PR #30247 完整报告

sgl-project/sglang

Optimize LongCat-Flash router GEMM with the HPC-Ops bf16xfp32 kernel

合并时间：2026-07-21 20:05

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/30247

执行摘要

- 一句话：用 HPC-Ops bf16xfp32 内核加速 LongCat-Flash 路由器 GEMM
- 推荐动作：值得精读，尤其是权重拆分缓存的设计、通过 min-M 阈值在性能与兼容性之间平衡的策略，以及外部核集成模式。同时关注后续 #31943 对缓存一致性的修复，以填补本 PR 遗留的安全缺口。

功能与动机

LongCat-Flash 模型的路由器 GEMM（bf16 激活 × fp32 权重）是 moe 路由部分的计算热点。当前实现使用 fp32 矩阵乘法（将激活转换为 float），不能充分利用 bf16 tensor core。HPC-Ops 库的 gemm_bf16xfp32 内核通过分解权重并融合两个 bf16 GEMM，在保持 fp32 权重精度的同时获得大幅加速。PR body 中展示了详细性能数据，对关键形状加速比达 2-4 倍。

实现拆解

1. 在 python/sglang/jit_kernel/dsv4/gemm.py 中新增 HPC-Ops 分支：添加 _hpc_gemm_bf16xfp32_available() 检查 HPC-Ops 是否安装且 GPU 为 Hopper (SM90)；_can_use_hpc_gemm_bf16xfp32() 进行形状、连续性等检查；_get_bf16xfp32_weight_split() 将 fp32 权重拆分为两个 bf16 部分并缓存到权重张量上（使用 _sglang_bf16xfp32_weight_cache 属性）；_linear_bf16_fp32_hpc() 调用 HPC-Ops 内核，若检查不通过返回 None。修改 linear_bf16_fp32() 入口，新增 hpc_kernel_min_m 参数以支持按形状调度。
2. 在 python/sglang/srt/models/longcat_flash.py 中定义每个路由器形状的最小 M 阈值字典 _LONGCAT_FLASH_ROUTER_HPC_GEMM_MIN_M，在 LongcatFlashRouter.forward() 中对匹配形状且无 router_bias、权重 dtype 为 fp32 的情况直接调用 linear_bf16_fp32 并传入阈值，否则回退至原始 classifier 前向路径。
3. 添加两个测试文件：test/registered/gemm/test_linear_bf16_fp32_hpc.py 验证 HPC 路径与 fp32 参考的数值一致性、min_M 下界行为以及权重拆分缓存的复用；test/registered/unit/models/test_longcat_flash_router_hpc_gemm.py 通过 mock 验证 LongCat 路由器向 HPC 内核的正确调度以及未受支持形状的回退。
4. 配置开关：通过环境变量 SGLANG_OPT_BF16_FP32_GEMM_ALGO=hpc 可全局启用，但 LongCat 模型默认使用内部阈值路径；无 HPC-Ops 时所有路径自动回退至 cublas，对其他用户无影响。

关键文件：

- python/sglang/jit_kernel/dsv4/gemm.py（模块 JIT 内核；类别 source；类型 core-logic；符号 _hpc_gemm_bf16xfp32_available, _can_use_hpc_gemm_bf16xfp32, _get_bf16xfp32_weight_split, linear_bf16_fp32）：核心 GEMM 调度实现，新增 HPC-Ops 分支、可用性检查、形状约束、权重拆分缓存。这是性能加速的关键模块。
- python/sglang/srt/models/longcat_flash.py（模块 模型定义；类别 source；类型 data-contract；符号 _LONGCAT_FLASH_ROUTER_HPC_GEMM_MIN_M, LongcatFlashRouter.forward）：路由器的 forward 方法增加了对 HPC 内核的调度逻辑，并定义了每个形状的最小 M 阈值。
- test/registered/gemm/test_linear_bf16_fp32_hpc.py（模块 单元测试；类别 test；类型 test-coverage；符号 TestLinearBf16Fp32Hpc, test_matches_fp32_reference, test_min_m_dispatch, test_weight_split_cache_reused）：数值测试，验证 HPC 路径与 fp32 参考的一致性、min_M 调度及缓存复用。
- test/registered/unit/models/test_longcat_flash_router_hpc_gemm.py（模块 单元测试；类别 test；类型 test-coverage；符号 TestLongcatFlashRouterHpcGemm, test_chat_shape_dispatches_with_benchmark_guard, test_lite_shape_dispatches_with_benchmark_guard, test_unbenchmarked_shape_uses_classifier）：调度逻辑测试，验证在不同形状、配置下正确启用 HPC 或回退到 classifier。

关键符号：linear_bf16_fp32, _linear_bf16_fp32_hpc, _get_bf16xfp32_weight_split, _can_use_hpc_gemm_bf16xfp32, _hpc_gemm_bf16xfp32_available, _linear_bf16_fp32_cublas, LongcatFlashRouter.forward

评论区精华

Review 中主要讨论了三个核心问题：

- 权重缓存一致性问题：VAthree 指出在 param.data.copy_() 进行在线权重更新时，缓存键中的 _version 不会递增，导致缓存未失效而返回旧权重。BBuf 确认并在后续 PR #31943 中修复，改为在权重更新 API 中主动刷新缓存或直接报错。
- 统一 GEMM 接口建议：DarkSharpness 建议将多种 GEMM 统一到一个类似 torch.mm 的接口，BBuf 同意并跟踪到 issue #29630。
- 性能优化建议：gemini-code-assist[bot] 建议缓存 get_device_capability 调用；但代码中已通过 functools.cache 缓存了整个可用性检查，覆盖了此建议。
  - 权重缓存在线更新时可能变 stale (correctness): 已解决：在权重更新 API 中主动刷新权重拆分缓存，或在无法刷新时快速失败。
  - 统一 GEMM 接口以改善代码复用 (design): 创建 issue #29630 跟踪统一 GEMM 接口重构。
  - 缓存 get_device_capability 以减少 Python 开销 (performance): 代码已覆盖，无需额外修改。

风险与影响

- 风险：
  - 权重缓存 stale 风险：在线权重更新场景下，缓存不能正确失效，可能导致静默错误。已在 #31943 中通过强制刷新或失败快速修复，但本 PR 合并时尚未包含该修复。
  - HPC-Ops 依赖与平台限制：内核仅运行于 Hopper (SM90) GPU，且需要安装 HPC-Ops 库。无此环境时自动回退，但回退路径必须经过严格测试。
  - min-M 阈值泛化性：阈值基于 H200 基准测试，在 H100 等其他 Hopper 设备上可能不是最优，但仅影响加速比例，不影响正确性。
  - CUDA graph 兼容性：权重拆分缓存的固定地址可能被 CUDA graph 捕获，若图重播时权重发生变化可能导致错误。已在 #31943 中通过持久化缓存和失败快速处理。
- 影响：
  - 用户：LongCat-Flash 模型用户无需修改配置即可获得 prefill 吞吐提升（2-5%），decode 阶段因 token 数不足 min_M 阈值而保持原性能。其他模型用户不受影响。
  - 系统：需要 Hopper GPU 和 HPC-Ops 库（已由 #31390 集成到镜像）。
  - 团队：维护负担低，内核由 HPC-Ops 上游维护；后续内核调优只需更新 pins 版本。
  - 风险标记：权重缓存 stale 风险，HPC-Ops 外部依赖，仅 Hopper GPU 加速

关联脉络

- 暂无明显关联 PR