

PR #53940 完整报告

vllm-project/vllm

[ROCm][Kimi-K3] Enable a4w4 flydsl kernels for KimiK3

合并时间: 2026-09-16 09:38

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/53940>

执行摘要

- 一句话: Kimi-K3 SiTUv2 MoE 默认改用 AITER a4w4 FlyDSL 内核
- 推荐动作: 值得精读。该 PR 是理解 vLLM 与外部 kernel 库 (AITER) 协作契约的典型案列: 环境变量版本兼容 (deprecated alias)、模块导入时一次性 env 同步 (`_sync_aiter_situv2_moe_env`)、workaround 作用域收窄, 都是可复用的设计手法。建议重点关注 `_sync_aiter_situv2_moe_env` 的幂等设计与 AITER 优先检查 A8W4 的 dispatch 细节, 以及 `oracle/mx_fp4.py` 中权重布局与 kernel 选择的耦合方式。若关注 ROCm 性能演进, 可结合 AITER #4463 的 tuned CSV 数据一起阅读。

功能与动机

PR body 指出 trace 对比显示 vLLM 在 Kimi-K3 / MI355X 上调度 `mfma_moe1_silu_mul_afp8_wfp4_fp8_gui_xcd4`, 而 ATOM 使用 `mfma_moe1_silu_mul_afp4_wfp4_fp4_situv2`; AITER 已通过 `AITER_SITUV2_A4W4` 支持 a4w4 (ROCm/aiter#4463), vLLM 之前却把 recipe toggle 接到了 a8w4。AITER #4463 的隔离内核计时显示 tuned CSV 就位后 a4w4 在所有 token 档位都领先 a16w4 (8k tokens 时 +48.8%), 相比 a8w4 在大 batch 上最高领先 18.3%。本 PR 让 a4w4 成为现有 SiTUv2 recipe 路径的默认行为, 并以 separated gate/up 布局匹配 ATOM 的 kernel 选择。

实现拆解

1. 环境变量契约调整 (`vllm/envs.py`): 新增布尔开关 `VLLM_ROCM_USE_AITER_MOE_SITUV2`, 解析逻辑与旧名 `VLLM_ROCM_USE_AITER_MOE_SITUV2_A8W4` 取或, 旧名作为 deprecated alias 保留。注释明确旧名不再选择 a8w4 内核, 且需要 AITER $\geq$ v0.1.20 与 tuned `kimik3_a4w4_*_fmoe.csv`, 否则 FlyDSL 走 heuristic fallback。这样现有 recipe 无需改动即可启用同一 a4w4 路径。
2. AITER 侧环境同步 (`vllm/_aiter_ops.py`): 新增 `_sync_aiter_situv2_moe_env()`, 在模块导入末尾和 `refresh_env_variables()` 中调用。启用时写 `AITER_SITUV2_A4W4=1` 并删除 `AITER_SITUV2_A8W4` (AITER 先检查 A8W4, 留着会仍选 `afp8`); 禁用时删除 `A4W4`, 避免残留影响默认 a16w4 路径。同时将类字段 `_MOE_SITUV2_A8W4` 与检查函数 `is_fused_moe_situv2_a8w4_enabled()` 重命名为 `_MOE_SITUV2 / is_fused_moe_situv2_enabled()`, 语义从 A8W4 变体变为 SiTUv2 FlyDSL 路径。

3. kernel 路由与权重布局 (fused_moe 两个文件) : experts/rocm_aiter_moe.py 中 SiTU 分支的 gate_mode 固定为 GateMode.SEPARATED.value, 删除原先基于 is_fused_moe_situv2_a8w4_enabled() 的 INTERLEAVE 分支; oracle/mxvp4.py 中 SiTU 分支 guinterleave 固定为 False。两处共同保证 gate/up 权重按 separated 布局 shuffle, 与 a4w4 FlyDSL kernel 及 ATOM 选择一致。
4. workaround 作用域收窄 (oracle/mxvp4.py) : AITER_BF16_FP8_MOE_BOUND=0 原来在 SiTU 分支无条件设置, 现移到非 SiTU 的 interleaved a16w4 路径 (DeepSeekV4 等)。因为 AITER 只在该分支读取此变量决定 bf16 还是 fp8 激活, SiTUV2 a4w4 separated 路径由 AITER_SITUV2_A4W4 独立选择 q_dtype_a, 不再需要它。
5. 测试与部署配套 (tests/kernels/moe/test_rocm_aiter_moe.py) : 新增 3 个用例锁定 env 同步契约——启用时写 AITER_SITUV2_A4W4 且清 A8W4、legacy alias 同样启用 a4w4、VLLM flag 禁用时清除 AITER_SITUV2_A4W4。部署侧由外部 recipes 仓库 vllm-project/recipes#954 同步更新 (移除 AITER_BF16_FP8_MOE_BOUND、改用新 env 名)。

关键文件:

- vllm/_aiter_ops.py (模块 AITER 适配; 类别 source; 类型 core-logic; 符号 _sync_aiter_situv2_moe_env, is_fused_moe_situv2_enabled, refresh_env_variables) : 核心逻辑所在: 新增 _sync_aiter_situv2_moe_env(), 把 vLLM 的 SiTUV2 开关镜像到 AITER 的 a4w4 dispatch 环境变量并清理 legacy A8W4 override; 同时将 SiTUV2 开关查询函数从 a8w4 语义重命名为通用语义。
- vllm/model_executor/layers/fused_moe/oracle/mxvp4.py (模块 权重转换; 类别 source; 类型 data-contract; 符号 convert_weight_to_mxvp4_moe_kernel_format) : 权重 shuffle 布局从按 a8w4 判断的 interleaved 改为固定 separated, 并把 AITER_BF16_FP8_MOE_BOUND workaround 移到 interleaved a16w4 路径, 是本次数据契约变更的核心执行点。
- vllm/model_executor/layers/fused_moe/experts/rocm_aiter_moe.py (模块 MoE 执行; 类别 source; 类型 data-contract; 符号 rocm_aiter_fused_experts) : gate_mode 选择逻辑从基于 a8w4 开关的 INTERLEAVE 分支改为固定 SEPARATED, 是 kernel 路由与权重布局配套的关键一环。
- vllm/envs.py (模块 环境变量; 类别 source; 类型 configuration; 符号 VLLM_ROCM_USE_AITER_MOE_SITUV2) : 定义新开关 VLLM_ROCM_USE_AITER_MOE_SITUV2, 并保留旧 A8W4 变量作为 deprecated alias, 注释明确了 AITER 版本下限与 tuned CSV 依赖。
- tests/kernels/moe/test_rocm_aiter_moe.py (模块 内核测试; 类别 test; 类型 test-coverage; 符号 test_aiter_moe_situv2_syncs_aiter_a4w4_env, test_aiter_moe_situv2_legacy_a8w4_alias_enables_a4w4, test_aiter_moe_situv2_clears_aiter_env_when_disabled) : 新增 3 个单测锁定 env 同步契约: a4w4 设置、legacy alias 启用、禁用清理, 防止后续改动破坏兼容行为。

关键符号: _sync_aiter_situv2_moe_env, is_fused_moe_situv2_enabled, refresh_env_variables, convert_weight_to_mxvp4_moe_kernel_format, rocm_aiter_fused_experts

关键源码片段

vllm/_aiter_ops.py

核心逻辑所在：新增 `_sync_aiter_situv2_moe_env()`，把 vLLM 的 SiTuv2 开关镜像到 AITER 的 `a4w4 dispatch` 环境变量并清理 legacy A8W4 override；同时将 SiTuv2 开关查询函数从 `a8w4` 语义重命名为通用语义。

```
# vllm/_aiter_ops.py
def _sync_aiter_situv2_moe_env() -> None:
    """把 vLLM 的 SiTuv2 MoE 开关镜像到 AITER 侧的 a4w4 调度环境变量。

    AITER 通过 AITER_SITUV2_A8W4 / AITER_SITUV2_A4W4 二选一决定 afp8 还是
    afp4 激活内核，且 A8W4 优先检查；因此启用本开关时必须显式清除遗留的
    A8W4 覆盖，否则会继续走旧的 afp8 (a8w4) 路径。
    """

    import os

    import vllm.envs as envs

    if envs.VLLM_ROCM_USE_AITER_MOE_SITUV2:
        # 启用：写入 a4w4 调度标志，并移除 legacy a8w4 覆盖
        os.environ["AITER_SITUV2_A4W4"] = "1"
        os.environ.pop("AITER_SITUV2_A8W4", None)
    else:
        # 禁用：清掉 a4w4 标志，避免残留影响默认 a16w4 路径
        os.environ.pop("AITER_SITUV2_A4W4", None)

    # 模块导入末尾执行一次（紧跟 register_ops_once()），保证进程启动即完成同步；
    # refresh_env_variables() 中也会再次调用，供测试与热更新场景复用
    rocm_aiter_ops.register_ops_once()
    _sync_aiter_situv2_moe_env()
```

vllm/model_executor/layers/fused_moe/oracle/mxfp4.py

权重 shuffle 布局从按 `a8w4` 判断的 `interleaved` 改为固定 `separated`，并把 AITER_BF16_FP8_MOE_BOUND workaround 移到 `interleaved a16w4` 路径，是本次数据契约变更的核心执行点。

```
# vllm/model_executor/layers/fused_moe/oracle/mxfp4.py (AITER_MXFP4_BF16 分支)
if activation == MoEActivation.SITU:
    from aiter.utility.fp4_utils import e8m0_shuffle

    from vllm._aiter_ops import rocm_aiter_ops

    fp4_dtype = torch.float4_e2m1fn_x2
    e8m0_dtype = torch.float8_e8m0fnu
    # SiTuv2 flydsl 使用 separated gate/up 布局 (a4w4 内核)，
    # 与 ATOM 的 kernel 选择保持一致；a8w4 时代才需要 interleaved
```

```

ginterleave = False
w13 = rocm_aiter_ops.shuffle_weight_a16w4(
    w13_weight.data.view(fp4_dtype), 16, ginterleave
)
w2 = rocm_aiter_ops.shuffle_weight_a16w4(
    w2_weight.data.view(fp4_dtype), 16, False
)
w13_scale_raw = w13_weight_scale.data.view(e8m0_dtype)
w2_scale_raw = w2_weight_scale.data.view(e8m0_dtype)
w13_scale = rocm_aiter_ops.shuffle_scale_a16w4(
    w13_scale_raw.view(-1, w13_scale_raw.shape[-1]),
    num_experts,
    ginterleave,
)
w2_scale = e8m0_shuffle(w2_scale_raw.view(-1, w2_scale_raw.shape[-1]))
w13.is_shuffled = True
w2.is_shuffled = True
return (w13, w2, w13_scale, w2_scale, w13_bias, w2_bias)

```

只有 interleaved a16w4 路径 (DeepSeekV4 等) 需要该 AITER 崩溃 workaround:
AITER 仅在此分支用 AITER_BF16_FP8_MOE_BOUND 决定 bf16 还是 fp8 激活,
SiTUV2 a4w4 走 separated, 由 AITER_SITUV2_A4W4 独立选择 q_dtype_a
import os

```
os.environ["AITER_BF16_FP8_MOE_BOUND"] = "0"
```

vllm/model_executor/layers/fused_moe/experts/rocm_aiter_moe.py

gate_mode 选择逻辑从基于 a8w4 开关的 INTERLEAVE 分支改为固定 SEPARATED, 是 kernel 路由与权重布局配套的关键一环。

```

# vllm/model_executor/layers/fused_moe/experts/rocm_aiter_moe.py (gate_mode 选择)
gate_mode = ""
if activation == MoEActivation.SITU:
    # SiTUV2 flydsl (VLLM_ROCM_USE_AITER_MOE_SITUV2=1) 使用 a4w4 fp4
    # 激活 + separated gate/up 权重 (AITER #4463) ; 默认 a16w4 SiTU
    # 也保持 separated, 因此不再需要 a8w4 时代的 INTERLEAVE 分支
    gate_mode = GateMode.SEPARATED.value
elif quant_config.use_mxfp4_w4a16:
    gate_mode = GateMode.INTERLEAVE.value
elif activation_interleave is not None:
    gate_mode = (
        GateMode.INTERLEAVE.value
        if activation_interleave
        else GateMode.SEPARATED.value
    )

```

评论区精华

review 中的核心讨论集中在三处：其一，dllehr-amd 提议将环境变量改名为 VLLM_ROCM_USE_AITER_MOE_SITUV2，理由是避免在用户不知情的情况下把 a8w4 自动转成 a4w4（原文："This way we aren't exposed to having something auto convert a8w4 into a4w4 without a user knowing the datatype changed"），最终采纳新名并保留 deprecated alias；其二，dllehr-amd 质疑测试中手动 setenv AITER_SITUV2_A4W4 是否冗余，ppalanga 澄清 refresh_env_variables() 只在 VLLM 开关开启时才写 AITER env，禁用分支需要显式构造残留状态；其三，hongxiayang 追问 AITER_BF16_FP8_MOE_BOUND 是否还需要，ppalanga 确认该变量只被 AITER 的 interleaved Swiglu 分支读取，因此把 workaround 移出 SiTUV2 路径并同步从 recipes#954 中移除，hongxiayang 提醒该变量当初仅针对 kimi-k3 添加并 cc @tjtanaa 关注 DeepSeekV4。

- 环境变量命名与语义：VLLM_ROCM_USE_AITER_MOE_SITUV2 vs A8W4 (design): 采纳新名，旧名 VLLM_ROCM_USE_AITER_MOE_SITUV2_A8W4 保留为 deprecated alias，语义上仍启用同一 a4w4 路径。
- 测试中手动设置 AITER_SITUV2_A4W4 是否冗余 (testing): 保留显式 setenv，测试逻辑正确。
- AITER_BF16_FP8_MOE_BOUND 作用域收窄 (correctness): workaround 移出 SiTUV2 路径，仅保留在 interleaved a16w4 路径；recipes#954 同步删除。
- env 变更需要配套 recipe 更新 (other): 外部 recipes 仓库同步更新，维护部署一致性。

风险与影响

- 风险：
 1. 旧名新义风险：VLLM_ROCM_USE_AITER_MOE_SITUV2_A8W4 名称含 A8W4 但实际路由 a4w4，外部脚本与 recipe 长期沿用旧名会形成理解偏差；代码注释和测试已明确，但用户侧不易察觉。
 2. AITER 版本下限：AITER < v0.1.20 时虽仍会启动 afp4 内核，但缺少 tuned CSV 会 fallback 到 heuristic FlyDSL 配置，属于静默性能回退而非报错，排查困难。当前 vLLM main 已 pin AITER v0.1.21.post1，但自定义镜像可能使用旧版。
 3. 数据布局契约变更：oracle/mx4.py 权重 shuffle 从 interleaved 改为 separated、rocm_aiter_moe.py 的 gate_mode 固定 separated，若未来 SiTU 分支被其他模型复用或 AITER 侧布局约定变化，会产生 kernel 级错误结果而非崩溃，风险隐蔽。
 4. 端到端收益不确定：A/B 数据显示 conc 1 基本持平（早期 -2.1%、后期 +0.8%），高并发档位（8/32/64）才有 1.7%–3.2% 提升；PR body 提到 agentic workload 在 conc=1 有提升但未贴出具体数据。
 5. 未完成验证：测试计划中 Verify cudagraph correctness under full decode 与 8k/1k serving A/B on AITER with #4463 tuned CSVs 仍未勾选，虽然后续 nightly 数据已部分覆盖后者。- 影响：影响范围集中在 ROCm gfx9 + Kimi-K3 SiTUV2 MXFP4 配方路径：启用 VLLM_ROCM_USE_AITER_MOE_SITUV2（或旧 alias）的用户会自动切换到 a4w4 kernel 且无需改名，但要求 AITER >= v0.1.20 才能拿到 tuned 配置；权重布局从 interleaved 变为 separated，与 ATOM kernel 选择对齐。对团队而言，这是 vLLM 与 AITER 外部 kernel 库的一次跨仓库契约调整，同时维护了向后兼容 alias，但引入了旧名新义的隐晦点。DeepSeekV4 等 interleaved a16w4 路径受

AITER_BF16_FP8_MOE_BOUND 移动影响，应无行为变化但需回归确认。整体影响面有限，属配方级默认行为切换。 - 风险标记：旧环境变量语义变更，依赖 AITER >= v0.1.20 tuned 配置，权重布局数据契约变更，低并发档位收益不确定，cudagraph 验证未完成

关联脉络

- PR #50813 Quark a8w4 opt-in (title not provided in materials): PR body 明确提及：#50813 为 Quark checkpoint 添加显式 opt-in a8w4，与本 PR 的 a4w4 默认互为补充，同属 SiTUv2 路径按量化来源分化。
- PR #56845 [Refactor] Remove dead kernel code: 近期清理了 kimi_k3 与 ROCm AITER 相关死内核代码（vllm/models/kimi_k3/nvidia、vllm/v1/attention/ops 等），与 Kimi-K3 ROCm kernel 演进线相关。
- PR #55966 [ROCm][Spec Decode] Add Aiter MLA decode support non-causal draft block: 同一 ROCm AITER 集成线，且改过 vllm/_aiter_ops.py，与本 PR 的 SiTUv2 a4w4 开关共享 env 同步机制。