

PR #36747 完整报告

sgl-project/sglang

Revert "[NPU] [bugfix] Fix import of ggml_moe_a8_vec and Fix NPU MLA HiCache backup accessing missing data_ptrs"

合并时间: 2026-08-28 07:16

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/36747>

执行摘要

- 一句话: 回滚 #36640, NPU MLA 修复引发回归
- 推荐动作: 此 PR 是临时回滚, 建议关注 #36813 的后续修复, 确保其中包含针对 NPU MLA 的修复且不引入回归。值得回顾原 PR 的设计权衡, 特别是如何处理跨硬件差异。

功能与动机

PR #36640 的目标是修复 NPU 上 `import ggml_moe_a8_vec` 失败以及 NPU MLA HiCache 备份访问缺失 `data_ptrs` 的问题。但该修复引入了回归: 在非 NPU 环境中, 无条件导入 `ExpertPackMoEMethod` 导致 `ggml_moe_a8_vec` 导入失败; 同时移除了 NPU 专用的 `kernel_ascend` 处理, 导致 NPU MLA HiCache 备份路径崩溃。因此需要回滚该 PR, 以恢复稳定状态。

实现拆解

本 PR 为纯回滚操作, 包含两个文件的变更:

1. `python/sglang/srt/layers/quantization/mx_fp4_flashinfer_trtllm_moe.py`: 恢复 `is_npu` 导入, 并恢复 `maybe_fuse_routed_scale_and_shared_add` 中原有的条件导入逻辑——仅在非 NPU 时才导入 `ExpertPackMoEMethod`, 避免在 CUDA 环境引发 `ggml_moe_a8_vec` 导入错误。
2. `python/sglang/srt/mem_cache/pool_host/mla.py`: 恢复 `backup_from_device_all_layer` 中对 `io_backend == "kernel_ascend"` 的特殊处理, 确保 NPU 池使用连续多层张量, 不构建 CUDA 风格的 `data_ptrs` 数组。

未增加测试, 回滚后回归风险已消除。

关键文件:

- `python/sglang/srt/layers/quantization/mx_fp4_flashinfer_trtllm_moe.py` (模块 量化层; 类别 source; 类型 dependency-wiring; 符号 `maybe_fuse_routed_scale_and_shared_add`): 该文件是本次回滚的主要部分, 恢复了 NPU 条件导入逻辑, 避免非 NPU 环境出现 `ggml_moe_a8_vec` 导入错误。
- `python/sglang/srt/mem_cache/pool_host/mla.py` (模块 缓存池; 类别 source; 类型 core-logic; 符号 `backup_from_device_all_layer`): 恢复 NPU 专用的 `kernel_ascend` 处理, 确保 NPU MLA HiCache 备份路径正确。

关键符号: maybe_fuse_routed_scale_and_shared_add, backup_from_device_all_layer

评论区精华

原作者在 Issue 评论中承认自己的修改并未完美解决问题，但认为对 MLA 的修复不应被回滚，并承诺通过新 PR #36813 单独修复 MLA 问题。

- 回滚决定 (question): 决定回滚 #36640，并由 #36813 单独修复 MLA 问题。

风险与影响

- 风险：回滚后，NPU 上的两个原始问题（ggml_moe_a8_vec 导入失败和 MLA HiCache 备份缺失 data_ptrs）仍然存在，在 NPU 上运行相关功能时可能崩溃。另外，回滚未包含测试，无法验证回滚后在其他平台上的稳定性。
- 影响：影响范围为所有使用 NPU 进行推理的用户，特别是依赖 MLA HiCache 备份的场景。回滚后，NPU 相关功能可能回到未修复状态，而其他平台（CUDA）则恢复稳定。
- 风险标记：缺少测试覆盖，NPU 回归未修复

关联脉络

- PR #36640 [NPU] [bugfix] Fix import of ggml_moe_a8_vec and Fix NPU MLA HiCache backup accessing missing data_ptrs: 被本 PR 回滚，是回归的源头。