PR #44334 完整报告

vllm-project/vllm

[10/n] Migrate cuda_view and silu_and_mul_per_block_quant kernels to torch stale ABI.

合并时间：2026-06-05 11:14

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/44334

执行摘要

本 PR 是 libtorch stable ABI 迁移的第 10 阶段，将 get_cuda_view_from_cpu_tensor 和 silu_and_mul_per_block_quant 两个核心 CUDA 内核从 legacy _C 迁移到 _C_stable_libtorch。同时剔除了重复编译的源文件，并提升了 torch 版本底线至 2.11，简化了实现。但该变更导致 ROCm 平台构建暂时受阻（需后续修复）。

功能与动机

继续推进 stable ABI 迁移，减少对 torch 私有 / 不稳定 ABI 的依赖，最终实现 torch 跨版本二进制兼容。PR 描述显示目标是将不稳定内核从 85 个降至 0，并消除 _C 和 _C_stable_libtorch 中的重复编译。

实现拆解

1. 注册迁移：在 csrc/libtorch_stable/torch_bindings.cpp 的 STABLE_TORCH_LIBRARY_FRAGMENT 中新增 get_cuda_view_from_cpu_tensor 和 silu_and_mul_per_block_quant 的 op 定义和实现绑定。
2. 旧注册清理：在 csrc/torch_bindings.cpp 中删除相同的 op 注册和 cuda_utils 函数库。
3. 文件搬迁：将 fused_silu_mul_block_quant.cu、cuda_view.cu、cutlass_extensions/common.cpp 和 common.hpp 移至 csrc/libtorch_stable/ 并更新所有包含路径。
4. 构建配置调整：在 CMakeLists.txt 中将 cuda_utils_kernels.cu 和 cutlass_extensions/common.cpp 从 VLLM_EXT_SRC 剔除，避免重复编译。
5. Torch 版本升级：TORCH_TARGET_VERSION 从 2.10 提升至 2.11，简化 cuda_view.cu 的 deleter 逻辑，消除了 2.10 fallback 的 use-after-free 风险。

csrc/libtorch_stable/torch_bindings.cpp

核心注册文件，新增两个 op 的 stable ABI 注册和 cuda_utils 的迁移

```
// 在 libtorch_stable 的 op 注册中，新增了两个关键操作：

STABLE_TORCH_LIBRARY_FRAGMENT(_C, ops) {
  // ... 其他已有注册 ...

  // [ 新增 ] 将 CPU tensor 转换为 CUDA 统一虚拟地址 (UVA) 视图
  ops.def("get_cuda_view_from_cpu_tensor(Tensor cpu_tensor) -> Tensor");
```

```
  // [ 新增 ] 融合 SiLU + 逐元素乘法 + 逐块量化 (FP8/INT8)
  ops.def(
      "silu_and_mul_per_block_quant("
      "Tensor! out, "
      "Tensor input, "
      "Tensor! scales, "
      "int group_size, "
      "Tensor? scale_ub=None, "
      "bool is_scale_transposed=False) -> ()");
}

STABLE_TORCH_LIBRARY_IMPL(_C, CUDA, ops) {
  // ... 其他已有 impl ...

  // [ 新增 ] silu_and_mul_per_block_quant 的 CUDA 实现绑定
  ops.impl("silu_and_mul_per_block_quant",
           TORCH_BOX(&silu_and_mul_per_block_quant));
}

STABLE_TORCH_LIBRARY_IMPL(_C, CPU, ops) {
  // [ 新增 ] get_cuda_view_from_cpu_tensor 的 CPU 实现绑定
  ops.impl("get_cuda_view_from_cpu_tensor",
           TORCH_BOX(&get_cuda_view_from_cpu_tensor));
}

// [ 新增 ] cuda_utils 操作也迁移到 stable ABI
STABLE_TORCH_LIBRARY_FRAGMENT(_C_cuda_utils, cuda_utils) {
  cuda_utils.def("get_device_attribute(int attribute, int device_id) -> int");
  cuda_utils.def(
      "get_max_shared_memory_per_block_device_attribute(int device_id) -> int");
}
```

csrc/libtorch_stable/ops.h

声明迁移后的函数接口，使用 torch::stable::Tensor

```
// 在 ops.h 中新增了以下函数声明，采用 torch::stable::Tensor 代替 torch::Tensor

// CPU tensor -> CUDA UVA view (shared CUDA/ROCm)
torch::stable::Tensor get_cuda_view_from_cpu_tensor(
    torch::stable::Tensor& cpu_tensor);

// 融合 SiLU+Mul + per-block 量化 (FP8/INT8) (shared CUDA/ROCm)
void silu_and_mul_per_block_quant(torch::stable::Tensor& out,
                                  torch::stable::Tensor const& input,
                                  torch::stable::Tensor& scales,
                                  int64_t group_size,
                                  std::optional<torch::stable::Tensor> scale_ub,
                                  bool is_scale_transposed);
```

评论区精华

- 最低 torch 版本：janeyx99 提议提升至 2.11，Harry-Chen 确认可行但目标仍是 2.12。
- Use-after-free 风险：depthfirst-app[bot] 指出 2.10 fallback 缺失 deleter；版本提升后消除。
- is_pinned 用法：janeyx99 建议使用 torch_call_dispatcher，作者已采纳。
- ROCm构建断裂：tjtanaa报告ROCm构建失败，Harry-Chen承诺后续修复（见#44648）。

风险与影响

- ROCm 兼容性：提升版本导致 ROCm 构建立即断裂，需紧急修补。
- 内存安全：若 torch 2.10 运行时环境仍使用旧 fallback，存在 use-after-free 风险，但 PR 已移除该路径。
- 头文件路径：大量文件包含路径更新，漏改可能导致编译错误，需仔细 review。

关联脉络

本 PR 是 stable ABI 迁移系列的第 10 步，前序阶段将 cache ops 和 layernorm 等内核已迁移至 _C_stable_libtorch。引发的 ROCm 回归在后续 PR #44648 中修复。未来计划继续迁移 _moe_C、quantization/marlin 等模块，并最终将 torch 底线推至 2.12。