PR #46508 完整报告

vllm-project/vllm

[Kernel] Enable PDL for per_token_group_quant_8bit_kernel

合并时间：2026-06-25 08:42

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/46508

执行摘要

为 FP8 per-token-group 量化 CUDA kernel 启用 PDL（Programmatic Dependency Launch），使用官方 API 替代 asm，提升 grid 间依赖管理的正确性与可维护性。

功能与动机

PR#42996 引入了 PDL 框架以支持 CUDA grid 间依赖同步，但 per_token_group_quant_8bit_kernel 尚未接入。Review 指出原先使用的 asm volatile("griddepcontrol.launch_dependents;") 缺少 memory clobber 约束，编译器可能将输入数据的加载提前到 wait 之前，导致数据竞争。该 PR 修正此问题，确保正确启用 PDL。

实现拆解

1. kernel 函数内部：在数据加载前插入 cudaGridDependencySynchronize()，在量化完成后插入 cudaTriggerProgrammaticLaunchCompletion()，使用 #if 保护仅对 sm90+ 生效。
2. 启动宏重构：将原来的 LAUNCH_KERNEL 宏重写为 LAUNCH_KERNEL_INST，通过 cudaLaunchConfig_t 和 cudaLaunchAttributeProgrammaticStreamSerialization 配置流序列化，代替传统 <<<>>> 语法。
3. 条件编译：用 !defined(USE_ROCM) && (defined(__CUDA_ARCH__) && __CUDA_ARCH__>= 900) 隔离 PDL 代码，确保 ROCm 和不支持 PDL 的架构不受影响。

csrc/libtorch_stable/quantization/w8a8/fp8/per_token_group_quant.cu

唯一变更文件，修改了 CUDA kernel 和启动宏以启用 PDL。

```
// 文件：csrc/libtorch_stable/quantization/w8a8/fp8/per_token_group_quant.cu
//
// kernel 内部启用 PDL：在加载输入前等待前驱 grid，在量化后通知后继 grid
__global__ void per_token_group_quant_8bit_kernel(/* ... 参数 ... */) {
    // ... 前置计算 (groups_per_block 等) ...

#if (defined(__CUDA_ARCH__) && (__CUDA_ARCH__ >= 900))
    // PDL: 等待前驱 grid 完成，确保共享内存或其他依赖已就绪
    cudaGridDependencySynchronize();
#endif

    if constexpr (IS_COLUMN_MAJOR) {
        // ... column-major 路径：加载、量化、存储 ...
```

```
    } else {
        // ... row-major 路径 ...
    }

    // 执行量化核心 ( 向量化加载到 smem 并计算 scale/quant)
    QuantizeGroup<T, DST_DTYPE>(smem_group, group_output, group_size, lane_id,
                                threads_per_group, y_s, min_8bit, max_8bit);

#if (defined(__CUDA_ARCH__) && (__CUDA_ARCH__ >= 900))
    // PDL: 标记本 grid 完成，允许依赖本 grid 的后继 grid 启动
    cudaTriggerProgrammaticLaunchCompletion();
#endif
}

// 启动 kernel 的辅助宏：使用 cudaLaunchConfig_t + cudaLaunchAttribute
// 启用 programmatic stream serialization 以支持 grid 间依赖
#define LAUNCH_KERNEL_INST(T, DST_DTYPE, COL_MAJOR, UE8M0, SMEM_BYTES) \
    do { \
        cudaLaunchConfig_t config = {}; \
        config.gridDim = dim3(num_blocks); \
        config.blockDim = dim3(num_threads); \
        config.dynamicSmemBytes = (SMEM_BYTES); \
        config.stream = stream; \
        cudaLaunchAttribute attrs[1]; \
        attrs[0].id = cudaLaunchAttributeProgrammaticStreamSerialization; \
        attrs[0].val.programmaticStreamSerializationAllowed = 1; \
        config.attrs = attrs; \
        config.numAttrs = 1; \
        cudaLaunchKernelEx(&config, per_token_group_quant_8bit_kernel<T, DST_DTYPE, \
                           COL_MAJOR, UE8M0>, \
                           input_ptr, output_q_ptr, output_s_ptr, /* ... */); \
    } while(0)
```

评论区精华

- mgoin：指出 asm griddepcontrol.launch_dependents 缺少 memory clobber，编译器可自由重排指令。建议采用 cudaGridDependencySynchronize 和 cudaTriggerProgrammaticLaunchCompletion（参考 fused_deepseek_v4_qnorm_rope_kv_insert_kernel.cu）。
  - 作者立即采纳，将 asm 替换为官方 API，并通过新的启动宏统一配置。

风险与影响

- 风险：低。条件编译确保旧架构不受影响；唯一依赖是 CUDA 12.4+ 且架构为 sm90+。
- 影响：仅限于单个 kernel 的 FP8 量化路径。性能可能微升（因为正确的依赖管理可提升流并发），更关键的是消除了潜在的竞态数据错误。

关联脉络

该 PR 是 #42996（PDL 框架引入）的直接后续，为该框架下第一个完整接入的 kernel。后续可能有更多 kernel 采用相同模式，建议关注相近的量化 kernel 是否也需要同步更新。