PR #35630 完整报告

sgl-project/sglang

[AMD] Enable Mori-EP on kimi-k3

合并时间：2026-08-25 16:48

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/35630

执行摘要

- 一句话：Kimi-K3 支持 MoRI EP 后端，AITER 路径提前返回
- 推荐动作：值得精读。这个 PR 是理解 SGLang MoE 后端扩展模式的很好样例：kimi_k3.py 的 _ep_a2a / _sp_moe 谓词展示了 backend 能力抽象如何决定模型的执行布局，mxfp4.py 的 _apply_aiter() 提取则体现了“按 dispatch 格式契约在统一入口分流”的设计。建议阅读时重点看 DispatchOutputChecker.format_is_deepep() 与 use_deep_gemm 分支的对称关系，并留意后续是否有针对 MoRI 的专项测试补充。

功能与动机

PR body 的动机是 Enable Mori EP for Kimi-K3。提交信息进一步说明根因：models/kimi_k3.py 的 _ep_a2a 和 _sp_moe 只识别 megamoe / DeepEP，导致 --moe-a2a-backend mori 回退到普通 TP 布局（shared experts 保持 TP-sharded、MoE 区域保留 DP gather），而不是跑在 SP-MoE token shard 上；mxfp4.py 的 apply() 会解包 .topk_output，而 DeepEP 家族（MoRI 在内）的 dispatch 输出没有该字段，需要在解包前交给 AITER runner。

实现拆解

本 PR 的变更入口是 Kimi-K3 的 MoE 后端选择逻辑与 MXFP4 量化入口，共 2 个文件：

1. 模型层：扩展 EP-a2a 后端识别（python/sglang/srt/models/kimi_k3.py） 在模型初始化中，_ep_a2a 与 _sp_moe 两个谓词的判断条件里追加 _a2a_backend.is_mori()，并同步更新相关注释。这样 --moe-a2a-backend mori 不再回退到普通 TP 布局，而是与 megamoe / DeepEP 一样进入 SP-MoE 布局：shared experts 按 tp1 复制、o_proj 延迟注意力 TP reduce 等下游分支都会被激活。_ep_a2a 决定 _shared_experts_tp1、_shared_experts_attn_tp_comm 与 MoE 区域的数据流，_sp_moe 决定 o_proj 的 reduce-scatter 时机，因此该谓词变更会连带改变整条 MoE 前向执行布局。
2. 量化层：AITER路径提取与提前分流（python/sglang/srt/layers/quantization/mxfp4.py） 在 apply() 中新增 DispatchOutputChecker.format_is_deepep() 检查，当启用 AITER 且 dispatch 输出属于 DeepEP 系列格式时，在解包 .topk_output 之前直接返回 _apply_aiter() 的结果；把原有 if _use_aiter: 分支的函数体原样提取为新的 _apply_aiter()，原分支改为委托调用。原因是 MoRI 属于 DeepEP 家族，dispatch 输出直接携带 topk_ids / topk_weights、没有 .topk_output 可解包，与 use_deep_gemm 分支面临同样的契约约束，所以必须在标准解包之前路由走。提取后 AITER 路径可由两个调用点（标准

路径末尾与新 DeepEP 提前返回）共同复用，避免重复代码；同时保证 deepep_normal / deepep_ll / MoRI 格式都走 AITER runner。

3. 死代码清理 依据 review 意见，移除 kimi_k3.py 中已废弃的 self._moe_front_needs_contiguouts 字段（main 分支已改用 self._moe_front_needs_dense_bf16）。
4. 测试与验证配套 本次未新增单元测试；仅在 PR 描述中给出 gsm8k 准确度 0.951 与完整的启动命令（SGLANG_USE_AITER=1、AITER_SITUV2_A8W4=1、SGLANG_MORI_NUM_MAX_DISPATCH_TOKENS_PER_RANK=2048 等）。CI 中 AMD ROCm 7.2 与 Extra 任务标注为失败，需关注。

关键文件：

- python/sglang/srt/layers/quantization/mxfp4.py（模块 量化层；类别 source；类型 core-logic；符号 apply, _apply_aiter）：核心量化入口：新增 DispatchOutputChecker.format_is_deepep() 提前分流，并提取 _apply_aiter()，是本次改动的主体。
- python/sglang/srt/models/kimi_k3.py（模块 K3 模型；类别 source；类型 data-contract；符号 _ep_a2a, _sp_moe）：模型层 backend 谓词扩展：_ep_a2a / _sp_moe 纳入 is_mori()，使 MoRI 走 SP-MoE 布局；同时清理死代码。

关键符号：apply, _apply_aiter

关键源码片段

python/sglang/srt/models/kimi_k3.py

模型层 backend 谓词扩展：_ep_a2a / _sp_moe 纳入 is_mori()，使 MoRI 走 SP-MoE 布局；同时清理死代码。

```
# python/sglang/srt/models/kimi_k3.py
# EP a2a 后端（Megamoe / DeepEP / MoRI）会把每行 token 直接送到其
# experts，因此 MoE 区域可以消费本 rank 持有的任意行 —— SP-MoE token
# shard（attn_tp > 1）或 DP-local batch（DP attention）—— 每个全局
# token 恰好被 dispatch 一次，区域内部无需任何 DP gather 或 TP reduce。
_a2a_backend = get_moe_a2a_backend()
self._ep_a2a = (
    _a2a_backend.is_megamoe()
    or _a2a_backend.is_deepep()
    or _a2a_backend.is_ascend_fuseep()
    or _a2a_backend.is_mori()
)
```

评论区精华

核心 review 交锋集中在两点：

- Duyi-Wang 指出 kimi_k3.py 中 self._moe_front_needs_contiguouts 是死代码，与本次 PR 无关，main 分支已经改用 self._moe_front_needs_dense_bf16。作者 RolaoDenthu 回复致谢并确认已移除，该线程已解决。

- 设计层面：mxfp4.py 中新增的 DeepEP 提前返回分支与既有 use_deep_gemm 分支在注释中形成对称说明，核心权衡是“dispatch 格式契约不同，必须在解包 .topk_output 之前按格式分流”；AITER 与 DeepGEMM runner 各自通过注册的 pre/post-permute 或专用 quant_info 处理 raw topk 格式，避免在统一入口里做格式特判扩散。
- kimi_k3.py 中遗留死代码字段 (style): 作者 RolaoDenthu 回复感谢并已移除该字段。

风险与影响

- 风险：
  - 核心路径重构回归：mxfp4.py 的 apply() 是所有 MXFP4 MoE 的公共入口，AITER 分支从末尾提前到开头，任何 format_is_deepep() 判定边界（如新格式未登记）都可能导致行为偏差；_apply_aiter() 提取后需确保所有调用点语义一致。
  - 后端布局切换：kimi_k3.py 将 MoRI 纳入 _ep_a2a / _sp_moe 后，shared experts 复制、o_proj reduce-scatter 等路径从 TP 布局切换为 SP-MoE 布局，若 MoRI 的实际 dispatch 契约与 DeepEP 存在细微差异，可能引入精度或性能回归。
  - 测试覆盖缺失：PR 没有配套单元测试，仅依赖 gsm8k 0.951 与手工启动验证；AMD ROCm 7.2 与 CI Extra 任务标红，需要人工确认是否为环境问题。
  - 配置面广：启用 MoRI 依赖 SGLANG_USE_AITER、AITER_FLYDSL_FORCE、AITER_SITUV2_A8W4、SGLANG_MLA_DECODE_TUNE、MORI_SHMEM_MODE 等多个环境变量组合，排障面较大。
- 影响：
  - 用户：AMD 平台上运行 Kimi-K3 的用户新增 --moe-a2a-backend mori 可选路径，MoE 区域切换到 SP-MoE 布局并走 AITER runner，预期可获得更优的 EP 通信与计算重叠。
  - 系统：默认路径（非 MoRI、非 AITER）完全不受影响；只有显式启用 MoRI 或 DeepEP 系列格式并开启 AITER 时才进入新分支。
  - 团队：改动集中在两个文件，规模小，但 mxfp4.py 的 apply 入口是量化 MoE 的公共路径，后续新增 backend 时需继续遵守“格式契约先行拆分”的模式；缺少测试意味着回归风险靠人工兜底。
  - 风险标记：核心路径重构，缺少测试覆盖，AMD 专用路径，后端布局切换

关联脉络

- PR #35672 [AMD] Enable draft_extend CUDA graph for HIP DSA backend: 同为 AMD 后端能力使能类改动，且都围绕 K3/DeepSeek 家族的 HIP 路径。
- PR #33021 [AMD] Drop redundant FP8 bpreshuffle scale transpose via fused AR kernel: AMD + 量化层（layernorm/ 通信）的相邻优化，与本次 MXFP4 量化路径调整同属 AMD 量化执行链路。