PR #33090 完整报告

sgl-project/sglang

[AMD][Fix] Restore aiter-padded MoE weight dims for serialized checkpoints

合并时间：2026-08-01 13:09

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/33090

执行摘要

- 一句话：恢复序列化 MXFP4 MoE 权重 aiter 对齐维度
- 推荐动作：值得精读：PR body 的 bisect 过程和根因分析是排查此类静默精度回归的范例。建议合并后立即跟进补充形状一致性断言或序列化 + aiter padding 路径的单测，并关注 fxmarty-amd 提到的 #21097 与 #28291 冲突解决教训。

功能与动机

PR body 指出 dcd9014f15（#28291）静默破坏了 ROCm 上序列化 MXFP4 Quark MoE checkpoint 的准确率：输出仍流畅但模型丢失知识且频繁无法终止，单次 smoke prompt 无法可靠捕捉。git bisect 在 v0.5.16 分支点与 main 之间的 422 个提交中定位首个坏提交 dcd9014f15，父提交正常；触发条件是 moe_intermediate_size / TP < 256。

实现拆解

1. 定位根因：作者对 422 个提交做 bisect，确认 dcd9014f15 为坏提交。对比发现权重与 scales 的形状各自自洽但互相矛盾——权重用原始维度（TP8 下 w13 中间维 256、w2 中间维 64），scales 用填充维度（512/128），aiter 推断 inter_dim 从 256 变成 128，导致准确率崩溃并丢失 tuned 内核配置。
2. 修改源码：在 python/sglang/srt/layers/quantization/quark/schemes/quark_w4a4_mxfp4_moe.py 的 QuarkW4A4MXFp4MoE.create_weights 中，将 is_checkpoint_mxfp4_serialized 分支的 w13_shape 中间维度从 2 * intermediate_size_per_partition 改回 w13_up_dim，w2_shape 中间维度从 intermediate_size_per_partition // 2 改回 w2_down_dim，并添加注释解释为何序列化分支必须保持 padded 维度。
3. 保持在线量化分支不变：非序列化分支继续使用 #28291 引入的形状，避免影响在线量化路径。
4. 验证：8x MI355（gfx950）TP8 下 GSM8K 从 main 的 0.54 恢复到 0.98，inter_dim 从 128 回到 256；内核选择日志恢复 112 个 tuned FlyDSL 配置、0 个 fallback；定性测试 Kalevipoeg 从拒绝回答 / 跑满 max_tokens 恢复为正确回答。
5. 测试配套：本 PR 未加单测。PR 说明现有 suite 无法捕捉此回归——权重与 scale 各自自洽、仅关系错误且只在 TP ≥ 8 出现；作者建议后续在 test/registered/quant/test_quark_mxfp4.py 中补充序列化 + aiter padding 路径或添加 shape assertion。

关键文件：

- python/sglang/srt/layers/quantization/quark/schemes/quark_w4a4_mxfp4_moe.py（模块 量化层；类别 source；类型 core-logic；符号 create_weights, QuarkW4A4MXFp4MoE）：修复根因所在的单一文件：QuarkW4A4MXFp4MoE.create_weights 中序列化分支权重维度与 scales 解同步。

关键符号：QuarkW4A4MXFp4MoE.create_weights

关键源码片段

python/sglang/srt/layers/quantization/quark/schemes/quark_w4a4_mxfp4_moe.py

修复根因所在的单一文件：QuarkW4A4MXFp4MoE.create_weights 中序列化分支权重维度与 scales 解同步。

```
# WEIGHTS
# 序列化 MXFP4 必须保持 get_moe_weight_sizes() 返回的 aiter 对齐维度：
# 下方的 w13_weight_scale / w2_weight_scale 仍按 w13_up_dim / w2_down_dim 分配，
# 且 extra_weight_attrs 中的 weight_padded 标记也会传给 loader。
# 若在此分配原始未填充维度，权重与 block scale 会描述不同的中间层尺寸，
# 导致 TP >= 8 时（例如 Qwen3.5-397B-A17B-MXFP4，moe_intermediate_size / TP < 256）
# GSM8K 准确率从 0.99 崩溃到 0.54。
w13_shape = (
    num_experts,
    (
        w13_up_dim
        if self.is_checkpoint_mxfp4_serialized
        else 2 * intermediate_size_per_partition
    ),
    hidden_size // 2 if self.is_checkpoint_mxfp4_serialized else hidden_size,
)
w13_weight = torch.nn.Parameter(
    torch.empty(w13_shape, dtype=weight_dtype, device=weight_device),
    requires_grad=False,
)
layer.register_parameter("w13_weight", w13_weight)
set_weight_attrs(w13_weight, extra_weight_attrs)

w2_shape = (
    num_experts,
    hidden_size,
    (
        w2_down_dim
        if self.is_checkpoint_mxfp4_serialized
        else intermediate_size_per_partition
    ),
)
w2_weight = torch.nn.Parameter(
```

```
        torch.empty(w2_shape, dtype=weight_dtype, device=weight_device),
        requires_grad=False,
    )
    layer.register_parameter("w2_weight", w2_weight)
    set_weight_attrs(w2_weight, extra_weight_attrs)

    # WEIGHT_SCALES
    # scales 始终按 get_moe_weight_sizes() 的填充维度分配,
    # 因此权重必须使用同一组维度，否则二者描述不同的中间层尺寸。
    extra_weight_attrs["weight_loader"] = original_weight_loader

    w13_weight_scale = torch.nn.Parameter(
        torch.ones(
            num_experts,
            w13_up_dim,
            hidden_size // OCP_MX_BLOCK_SIZE,
            dtype=params_dtype,
        ),
        requires_grad=False,
    )
    # w2 的 scale 形状类似，按 w2_down_dim 与 OCP_MX_BLOCK_SIZE 计算,
    # 这里仅截取核心部分以说明维度同步关系。
```

评论区精华

fxmarty-amd（#28291 作者）在批准时承认冲突解决不当："For context, #21097 was merged without unit tests capturing this failure, and its development collided with #18182 / #28291 where I obviously did not properly solve conflicts." yichiche 在 issue 评论中详细解释为何只有 TP ≥ 8 受影响：触发条件为 moe_intermediate_size / TP < 256，且与 attention backend 无关，是纯粹的 MoE 权重形状算术问题。HaiShaw 则请求 BowenBao 和 fxmarty-amd 在后续 PR 中补充单元测试覆盖这类回归。

- 修复方案是否正确（fxmarty-amd 确认） (design): 批准修复，确认是冲突解决不当导致。
- 为何只有 TP >= 8 受影响 (question): 确认根因是权重与 scales 维度解同步，非 attention 后端问题。
- 补充单元测试计划 (testing): 未在本 PR 内解决，留待后续 PR。

风险与影响

- 风险：回归风险：改动仅在序列化分支两处维度，但该路径无自动化测试覆盖，未来改动 get_moe_weight_sizes() 或 weight_padded 语义时容易再次解同步。性能影响：修复恢复了 tuned aiter FlyDSL 内核选择（日志从 0/112 变回 112/0），应恢复 #28291 之后损失的 MoE 吞吐，但本 PR 未做专门吞吐 sweep。兼容范围：仅作用于 is_checkpoint_mxfp4_serialized 分支，在线量化分支与非 AMD 平台不受影响。部署影响：需要重新加载序列化 checkpoint 才能生效。
- 影响：用户侧：修复 ROCm 上序列化 MXFP4 MoE 大模型（如 Qwen3.5-397B-A17B-MXFP4）在 TP ≥ 8 时的准确率崩溃，并恢复 tuned 内核带来的性

能。系统侧：改动极小，影响范围限定在 quark_w4a4_mxfp4_moe.py 一个文件。团队侧：提供了完整的 bisect 方法论与维度一致性检查思路，后续需补充 UT 防止回归。

- 风险标记：缺少测试覆盖，高 TP 回归风险，AMD 专属路径

关联脉络

- PR #28291 [AMD][MXFP4] Reland "Online MXFP4 quantization 2/N - FP8 to MXFP4 requantization on AMD GPUs": 引入回归的提交 dcd9014f15 所在 PR，本 PR 的直接修复对象。
- PR #18182 [AMD][MXFP4] Online MXFP4 quantization: 在线 MXFP4 量化原始 PR，被 revert 后由 #28291 reland，冲突解决不当导致本回归。
- PR #21097 Related MXFP4/MoE change referenced in review discussion: fxmarty-amd 在批准评论中点名：该 PR 无 UT 合并，并与 #18182/#28291 的开发冲突，是回归漏网的原因之一。