PR #34953 完整报告

sgl-project/sglang

[Perf] Restore the 16-token router GEMM threshold on SM10X

合并时间：2026-08-19 10:53

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/34953

执行摘要

- 一句话：恢复 SM10X 路由 GEMM 阈值至 16，EAGLE 重回 JIT 快路径
- 推荐动作：值得快速阅读 PR body 的 benchmark 方法论：它演示了“阈值调优必须与被测量 fallback 绑定”的教训，以及 speculative decoding 下 M 的放大效应。改动本身只是字面回退，建议合入后关注 Extra CI 失败是否残留。

功能与动机

PR body 明确指出：#29470 dropped max_router_gemm_tokens from 16 to 4 on SM10X. That doesn't hold anymore: I re-ran its sweep and the JIT dsv3_router_gemm wins every M up to 12, ties 13-16. 同时强调 speculative decoding 与普通 decode 的差异：router 看到的是 M = batch_size * num_draft_tokens，EAGLE 5/1/6 在并发 1 时 M 已是 6，plain decode 则要到并发 4 才触及阈值，因此 EAGLE 永远拿不到快路径。

实现拆解

1. 定位回归源头：在 python/sglang/srt/models/deepseek_v2.py 的 DeepseekV2MoE.forward 中，路由 GEMM 的 JIT 触发条件被 #29470 改为 max_router_gemm_tokens = 4 if _device_sm in (100, 103) else 16，使 SM10X 上 M > 4 的路由调用全部落到 cuBLAS fallback。
2. 重新测量：作者在 4x GB300 + GLM-5.2-NVFP4 TP4 上做了 16 个 M 点完整 sweep，以及 7 次服务启动共 19 次 E2E 对比（EAGLE 5/1/6、并发 1），证明新基线 linear_bf16_fp32 下 JIT 在 M=1-12 全胜、13-16 打平，E2E 前后区间不重叠。
3. 实施变更：删除阈值变量与注释，条件简化为 hidden_states.shape[0] <= 16，与 linear_with_fused_a_gemm 的编译范围一致，恢复对所有 CUDA SM >= 90 架构的统一快路径行为。
4. 测试与 CI 配套：未新增自动化测试；PR 带 run-ci 标签，PR Test 通过、Extra 阶段失败（PR body 未说明原因），验证主要依赖手工 benchmark。

关键文件：

- python/sglang/srt/models/deepseek_v2.py（模块 路由选择；类别 source；类型 core-logic；符号 DeepseekV2MoE.forward）：DeepseekV2MoE.forward 内路由 GEMM 的 JIT 触发阈值由 SM10X 专属 4 恢复为统一的 16，是本次性能恢复的唯一落点，直接决定 EAGLE 等场景是否走 dsv3_router_gemm 快路径。

关键符号：DeepseekV2MoE.forward

关键源码片段

python/sglang/srt/models/deepseek_v2.py

DeepseekV2MoE.forward 内路由 GEMM 的 JIT 触发阈值由 SM10X 专属 4 恢复为统一的 16，是本次性能恢复的唯一落点，直接决定 EAGLE 等场景是否走 dsv3_router_gemm 快路径。

```
# DeepseekV2MoE.forward 中路由 GEMM 的选路分支（省略 AMX / 确定性推理 / prefill CP
等前置短路）
else:
    # 恢复统一的 16-token 阈值：PR #29470 曾在 SM10X 上降到 4，但在 fallback 换为
    # linear_bf16_fp32（PR #29783）后重新 sweep 显示，JIT 在 M <= 12 全胜、13-16 打平；
    # 且 EAGLE 等 speculative decoding 的 M = batch_size * num_draft_tokens，
    # 旧阈值 4 会把这类请求完全挡在 JIT 快路径之外。
    if (
        _is_cuda
        and hidden_states.shape[0] <= 16 # 上限与 dsv3_router_gemm 编译的 1..16 范围一致
        and hidden_states.shape[1] % 1024 == 0
        and (self.weight.shape[0] == 256 or self.weight.shape[0] == 384)
        and _device_sm >= 90
    ):
        # JIT 编译的 bf16 router GEMM，直接输出 fp32 logits
        logits = dsv3_router_gemm(hidden_states, self.weight, out_dtype=torch.float32)

    elif _use_aiter:
        logits = aiter_dsv3_router_gemm(hidden_states, self.weight)
    elif not _is_cuda:
        logits = F.linear(hidden_states, self.weight, None)
    else:
        # cuBLAS bf16 x bf16 -> fp32 GEMM（torch.mm 的 out_dtype 参数仅 CUDA 可用）
        from sglang.kernels.ops.attention.dsv4 import linear_bf16_fp32

        logits = linear_bf16_fp32(hidden_states, self.weight)

return logits
```

评论区精华

该 PR 没有产生 review 评论。两位 reviewer 均直接批准：b8zhong（#29470 中原阈值的注释署名者）批复 “Amaze”，Fridge003 批准。没有未解决的疑虑。

- 暂无高价值评论线程

风险与影响

- 风险：
  - SM10X 上 M=13-16 时 JIT 与 cuBLAS 打平，恢复 16 不会引入明显劣化；非 SM10X 架构阈值本就是 16，行为不变。

- 风险集中在“阈值正确性依赖 benchmark 基线”：如果未来 fallback 再从 linear_bf16_fp32 变化，胜负可能再次反转；且本轮只测了 GLM-5.2-NVFP4 一种模型 / 量化组合。
- 缺少针对该阈值的自动化测试，后续回归只能靠人工跑 M sweep 或 E2E 复现。
- CI Extra 阶段失败，正文未说明原因，需确认与本次改动无关。
- 影响：影响面集中在 DeepSeek 系（DeepSeek-V3 系列与 GLM 等复用该 MoE 实现）在 SM10X GPU 上的路由 GEMM 路径。受益最大的是 speculative decoding 场景——EAGLE 的 router M 等于 batch_size * num_draft_tokens，4-token 旧阈值会完全屏蔽 JIT；恢复后该场景实测吞吐 +2.16%、TPOT -2.38%。普通 decode 的并发 1-4 也会重新吃到 JIT 快路径。无 API、配置或数据契约变化，对用户不可见，仅内部性能表现改变。
- 风险标记：阈值依赖手工 benchmark 验证，无自动化测试覆盖，CI Extra 阶段失败待确认

关联脉络

- PR #29470 PR body 提及的 SM10X 阈值下调（原文未给标题）：本 PR 是对该改动的字面回退（literal revert），重新在 16 点 M sweep 上验证原结论失效。
- PR #29783 PR body 推测的 fallback 切换（F.linear 到 linear_bf16_fp32）：作者推测正是这次 fallback 替换让 #29470 的测量基线失效，是胜负反转的疑似根因，但未经验证。