PR #21657 完整报告

sgl-project/sglang

[AMD] Use tgemm.mm for MoEGate router gemm in deepseek_v2.py

合并时间：2026-03-31 15:55

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/21657

执行摘要

本 PR 通过将 MoEGate 路由器 GEMM 内核切换为使用 tgemm.mm 自动分发器，显著优化了 GLM-5-fp8 和 deepseek 模型的推理性能，吞吐量提升最高达 143%，同时简化了代码逻辑，移除手动形状约束。

功能与动机

MoEGate router GEMM 在 GLM-5-fp8 模型中占用约 30% 的单层时间，成为主要性能瓶颈。为解决此问题，PR 遵循 ATOM 方法，将内核选择自动化，以提高推理效率。PR body 中明确提到："When running the GLM-5-fp8 model, the MoEGate router GEMM is a major bottleneck, taking about ~30%of a single layer's time."

实现拆解

实现主要包括两个文件的修改：

- python/sglang/srt/layers/rocm_linear_utils.py：重构 aiter_dsv3_router_gemm 函数，从原本的复杂逻辑（如使用 gemm_a16w16 和 gemm_a16w16_atomic，并依赖形状检查和零分配器）简化为直接调用 tgemm.mm。关键代码如下：
- python/sglang/srt/models/deepseek_v2.py：更新条件分支，从依赖 _use_aiter_gfx95 和形状约束（如 hidden_states.shape[0] <= 256）改为仅使用 _use_aiter 标志，确保 GLM 和 DeepSeek 模型共享相同路径。

评论区精华

review 讨论简单，无争议点：

- gemini-code-assist[bot]评论："This pull request refactors the aiter_dsv3_router_gemm function to utilize the aiter tuned GEMM dispatcher (tgemm.mm), simplifying the implementation by removing manual kernel selection and shape-based constraints."
- HaiShaw直接批准，无额外评论。 讨论迅速收敛，结论是变更合理且无风险。

风险与影响

- 风险：

    - tgemm.mm 分发器的稳定性未充分验证，可能在不同硬件或输入形状下表现不一致。
    - 移除形状约束可能引入边缘情况性能下降或错误，例如原逻辑针对小矩阵（M <= 256）有特殊处理。
    - 修改核心推理路径，需确保准确性，尽管测试显示 GSM8k 准确率保持不变（约 0.949）。
- 影响：
    - 用户：推理速度大幅提升，benchmark 显示 tok/s 在并发数 4 时提升 143.68%，改善用户体验。
    - 系统：代码简化减少维护成本，但增加对 aiter.tuned_gemm 的依赖，可能影响其他模型组件。
    - 团队：提供性能优化案例，促进自动化内核选择在项目中的应用。

关联脉络

与近期 PR 关联显示性能优化是仓库的持续方向：

- PR #21458：同样针对 AMD 平台优化，涉及内核融合，与本 PR 在优化策略上呼应。
- PR #21314：涉及 GEMM 内核改进，技术领域相似，可作为参考以了解更广泛的性能优化趋势。 这些 PR 共同表明仓库正积极通过内核级优化提升推理效率，尤其是在 AMD 硬件上。