linear/gdn_backend.py（模块 GDN 后端；类别 source；类型 dependency-wiring）：唯一修改的文件，包含导入和分派条件的变更（+3/-3）。

关键符号：未识别

关键源码片段

python/sglang/srt/layers/attention/linear/gdn_backend.py

唯一修改的文件，包含导入和分派条件的变更（+3/-3）。

```
# gdn_backend.py ( 片段 )
```

```
from sglang.srt.utils import is_cpu, is_cuda, is_hip, is_npu # 新增 is_hip 导入

# 之前：if is_cuda():
if is_cuda() or is_hip(): # 开启 HIP 平台的 fused kernel
    from sglang.jit_kernel.triton.gdn_fused_proj import fused_qkv_split_gdn_prefill

# ... 在 forward_extend 方法中
qkv_dim = layer.q_dim + layer.k_dim + layer.v_dim
# 之前：if is_cuda() and qkv_dim <= MAX_FUSED_QKV_SPLIT_DIM:
if (is_cuda() or is_hip()) and qkv_dim <= MAX_FUSED_QKV_SPLIT_DIM: # HIP 也可使用 fused
kernel
    query, key, value = fused_qkv_split_gdn_prefill(
        mixed_qkv,
        layer.num_q_heads,
        layer.num_k_heads,
        layer.num_v_heads,
        layer.head_q_dim,
        layer.head_k_dim,
        layer.head_v_dim,
    )
else:
    # fallback 到 torch.split() + .view()
    query, key, value = torch.split(mixed_qkv, [layer.q_dim, layer.k_dim, layer.v_dim], dim=-1)
    query = query.view(1, actual_seq_len, layer.num_q_heads, layer.head_q_dim)
    key = key.view(1, actual_seq_len, layer.num_k_heads, layer.head_k_dim)
    value = value.view(1, actual_seq_len, layer.num_v_heads, layer.head_v_dim)
```

评论区精华

审核均通过，无实质讨论。sogalin 评论 "LGTM, it is clean and easy to understand."

- 暂无高价值评论线程

风险与影响

- 风险：风险较低。改动仅放宽条件判断，CUDA 路径完全不变（is_hip() 在 CUDA 上返回 False）。HIP 路径下 Triton kernel 本身已验证可原生运行。但 AMD CI 未覆盖该代码路径，可能引入潜在回归（如 kernel 在特定输入尺寸下出错），不过 kernel 已通过 GSM8K 精度测试。
- 影响：影响范围：仅 AMD GPU 上使用 GDN 层（如 Qwen3.5-397B-A17B-MXFP4）的预填充推理。CUDA 用户无变化。性能提升在 benchmark 中不显著（总吞吐量 +0.15%），但 kernel launch 数减少，对长序列推理有益。
- 风险标记：CI 未覆盖变更代码路径

关联脉络

- 暂无明显关联 PR