

PR #39763 完整报告

sgl-project/sglang

[AMD] Clamp MORI intranode grid GPUs

合并时间: 2026-09-16 17:14

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/39763>

执行摘要

- 一句话: MORI intranode grid 按设备 CU 数收敛, 避免小 CU 设备 barrier 死锁
- 推荐动作: 值得精读, 但只需读一处的 9 行代码 —— 它是一个信息密度很高的小修复: 既展示了 MoE dispatch kernel 中 grid-wide barrier 对 block 驻留数的硬约束, 也展示了“保守上界 vs 实测驻留上限”的典型工程取舍。

精读建议关注三个点: (1) clamp 放在 dtype 分支之后而非 fp4 分支内部, 是否是有意的“全局收敛”; (2) `EpMode.INTRA_NODE` 与 `mori.ops.EpDispatchCombineKernelType.IntraNode` 双枚举判定是否等价; (3) `torch.cuda.current_device()` 在初始化早期是否已被正确设置。

对审阅者的行动建议: 合入方向正确, 但应把两件事补上——一是为 96 CU DPX 场景和非 fp4 dtype 路径补一个可复现的回归测试 (哪怕是在 CU 数受限的模拟配置上断言 `block_num <= cu_count`); 二是补充 SPEED 数据或至少在 PR/ 文档中记录 grid 缩小对 EP4 吞吐的实测影响, 避免后续把性能回退误判为 kernel 问题。同时建议在创建 MORI 配置后打印最终的 `block_num`, 让部署时的自适应行为可观测。

功能与动机

PR body 明确说明目标是 “supports gfx1250 deployment rather than one fixed lab topology”: 同一机架上的 gfx1250 A0 既可能以 192-CU 的 SPX 设备暴露, 也可能被拆成两个 96-CU 的 DPX agent。而 MORI 的 intranode 标准 MoE dispatch kernel “ends with a grid-wide software barrier that requires every block to become resident”, 默认 256 块 grid 在少 CU 设备上 “can deadlock the kernel at the barrier”。作者希望在不改动上层配置的前提下, 让同一条部署路径同时适配 SPX/DPX 两种拓扑, 因此选择 “Clamp the MORI IntraNode block count to the current device's compute-unit count. Configurations already below that limit are unchanged.”

实现拆解

1. 变更入口与上下文: 唯一改动文件是 `python/sglang/srt/layers/moe/token_dispatcher/moriep.py`, 落在 `init_mori_op` 内。该函数负责组装 MORI 的 `EpDispatchCombineConfig`, 其中 `block_num / warp_num_per_block` 决定 dispatch/combine 的 grid 形状; 此前 `block_num` 默认 256, 仅在 `DispatchDtype.fp4` 且 `mode == EpMode.INTRA_NODE` 分支里按 `num_max_dispatch_tokens_per_rank` 覆写为 225 或 256。

2. 核心逻辑 (+9 行守卫) : 新增 `if kernel_type == mori.ops.EpDispatchCombineKernelType.IntraNode`: 判断, 读取 `torch.cuda.get_device_properties(torch.cuda.current_device()).multi_processor_count`, 执行 `block_num = min(block_num, cu_count)`。插入点位于 `dtype` 分支之后、`combine_quant_type` 计算之前, 因此对 bf16/fp8/mxfp8/fp4 全部 dispatch dtype 生效。这样做的原因很直接: 任何 dtype 走 intra-node 且 `block_num` 超过驻留能力都会卡在 barrier 上; 但这也意味着在 CU 数小于 256 的设备上, 全部 dtype 的 grid 都会被缩小, 行为面比标题暗示的 fp4 场景更宽。
3. 设计取舍: 代码注释明确写出“clamping to the CU count is a conservative bound, not the measured residency limit”。即作者放弃运行期 occupancy 探测, 改用 CU 数这个静态、零成本的上界。收益是初始化阶段不需要额外 kernel/ 查询即可消除死锁; 代价是并行度损失: 192-CU 设备从 256 降到 192 (-25%), DPX 96-CU 分区降到 96 (-62.5%), 而 PR 未提供任何 speed 数据来量化这部分影响。
4. 对下游逻辑的影响: `block_num` 会被写入传给 `mori.ops` 的 config, 派生出实际启动的 block 数; 由于只做 `min`, 对 CU 数 ≥ 256 的设备 (如 MI355X) 数值与改前完全一致, 属严格向后兼容的收紧。`warp_num_per_block` 未同步调整, clamp 后 fp4 路径仍保留 5/16 的 warp 配置, 是否与缩小后的 grid 仍匹配没有在 PR 中说明。
5. 测试、配置与部署配套: 本次没有任何测试文件、配置面或 Docker/CI 改动。作者在“Accuracy Tests”中只声明“The patched SGLang image builds successfully and the 192-CU clamp was verified on MI455 A0 in SPX mode”, 并把“Full DeepSeek-V4-Flash EP4 serving validation is in progress”留作未闭环项; checklist 中 format、unit test、doc、benchmark 四项均未勾选。PR body 的 CI slot 显示 AMD ROCm 10 任务为失败状态。

关键文件:

- `python/sglang/srt/layers/moe/token_dispatcher/moriep.py` (模块 专家分发; 类别 source; 类型 core-logic; 符号 `init_mori_op`, `block_num`, `warp_num_per_block`) : 唯一改动文件。在 `init_mori_op` 中新增 `IntraNode` 守卫, 把 MORI 的 `block_num` 收敛到设备 CU 数, 直接消除 192/96 CU 设备上 grid-wide barrier 死锁; 同时把 clamp 放在 dtype 分支之后, 使 bf16/fp8/mxfp8/fp4 全部分发路径共享该上界, 是本 PR 全部行为变化的来源。

关键符号: `init_mori_op`

关键源码片段

`python/sglang/srt/layers/moe/token_dispatcher/moriep.py`

唯一改动文件。在 `init_mori_op` 中新增 `IntraNode` 守卫, 把 MORI 的 `block_num` 收敛到设备 CU 数, 直接消除 192/96 CU 设备上 grid-wide barrier 死锁; 同时把 clamp 放在 dtype 分支之后, 使 bf16/fp8/mxfp8/fp4 全部分发路径共享该上界, 是本 PR 全部行为变化的来源。

```
if dispatch_dtype == DispatchDtype.fp4:
    # FP4 kernel 仍然拿原始 hidden size 并在内部做量化,
    # 因此这里不能换成 packed 后的 hidden size.
    hidden_dim = hidden_size
    scale_dim = hidden_size // MXFP4_BLOCK_SIZE
```

```

data_type = torch.float4_e2m1fn_x2
scale_type_size = torch.float8_e8m0fnu.itemsize

if mode == EpMode.INTRA_NODE:
    # 单机 EP 场景按每条 rank 的最大 dispatch token 数选择默认 grid:
    # 小于 128 用 225 块 / 5 warp, 否则用 256 块 / 16 warp。
    if num_max_dispatch_tokens_per_rank < 128:
        block_num = 225
        warp_num_per_block = 5
    else:
        block_num = 256
        warp_num_per_block = 16

# 新增守卫: 注意它位于 dtype 分支之外, 因此对 bf16 / fp8 / mxfp8 / fp4 全部生效,
# 而不是只修 fp4 那一条路径。
if kernel_type == mori.ops.EpDispatchCombineKernelType.IntraNode:
    # mori 的 intranode combine 内核以 grid-wide software barrier 收尾,
    # 要求 grid 中每个 block 都能同时驻留; 只要有 block 起不来,
    # 先到达 barrier 的 block 就会一直自旋, 最终整个 kernel 死锁。
    # gfx1250 A0 在 SPX 模式下只暴露 192 CU, DPX 模式下每个 partition 只有 96 CU,
    # 都装不下默认的 256 块 grid。
    # 这里用设备 CU 数作为保守上界 (并非实测驻留上限):
    # 高于上界的配置被收敛, 低于上界的配置数值保持不变。
    cu_count = torch.cuda.get_device_properties(
        torch.cuda.current_device()
    ).multi_processor_count
    block_num = min(block_num, cu_count)

# 该 block_num 随后会随其它参数写入 EpDispatchCombineConfig,
# 决定实际启动的 grid 大小; warp_num_per_block 未做同步调整。
combine_quant_type = "none"
if combine_dtype == CombineDtype.fp8:
    combine_quant_type = "fp8_blockwise"
elif combine_dtype == CombineDtype.fp8_direct_cast:
    combine_quant_type = "fp8_direct_cast"
elif combine_dtype == CombineDtype.fp4:
    combine_quant_type = "fp4_blockwise"

```

评论区精华

本 PR 的 review 记录极简: 仅 HaiShaw 给出一次 APPROVED, body 为空, 没有任何 inline review comment, 也没有 issue 讨论。因此全部设计说明都沉淀在代码注释与 PR body 里, 最有价值的两条自述是:

“clamping to the CU count is a conservative bound, not the measured residency limit.” —— 作者主动承认上界是保守估计而非实测值, 隐含了性能可能被过度限制的风险。

“Full DeepSeek-V4-Flash EP4 serving validation is in progress.” —— 验证尚未完成即合入, 说明这是一次以“先止血”为优先的 hotfix。

值得注意的是，review 过程没有对两点提出质疑：一是 `torch.cuda.current_device()` 取 CU 数依赖隐式设备上下文；二是新增守卫用 mori 的 `EpDispatchCombineKernelType.IntraNode` 判定，而上方的 fp4 分支用 SGLang 侧的 `EpMode.INTRA_NODE` 判定，两套枚举并存是否为恒等映射未被确认。

- IntraNode grid 用 CU 数作为保守上界的设计取舍 (design): reviewer HaiShaw 直接 APPROVED 且无评论，保守上界方案被接受为快速止血；实测驻留上限、occupancy 探测与性能代价均未在 PR 内闭环。
- 缺少自动化测试与 DPX 场景验证 (testing): 未解决：96 CU 分区、非 fp4 dtype 路径以及 CU 数介于 96 与 256 之间的设备都没有可复现的验证基线。
- mode 与 kernel_type 两套枚举判定的等价性 (correctness): PR 内未讨论，属于需要在后续跟进确认的开放问题。

风险与影响

- 风险：

1. 保守上界的双向风险 (`moriep.py / init_mori_op`) : CU 数并非可靠的驻留上限。若某 dtype 下每个 SM 只能驻留 1 个 block，clamp 到 CU 数恰好够用；但如果 block 因 shared memory / register 压力导致单 SM 驻留数低于 1（即部分 SM 根本无法容纳一个 block），`min(block_num, cu_count)` 仍可能大于真实可驻留 block 数，barrier 死锁依旧存在，只是触发概率降低。注释中“conservative bound”的表述在这种情况下并不成立。
1. 设备上下文依赖：`torch.cuda.current_device()` 读取的是当前线程的隐式 CU 设备。若在多卡 / 多进程初始化早期 current device 尚未切到本地 rank，会读到别的设备（甚至 CPU 侧默认 0 号卡）的 CU 数，可能偏大（不做收敛）或偏小（无谓降并行）。同仓库近期 PR#38774 正是同类问题的修复先例。
2. 判定枚举不一致：守卫条件使用 `kernel_type == mori.ops.EpDispatchCombineKernelType.IntraNode`，而决定 `block_num` 数值的 fp4 分支使用 `mode == EpMode.INTRA_NODE`。如果这两个变量在某些配置下不等价，最坏情形是 fp4 + intra-node 但 `kernel_type` 非 `IntraNode`，导致 256 块 grid 不被收敛——恰恰是 PR 想修的场景。
3. 缺少测试覆盖：无单测、无集成测试、无 benchmark。96 CU 的 DPX 分区、非 fp4 dtype 路径、以及 CU 数介于 96 与 256 之间的设备均无验证基线；PR 描述也确认完整 EP4 服务验证仍在进行。
4. 性能回归未量化：clamp 会直接削减 dispatch/combine 的并行度（192 CU 为 -25%，96 CU 为 -62.5%），而 `warp_num_per_block` 未做配套调整，dispatch 吞吐与 EP 通信重叠可能退化。对延迟敏感的 EP4/EP8 服务属于直接暴露面。
5. CI 状态：PR body 中“Latest PR Test (Extra)”与“Latest PR Test (AMD ROCm 10)”两个 slot 均显示失败，需确认是否为与本改动无关的 flake，否则不能视为绿灯合入。
 - 影响：用户侧：影响面仅限 AMD ROCm 上使用 MORI 后端做专家并行（EP）的部署。对 gfx1250 SPX（192 CU）与 DPX（96 CU 分区）用户，本改动把“必现死锁”变为“可运行”，是功能可用性的硬修复；对 MI3xx 等 CU 数 ≥ 256 的设备，`min` 不改数值，行为零变化。但对同在 < 256 CU 设备上运行 bf16/fp8/mxfp8 intra-node dispatch 的用户，grid 被一并缩小，可能出现未被文档化的吞吐变化。

系统侧：MOE token dispatcher 是 EP 推理的关键路径，任何 grid 形状变化都会影响 dispatch/combine 时延，但由于该逻辑只在 `init_mori_op` 初始化阶段执行一次，不存在每 token 的额外开销（无 device sync、无 per-step 查询）。风险集中在初始化正确性与稳态吞吐，而非运行时抖动。

团队侧：这是一次典型的硬件拓扑适配型修复，说明 SGLang 的 MI455/gfx1250 支持正在从“固定实验室拓扑”走向“同一镜像适配 SPX/DPX 多种暴露形态”。代价是配置默认值开始依赖运行设备属性，后续若继续引入类似“按设备能力自适应”的逻辑，需要统一的设备属性探测与缓存策略，否则会散落在各处形成隐性行为差异。

- 风险标记：核心路径变更，缺少测试覆盖，性能回归未验证，依赖隐式设备上下文，枚举判定可能不等价

关联脉络

- PR #39547 [AMD][bugfix] Fix DSV4 MTP crash: 同属 AMD ROCm 上 DeepSeek 系 EP/MoE 路径的紧急 hotfix，改动同样高度集中（单文件、用门控条件规避特定硬件 / 配置下的崩溃），与本 PR 在修复模式与影响人群上高度重合。
- PR #38774 Fix device context during NIXL backend initialization: 同样针对初始化阶段隐式依赖 CUDA/HIP 设备上下文的问题。本 PR 通过 `torch.cuda.current_device()` 读取 CU 数，若设备上下文未正确设置会读到错误的 `multi_processor_count`，属于同一类风险的延续。
- PR #39613 Accept MXFP8 dispatch in FlashInfer A2A TRT-LLM MoE: 同为 MoE dispatch 入口的守卫 / 参数放宽类改动，关注 dispatch 配置与底层通信内核之间的兼容边界，可与本 PR 的“按设备能力收敛配置”思路对照阅读。