PR #50467 完整报告

vllm-project/vllm

[Hardware][AMD][Kernel][CI][Bugfix] Fix ROCm DeepEP FP8 max

合并时间：2026-07-31 10:37

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/50467

执行摘要

- 一句话：修复 ROCm DeepEP FP8 max 并收紧测试容差
- 推荐动作：该 PR 值得关注，因为修复了 FP8 最大值错误并保持了测试的严格性。设计决策值得借鉴：优先修复根因而非放松测试。

功能与动机

PR body 指出：在 FP8 e4m3fn-based 平台（gfx950）上，低延迟 FP8 内核因缺少宏保护而使用了错误的 e4m3fnuz 最大值 240.0 而非 448.0；在 e4m3fnuz-based 平台（gfx942）上，vLLM 为了与 e4m3fn 数学对齐（x2 缩放并刷新负零）使用了调整后的最大值 224.0。上游 DeepEP PR#17 已修复此问题，通过命令行参数支持可插拔的 e4m3fnuz 最大值。该 PR 旨在应用上游修复，而非像 PR#50145 那样放宽测试容差。

实现拆解

1. 更新 Dockerfile.rocm 中 DeepEP 的构建版本：将 DEEPEP_BRANCH 从 a9ea9774 更新为 0f63d3e9，该版本包含 ROCm/DeepEP#17 的修复。
2. 在 DeepEP 构建命令中添加 --rocm-gfx942-fp8-fnuz-max 224 参数，以对齐 vLLM 在 fp8_utils.py 中的 clamp 值（E4M3FNUZ 为 224）。
3. 在 test_deepep_moe.py 中，移除了针对 fp8_fnuz 平台的特殊容差放宽逻辑（不再使用 check_accuracy 的宽松容差，而是统一使用 torch.testing.assert_close 的严格容差 6e-2）。同时移除了不再使用的 check_accuracy 导入。

关键文件：

- docker/Dockerfile.rocm（模块 部署脚本；类别 infra；类型 infrastructure）：更新 DeepEP 分支并添加 FP8 最大值参数，是修复的核心部署变更。
- tests/kernels/moe/test_deepep_moe.py（模块 MoE 内核测试；类别 test；类型 test-coverage；符号 assert_deepep_close）：移除特殊容差放宽，恢复严格数值校验，并清理未使用的导入。

关键符号：assert_deepep_close

关键源码片段

docker/Dockerfile.rocm

更新 DeepEP 分支并添加 FP8 最大值参数，是修复的核心部署变更。

```
# DeepEP build stage - depends on ROCShmem, builds the HIP kernel wheel.
FROM build_rocshmem AS build_deepep
ARG DEEPEP_BRANCH="0f63d3e9" # 升级到包含 FP8 最大值修复的版本（ROCm/DeepEP#17）
ARG DEEPEP_REPO="https://github.com/ROCm/DeepEP.git"
ARG DEEPEP_NIC="cx7"

# 构建 DeepEP wheel。DeepEP 在 ROCSHMEM_DIR 查找 rocshmem。
# DeepEP 仅支持 gfx942 和 gfx950，因此默认列表避免 gfx90a。
# 我们传递 --rocm-gfx942-fp8-fnuz-max 224 使 DeepEP dispatch FP8 量化边界与 vLLM 的 clamp 对齐
#  （见 vllm/model_executor/layers/quantization/utils/fp8_utils.py）：E4M3FNUZ 为 224。
RUN --mount=type=cache,target=/root/.cache/ccache \
    export PYTORCH_ROCM_ARCH="gfx942;gfx950" \
    && git clone ${DEEPEP_REPO} \
    && cd DeepEP \
    && git checkout ${DEEPEP_BRANCH} \
    && LDFLAGS="-fuse-ld=mold" MAX_JOBS="${MAX_JOBS:-$(nproc)}" \
       python3 setup.py --variant rocm --rocm-explicit-ctx --nic ${DEEPEP_NIC} --rocm-gfx942-fp8-fnuz-max 224 bdist_wheel --dist-dir=/app/deep_install
```

tests/kernels/moe/test_deepep_moe.py

移除特殊容差放宽，恢复严格数值校验，并清理未使用的导入。

```
# tests/kernels/moe/test_deepep_moe.py

def assert_deepep_close(
    expected: torch.Tensor,
    actual: torch.Tensor,
    k: int,
    use_fp8_dispatch: bool,
) -> None:
    # 统一使用严格容差。之前在此处针对 fp8_fnuz 平台放宽了容差（1.5e-1），
    # 但那只是掩盖了 FP8 最大值错误的 bug；现在根因已修复，恢复严格校验。
    torch.testing.assert_close(
        expected,
        actual,
        atol=6e-2,
        rtol=6e-2,
    )
```

评论区精华

审核过程中只有一条来自 AndreasKaratzas 的批准评论，声明 LGTM。Claude bot 评论自动审核因 fork 而禁用。无其他讨论。

- 暂无高价值评论线程

风险与影响

- 风险：风险较低，但需注意：测试容差收紧后，若 DeepEP 在提交 0f63d3e9 中未完全修复问题，可能导致测试在新平台（如 gfx950）上失败。此外，该修改仅影响 ROCm 容器构建，但 DeepEP 版本更新可能引入其他行为变化。
- 影响：影响范围限定在 ROCm 平台（gfx942 和 gfx950）上的 DeepEP FP8 MoE 内核。用户将获得更准确的 FP8 量化边界，可能改善模型精度。对于直接使用容器或依赖 DeepEP 的用户有影响，但未改动核心 vLLM 代码。
- 风险标记：测试容差收紧，依赖版本更新

关联脉络

- PR #50145 [ROCm] Loosen DeepEP FP8 dispatch tolerances: PR body 明确提到该 PR 不是 PR#50145 的重复，后者错误地放宽了容差而非修复根因。