PR #27858 完整报告

sgl-project/sglang

[AMD] Fix the dsv4 performance of MoE issue.

合并时间：2026-06-11 16:06

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/27858

执行摘要

- 一句话：修复 AMD DSV4 MoE 性能：传递 intermediate_pad 参数
- 推荐动作：该 PR 值得精读，特别是性能调优方法：通过微观基准定位到内核参数缺失，再用端到端测试验证。设计上记录了填充量并在内核调用处传递，是一种清晰的解耦方式。Shuffle 函数的清理也提高了代码可维护性。

功能与动机

在 AMD MI355X/gfx950 上，aiter fused_moe 内核性能明显低于参考 ATOM 引擎。经过微观基准测试定位到 intermediate_pad 未正确设置（默认为 0），导致内核计算了填充的零值通道。PR body 提到："AITER FP4 MoE kernel requires the per-partition intermediate size to be 256-aligned, so during weight loading we pad it (…). But the pad amount was never propagated to fused_moe."

实现拆解

1. 记录填充量到 layer 属性：在 Fp8MoEMethod.process_weights_after_loading_block_quant 中，计算 padded_inter 后，将 layer.intermediate_pad = padded_inter - inter_per_part 和 layer.hidden_pad = 0 保存到 layer 上。
2. 传递 intermediate_pad 到 fused_moe：在 maybe_get_hip_aiter_quant_info 中，新增 hidden_pad 和 intermediate_pad 参数，通过 getattr(layer, ...) 读取并传入 AiterMoeQuantInfo，使得内核跳过填充计算。
3. 统一 FP4 权重 / 缩放洗牌：移除对 shuffle_scale_a16w4 / shuffle_weight_a16w4 的导入，改用通用 shuffle_scale / shuffle_weight，并引入 gu_intv 参数（来自环境变量 SGLANG_USE_AITER_MOE_GU_ITLV），使交错布局可配置。
4. 根据 review 建议将 gu_intv 改为局部变量：初始提交将 gu_intv 设为实例属性，审查要求改为局部变量，最终实现在 process_weights_after_loading_block_quant 中直接读取。
5. 数值等价性：填充通道权重 / 缩放为零，计算与否数值等效，GSM8K 精度测试确认无回归（0.970 vs 0.965，在运行间噪声范围内）。

关键文件：

- python/sglang/srt/layers/quantization/fp8.py（模块 量化层；类别 source；类型 core-logic；符号 process_weights_after_loading_block_quant, maybe_get_hip_aiter_quant_info）：所有性能修复和代码清理均在此文件：记录

intermediate_pad/hidden_pad 到 layer 属性、在 maybe_get_hip_aiter_quant_info 中传递给 AiterMoeQuantInfo、统一 shuffle 函数并引入 gu_intv 局部变量。

关键符号：process_weights_after_loading_block_quant, maybe_get_hip_aiter_quant_info

关键源码片段

python/sglang/srt/layers/quantization/fp8.py

所有性能修复和代码清理均在此文件：记录 intermediate_pad/hidden_pad 到 layer 属性、在 maybe_get_hip_aiter_quant_info 中传递给 AiterMoeQuantInfo、统一 shuffle 函数并引入 gu_intv 局部变量。

```
# python/sglang/srt/layers/quantization/fp8.py

# 在 process_weights_after_loading_block_quant 中记录填充量
fp4_k_align = 256
padded_inter = (
    (inter_per_part + fp4_k_align - 1) // fp4_k_align * fp4_k_align
)
# 记录填充量，告诉 fused_moe 实际中间大小
# aiter fused_moe 需要 intermediate_pad = padded - real # ATOM 传 128,
# SGLang 之前默认为 0 导致计算了填充区域
layer.intermediate_pad = padded_inter - inter_per_part # 例如 128
layer.hidden_pad = 0

if padded_inter != inter_per_part:
    # pad 权重逻辑不变 ...

# 在 maybe_get_hip_aiter_quant_info 中传递到 AiterMoeQuantInfo
AiterMoeQuantInfo(
    ...
    # 以下两个参数是关键修复：让内核跳过填充通道的计算
    hidden_pad=getattr(layer, "hidden_pad", 0),
    intermediate_pad=getattr(layer, "intermediate_pad", 0),
)
```

评论区精华

Review 评论仅一条：HaiShaw 要求将 gu_intv 从实例属性（self.gu_intv）改为 process_weights_after_loading_block_quant 内的局部变量，作者在第二个 commit 中已采纳。无其他争议。

- gu_intv 应设为局部变量而非实例属性 (design): 作者采纳建议，在第二个 commit 中将 gu_intv 改为函数内局部变量。

风险与影响

- 风险：该 PR 修改了量化核心逻辑，但变更集中且已验证：
  - 性能回归：无，正确传递 pad 后性能提升。

  - 精度回归：PR 提供了 GSM8K 对比数据，前后精度在运行噪声范围内，无实质回归。
  - 兼容性：默认行为不变（SGLANG_USE_AITER_MOE_GU_ITLV 默认 True，与之前硬编码交错一致）。
  - 风险：缺少独立测试覆盖，仅依赖作者提供的基准测试。如果环境变量或 aiter 版本变更可能受影响，但概率较低。
  - 影响：影响范围：仅影响 AMD GPU（MI355X/gfx950）上使用 aiter 后端的 DeepSeek-V4 FP4 MoE 路径。影响程度：显著——MoE 内核性能提升 18-19%，端到端解码吞吐提升 5-6%，TPOT/ITL 下降 6-8%。用户影响：AMD 用户无需配置变更即可获得性能提升；其他硬件平台无影响。
- 风险标记：核心路径变更，缺少测试覆盖

关联脉络

- PR #27747 fix: DSV4 BCG compress-prefill plan OOB on underfilled (tiny) prefill replay: 同为 DeepSeek-V4 相关修复，但领域不同（BCG vs MoE），无直接依赖。
- PR #27811 [AMD] Restore AMD piecewise CUDA graph support dropped by #23906: 同为 AMD 平台性能修复，但领域不同（CUDA graph vs MoE），无直接依赖。