PR #2014 完整报告

radixark/miles

fix: quantize non-interleaved DSA indexer wk

合并时间：2026-07-31 11:03

原文链接：http://prhub.com.cn/radixark/miles/pull/2014

执行摘要

- 一句话：按 indexer_rope_interleave 条件量化 DSA indexer wk
- 推荐动作：值得精读：它展示了对历史修复的回溯修正，以及基于下游引擎（SGLang）参数布局做条件化量化的设计模式。建议结合 #1928 一起阅读，理解 DSA indexer 在 interleaved/ 非 interleaved 下的差异；同时关注 indexer_rope_interleave 配置在模型转换链路中的传递可靠性。

功能与动机

PR body 明确说明：The Blackwell fix in #1928 removed both DSA indexer wk names from the FP8 quantization list unconditionally. That is correct for an interleaved/fused indexer... For non-interleaved indexers, SGLang instead keeps a standalone FP8 wk parameter and its scale. Sending only the BF16 weight makes the loader cast it into the FP8 parameter without updating the scale, corrupting the effective rollout weight. 该问题由 DeepSeek V3.2 FP8 权重更新检查暴露。

实现拆解

1. 回溯根因：确认 #1928 的无条件移除只对 interleaved/fused indexer 正确；非 interleaved 场景下需要同时量化权重与 scale。
2. 修改 quantizer_fp8.py：将固定参数列表改为 fp8_param_names，移除 self_attention.wk.weight 与 self_attention.indexer.linear_wk.weight，并通过 if not getattr(args, "indexer_rope_interleave", False) 条件决定是否追加这两个名称。默认 False 时恢复量化，True 时保持 bf16。
3. 同步修改 quantizer_mxfp8.py：第二个提交将同一条件逻辑应用到 MXFP8 路径，并补充注释说明 interleaved 场景下 loader 会误读 e8m0/mxfp8 scale，保持两个量化器行为一致。
4. 配置来源：indexer_rope_interleave 由 Miles 从 Hugging Face checkpoint 配置读取，DeepSeek V3.2 未定义（默认 False），GLM-5.2 显式设为 True。
5. 验证配套：仅运行 pre-commit、ruff、black、py_compile，无新增单元测试，依赖 run-ci-megatron CI 验证。

关键文件：

- miles/backends/megatron_utils/megatron_to_hf/processors/quantizer_fp8.py（模块 权重量化；类别 source；类型 core-logic；符号 quantize_params_fp8）：核心修复文件：

将 DSA indexer wk 的量化从无条件移除改为按 indexer_rope_interleave 条件控制，解决非 interleaved 场景下权重同步损坏问题。

- miles/backends/megatron_utils/megatron_to_hf/processors/quantizer_mxfp8.py（模块权重量化；类别 source；类型 core-logic；符号 quantize_params_mxfp8）：与 FP8 量化器同步修改，确保 MXFP8 路径也按相同条件处理 DSA indexer wk，避免两种量化格式行为不一致。

关键符号：quantize_params_fp8, quantize_params_mxfp8

关键源码片段

miles/backends/megatron_utils/megatron_to_hf/processors/quantizer_fp8.py

核心修复文件：将 DSA indexer wk 的量化从无条件移除改为按 indexer_rope_interleave 条件控制，解决非 interleaved 场景下权重同步损坏问题。

```
# 基础 FP8 量化参数清单：涵盖 MLA、DSA indexer 的 wq_b、linear attention、DeepSeek V4 等。
# 注意 wk 的两个名称不在此基础清单中，而是按 indexer_rope_interleave 条件追加。
fp8_param_names = [
    "self_attention.linear_proj.weight",
    "self_attention.linear_qkv.weight",
    "mlp.linear_fc1.weight",
    "mlp.linear_fc2.weight",
    # mla
    "self_attention.linear_q_proj.weight",
    "self_attention.linear_q_down_proj.weight",
    "self_attention.linear_q_up_proj.weight",
    "self_attention.linear_kv_down_proj.weight",
    "self_attention.linear_kv_up_proj.weight",
    # DSA indexer
    "self_attention.wq_b.weight",
    # linear attention
    "self_attention.linear_attn.in_proj_qkv.weight",
    "self_attention.linear_attn.in_proj_z.weight",
    "self_attention.linear_attn.out_proj.weight",
    # DeepSeek V4 attention
    "self_attention.wq_a.weight",
    "self_attention.wkv.weight",
    "self_attention.wo_b.weight",
    "self_attention.indexer.linear_wq_b.weight",
]
if not getattr(args, "indexer_rope_interleave", False):
    # 非 interleaved indexer 在 SGLang 中把 wk 保存为独立的 FP8 参数与 scale，
    # 因此必须连同 scale 一起量化，否则 loader 只 cast 权重而不更新 scale，
    # 会损坏有效的 rollout 权重。interleaved 时 wk 被融合进 bf16 的
    # wk_weights_proj，量化反而让 loader 无法正确读取 scale。
    fp8_param_names.extend(
        [
```

```
            "self_attention.wk.weight",
            "self_attention.indexer.linear_wk.weight",
        ]
    )

if rest in fp8_param_names:
    quantize_named_params = []
    for converted_name, param in converted_named_params:
        quantize_named_params.extend(_quantize_param(args, converted_name, param, weight_
        block_size))
    return quantize_named_params
```

miles/backends/megatron_utils/megatron_to_hf/processors/quantizer_mxfp8.py

与 FP8 量化器同步修改，确保 MXFP8 路径也按相同条件处理 DSA indexer wk，避免两种量化格式行为不一致。

```
mxfp8_param_names = [
    "self_attention.linear_proj.weight",
    "self_attention.linear_qkv.weight",
    "mlp.linear_fc1.weight",
    "mlp.linear_fc2.weight",
    # mla
    "self_attention.linear_q_proj.weight",
    "self_attention.linear_q_down_proj.weight",
    "self_attention.linear_q_up_proj.weight",
    "self_attention.linear_kv_down_proj.weight",
    "self_attention.linear_kv_up_proj.weight",
    "self_attention.wq_b.weight",
    # DeepSeek V4 attention
    "self_attention.wq_a.weight",
    "self_attention.wkv.weight",
    "self_attention.wo_b.weight",
    "self_attention.indexer.linear_wq_b.weight",
]
if not getattr(args, "indexer_rope_interleave", False):
    # 非 interleaved indexer 将 wk 保留为独立的量化参数；interleaved 时
    # wk 融合进 bf16 的 wk_weights_proj，量化产生的 uint8 e8m0 mxfp8 scale
    # 会被 loader 误读成整数，因此只有非 interleaved 才需要追加 wk。
    mxfp8_param_names.extend(
        [
            "self_attention.wk.weight",
            "self_attention.indexer.linear_wk.weight",
        ]
    )

if rest in mxfp8_param_names:
    quantize_named_params = []
    for converted_name, param in converted_named_params:
```

```
            quantize_named_params.extend(_quantize_param(converted_name, param))
    return quantize_named_params
```

评论区精华

无实质性 review 讨论；yueming-yuan 直接 approve，未提出异议。注意 gemini-code-assist[bot] 的评论仅为通知其代码审查活动已终止，不代表有效审查意见。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. 回归风险：若某 interleaved 模型的调用路径未传入 indexer_rope_interleave=True，getattr 默认 False 会错误量化 wk，重新引入 #1928 修复的问题。需确认所有转换入口都正确传递该配置。
  2. 行为一致性风险：FP8 与 MXFP8 两个量化器必须保持同条件逻辑，未来若单独修改可能造成同一模型不同量化格式行为分叉。
  3. 测试缺口：本 PR 无相关测试文件，interleaved/ 非 interleaved 两个分支未被自动化用例覆盖，回归风险较高。
  4. 兼容性：对依赖旧行为的已有 FP8 checkpoint 或在线更新脚本，行为变化可能影响部署，但这是修复预期。 - 影响：影响 DeepSeek V3.2/V4 风格与 GLM-5.2 等带 DSA indexer 的模型，在 FP8/MXFP8 在线权重更新（rollout 权重同步）场景下。对 interleaved 模型无行为变化（wk 仍保持 BF16）；对非 interleaved 模型恢复量化路径，确保 scale 与权重一起更新。改动面小（2 个文件，+26/-6），但涉及 megatron_to_hf 转换与 rollout 同步的关键链路。 - 风险标记：依赖配置字段传递，缺少测试覆盖，条件逻辑分叉风险

关联脉络

- PR #1928 [fix] DSA indexer on Blackwell: send the DSA indexer wk unquantized: 本 PR 是对 #1928 的回溯修正：#1928 无条件将 wk 移出 FP8 量化列表，本 PR 改为按 indexer_rope_interleave 条件决定是否量化，解决了其未覆盖的非 interleaved 场景。