PR #27111 完整报告

sgl-project/sglang

[AMD] Minimax M25：FP8 block-scale GEMM dispatch for ROCm 7.0 on gfx950

合并时间：2026-06-04 15:41

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/27111

执行摘要

- 一句话：为 ROCm 7.0/gfx950 新增 CK fp8 块量化 GEMM 回退
- 推荐动作：值得合入。变更精炼、风险低，性能收益明确。关注后续 ROCm 7.2+ 上 bpreshuffle 路径与 CK 路径的调度优先级可再评估。

功能与动机

ROCm 7.0 上 gfx950 的 hipcc 编译 bug 导致 gemm_a8w8_blockscale_bpreshuffle 内核不可用，通用 Triton 路径对 MiniMax-M2.5 等多样 GEMM 形状的 MoE 模型性能较差。PR 作者报告 MiniMax-M2.5 TP=4 FP8 下吞吐量提升 8-16%，GSM8K 准确率 94.1% 不变。

实现拆解

1. 新增 CK 内核导入：在 fp8_utils.py 中从 aiter 导入 gemm_a8w8_blockscale 并重命名为 ck_gemm_a8w8_blockscale。
2. 调整调度逻辑：在 aiter_w8a8_block_fp8_linear 中增加分支 elif _use_aiter_gfx95，当启用 aiter 且为 gfx950 时，即使 hipcc 版本 <7.2.0 也可走 CK 回退。
3. 条件化 scale 转置：仅当 _use_aiter_bpreshuffle_gfx95 且非 Triton 时才进行 scale 转置，CK 内核不需要。
4. 算子选择三路分支：优先 Triton（调优 shape），其次 bpreshuffle（ROCm ≥7.2），其余走 CK gemm_a8w8_blockscale。

关键文件：

- python/sglang/srt/layers/quantization/fp8_utils.py（模块 量化层；类别 source；类型 dependency-wiring）：唯一变更文件，包含内核导入、调度分支和算子选择逻辑的重构。

关键符号：aiter_w8a8_block_fp8_linear

关键源码片段

python/sglang/srt/layers/quantization/fp8_utils.py

唯一变更文件，包含内核导入、调度分支和算子选择逻辑的重构。

```
# python/sglang/srt/layers/quantization/fp8_utils.py

# 新增 CK 内核导入
```

```
if _use_aiter:
    import aiter
    from aiter import gemm_a8w8_blockscale as ck_gemm_a8w8_blockscale # ← 新增
    from aiter import (
        gemm_a8w8_blockscale_bpreshuffle,
        gemm_a8w8_bpreshuffle,
        get_hip_quant,
    )
    ...

def aiter_w8a8_block_fp8_linear(...):
    n, k = weight.shape

    # 调度分支：三路选择
    if _use_aiter_bpreshuffle_gfx95:
        use_triton = use_aiter_triton_gemm_w8a8_tuned_gfx950(n, k)
    elif _use_aiter_gfx95: # ← 新增：ROCm 7.0 gfx950 回退
        use_triton = use_aiter_triton_gemm_w8a8_tuned_gfx950(n, k)
    else:
        use_triton = True

    # scale 转置仅 bpreshuffle 路径需要
    if input_scale is not None:
        q_input = input_2d
        x_scale = input_scale
        if _use_aiter_bpreshuffle_gfx95 and not use_triton:
            x_scale = x_scale.transpose(-1, -2).contiguous().view(*x_scale.shape)
    else:
        q_input, x_scale = aiter_per1x128_quant(
            input_2d,
            quant_dtype=aiter.dtypes.fp8,
            transpose_scale=(_use_aiter_bpreshuffle_gfx95 and not use_triton),
        )

    # 算子选择：Triton → bpreshuffle → CK fallback
    if use_triton:
        gemm_a8w8_blockscale_op = triton_gemm_a8w8_blockscale
    elif _use_aiter_bpreshuffle_gfx95:
        gemm_a8w8_blockscale_op = gemm_a8w8_blockscale_bpreshuffle
    else:
        gemm_a8w8_blockscale_op = ck_gemm_a8w8_blockscale # ← 新增 CK 回退
```

评论区精华

无 review 评论。合并者 HaiShaw 确认 CI 失败由另一已合入 PR #27163 修复，非本 PR 引入。

- 暂无高价值评论线程

风险与影响

- 风险：低风险。变更仅限于 gfx950 且 _use_aiter 为真的条件路径，不影响 CUDA 或其他硬件。CK 内核已存在于 aiter 库，无需额外依赖。scale 转置条件精确限定，不影响已有逻辑。
- 影响：正面影响：显著提升 MiniMax-M2.5 等 MoE 模型在 MI355X 上的吞吐量（8-16%），无需用户配置。不影响其他硬件平台或已有调优 shape。
- 风险标记：仅影响 AMD gfx950 平台，依赖 aiter 库新导出符号

关联脉络

- PR #23319 [AMD] hipcc miscompilation on ROCm 7.0: PR body 引用的已知 bug，导致 bpreshuffle 内核在 ROCm 7.0 上不可用。
- PR #27163 [AMD] Fix CI failures: 合并者提及此 PR 修复了 CI 失败，与本 PR 无关但确保 CI 绿色。