PR #40845 完整报告

vllm-project/vllm

[BE][Torch 2.12] Remove workaround code for fixed cublas issue

合并时间：2026-04-29 12:07

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/40845

执行摘要

- 一句话：移除 B200 批次不变性 workaround，统一 SM100 与 SM90 路径
- 推荐动作：值得精读。PR 展示了如何在上游修复后干净地剥离临时 workaround，同时注意了交叉平台安全（is_cuda() 保护）。是学习 vLLM 如何处理 GPU 架构差异和 PyTorch 版本兼容性的好例子。

功能与动机

在 PyTorch 2.12 环境下，cuBLASLt on B200 已是 batch-invariant 的（参考 pytorch/pytorch#181248）。原 workaround 是 torch-2.9 时代引入的，用于规避 B200 在 bs=1 时选择非 batch-invariant GEMV 路径的问题。现在不再需要，故移除，并统一 SM90 与 SM100 的代码路径。

实现拆解

1. 移除 SM100 特判：将原来的 if is_device_capability_family(100) or is_device_capability_family(80): 改为只检查 capability 80。这样 SM100 不再进入 Triton 持久矩阵乘法覆盖的分支。
2. 统一 cuBLAS workspace 路径：SM100 与 SM90 一样进入 else 分支，设置 CUBLAS_WORKSPACE_CONFIG 和 CUBLASLT_WORKSPACE_SIZE 以禁用 split-k 引起的不确定性。
3. 修复 get_max_shared_memory_bytes 调用范围：将原来只在 SM100/80 分支中查询共享内存大小的逻辑独立出来，并加上 is_cuda() 检查，避免在非 CUDA 平台上误调用。
4. 更新注释：将注释从解释历史 workaround 改为说明当前分支的选择理由（Ampere 需要 Triton 覆盖，而 Hopper/Blackwell 只需禁用 split-k）。
5. 测试验证：在 B200 + torch 2.12 + triton 3.7.0 上运行 tests/v1/determinism/test_batch_invariance.py 全部 9 个用例通过。

关键文件：

- vllm/model_executor/layers/batch_invariant.py（模块 模型执行器；类别 source；类型 data-contract）：核心修改文件，修改了 enable_batch_invariant_mode 函数的条件分支，移除了 SM100 特殊路径，并调整了共享内存查询的保护逻辑。

关键符号：enable_batch_invariant_mode

关键源码片段

vllm/model_executor/layers/batch_invariant.py

核心修改文件，修改了 enable_batch_invariant_mode 函数的条件分支，移除了 SM100 特殊路径，并调整了共享内存查询的保护逻辑。

```
def enable_batch_invariant_mode():
    # ... 全局变量声明 ...

    if _batch_invariant_MODE:
        return

    _batch_invariant_MODE = True
    _batch_invariant_LIB = torch.library.Library("aten", "IMPL")

    if current_platform.is_device_capability_family(80):
        # SM80 (Ampere) 不能依赖 cuBLASLt 的确定性，因此安装 Triton 持久矩阵乘法覆盖。
        _batch_invariant_LIB.impl("aten::mm", mm_batch_invariant, "CUDA")
        _batch_invariant_LIB.impl("aten::addmm", addmm_batch_invariant, "CUDA")
        _batch_invariant_LIB.impl("aten::matmul", matmul_batch_invariant, "CUDA")
        _batch_invariant_LIB.impl("aten::linear", linear_batch_invariant, "CUDA")
    else:
        # Hopper (SM90) 和 Blackwell (SM100)：唯一的不确定性来源是 split-k，
        # 通过 cuBLAS workspace 配置禁用。
        _original_cublas_workspace_cfg = os.environ.get("CUBLAS_WORKSPACE_CONFIG", None)
        _original_cublaslt_workspace_size = os.environ.get("CUBLASLT_WORKSPACE_SIZE", None)
        os.environ["CUBLAS_WORKSPACE_CONFIG"] = ":16:8"
        os.environ["CUBLASLT_WORKSPACE_SIZE"] = "1"

    # Triton bmm 和持久矩阵乘法内核读取此值来确定 FP16 N-tile 大小。
    # 在所有 CUDA 平台上无条件设置，因为 bmm 被覆盖。
    if current_platform.is_cuda():
        _fp16_block_size_n = 256 if get_max_shared_memory_bytes() > 106496 else 128
```

评论区精华

- yewentao256: 指出 get_max_shared_memory_bytes 只对 CUDA 有效，但当前修改可能让 CPU worker 也进入该路径，存在风险。
  - Lucaskabela: 回复已 gated 到 CUDA cap 下。
- yewentao256: 建议简化 else 分支注释，去掉“previous update”的提法。
  - Lucaskabela: 采纳建议并更新注释。
  - get_max_shared_memory_bytes 调用范围 (correctness): PR 作者意识到问题，随后在代码中加入了 if current_platform.is_cuda(): 保护。
  - else 分支注释简化 (style): PR 作者采纳建议，简化了注释，只描述当前逻辑。

风险与影响

- 风险：
  - 上游依赖变更风险：本修改依赖 PyTorch 2.12 的 cuBLAS 修复。如果用户在低于 2.12 的环境运行，B200 可能丢失 batch-invariant 保证。但 PR 面向 torch 2.12 升级，合理假设目标环境已升级。
  - 回归风险：B200 从 Triton 路径切换到 cuBLASLt 路径，尽管本地测试通过，但 CI 尚未在曾经失败的 B200 runner 上验证。不过 Triton 路径的移除减少了一个容易出现分歧的分支，长期看降低了回归可能性。
  - 非 CUDA 平台风险：get_max_shared_memory_bytes 调用已用 is_cuda() 保护，避免在 CPU 等平台上调用出错。
  - 配置覆盖遗漏：原先 SM100 分支不设置 cuBLAS workspace 环境变量，现在设置；如果原流程依赖了省略这些变量的行为，可能有影响。但理论上配置只影响 split-k 行为，与 batch-invariance 目标一致。
- 影响：
  - 用户影响：B200 用户不再经过 Triton 持久矩阵乘法内核，转而使用 cuBLASLt，性能与确定性应与 H100 一致。所有用户不再需要为 B200 单独设置。
  - 系统影响：统一了 SM90 和 SM100 在 batch-invariance 模式下的行为，减少了平台特有的代码路径，便于未来维护。
  - 团队影响：移除了一段带有历史背景的 workaround 代码，降低了阅读负担。后续若 PyTorch 有类似变更，也可参考此 PR 的处理方式。
  - 风险标记：上游依赖变更，B200 特定路径修改，平台条件收紧

关联脉络

- 暂无明显关联 PR