PR #32045 完整报告

sgl-project/sglang

[Kernel] Phase 4 batch-3: migrate tangled JIT subsystems + new groups into kernels.ops (RFC #29630)

合并时间：2026-07-22 21:15

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/32045

执行摘要

本 PR 是 RFC #29630 Phase 4 batch-3，将 JIT kernel 子系统从 sglang.jit_kernel 迁移到 sglang.kernels.ops，涉及 389 个文件、8187 行新增和 8159 行删除。迁移后算子按 attention、moe、diffusion、model 等类别模块化组织，旧路径暂时由后续 PR 清理的 shim 保持兼容。

功能与动机

遵循 RFC #29630 统一内核命名空间，清理旧 jit_kernel 包内的杂乱结构。将原本分散在 dsa、dsv4、trtllm_lora_temp、diffusion、minimax_m3、triton 等子目录的算子按功能领域归并到 kernels.ops 的子模块（attention、moe、diffusion、layernorm、model.inkling 等），提升代码可维护性与可扩展性。

实现拆解

1. Stage 1 — 干净移动：将 19 个注意力算子（flash_attention_v3/v4、concat_mla、cutedsl_gdn/kda/paged_mqa、rope、fused_qknorm_rope、hadamard 等）、8 个 MoE 算子、timestep_embedding 直接复制到 kernels.ops/attention 和 kernels.ops/moe，删除旧文件。
2. Stage 2 — 纠缠子系统迁移：处理无法直接复制的复杂目录：
   - dsa/ 和 dsv4/ → 合并到 kernels.ops.attention 的现有 Phase-2.5 目录，__init__ 重导出保持兼容。
   - trtllm_lora_temp/（含自包含的 data/csrc 和 data/include）→ 整体搬到 kernels.ops.moe。
   - diffusion/（含嵌套的 cutedsl/flydsl/triton）→ 搬到 kernels.ops.diffusion。
   - minimax_m3 拆分：rmsnorm → kernels.ops.layernorm，qk_norm_rope → attention，swiglu → moe。
   - triton/ 拆分：gdn_fused_proj → attention，hash_topk/sigmoid_gate_mul → moe。
3. 新组构建：新增 lplb/、kv_canary/、model/inkling 三个算子组，在 _GROUPS 中注册。
4. 调用点更新：遍历 268 个文件（涉及 srt、multimodal_gen、jit_kernel 自身），将所有 sglang.jit_kernel.* 引用改写为新路径。
5. 测试验证：68 个 CPU 内核测试全部通过，lint 检查通过。

python/sglang/jit_kernel/trtllm_lora_temp/data/include/flashinfer/trtllm/fused_moe/runner.h

trtllm_lora_temp 的 MoE 核心头文件旧位置，被完整迁移到 kernels.ops.moe 下，代表最具挑战性的 C++ 代码搬迁 (586 行删除)。

```
/*
 * 旧位置：sglang/jit_kernel/trtllm_lora_temp/data/include/flashinfer/trtllm/fused_moe/runner.h
 * 该文件在 Phase 4 batch-3 中被删除，完整迁至 kernels/ops/moe 下对应路径。
 * 下为原始内容节选，展示了 MoE 路由枚举和辅助函数。
 */

#pragma once

#include "DevKernel.h"
#include "flashinfer/trtllm/fused_moe/RoutingKernel.h"
#include <string>
#include "flashinfer/trtllm/batched_gemm/KernelRunner.h"
#include "flashinfer/trtllm/batched_gemm/trtllmGen_bmm_export/trtllm/gen/DtypeDecl.h"
#include "flashinfer/trtllm/common/cudaUtils.h"

namespace tensorrt_llm {
namespace kernels {
namespace trtllmgen_moe {

namespace MoE {
class Runner;
} // namespace MoE

namespace Routing {

enum class RoutingMethodType : int64_t {
  Default = 0, // Softmax -> TopK
  Renormalize = 1, // TopK -> Softmax
  DeepSeekV3 = 2, // Sigmoid -> Group -> Expert
  Llama4 = 3, // Top1 -> Sigmoid
  RenormalizeNaive = 4, // Softmax -> TopK -> Renormalize
  TopK = 5, // TopK only
  SigmoidRenorm = 6, // Sigmoid -> TopK -> Renormalize
  MiniMax2 = 7, // Sigmoid + Bias -> TopK -> ScaledSumNormalize
  Sigmoid = 8, // Sigmoid -> TopK
  Unspecified = 9,
};

inline int32_t maybeGetMinTokenCount(int32_t numPaddedTokens, int32_t hiddenSize, int32_t
dtypeSizeBits) {
  int32_t minNumTokensRequired = common::divUp(128 * 1024 * 8, hiddenSize * dtypeSizeBits);
  return std::max(numPaddedTokens, minNumTokensRequired);
}
```

```
} // namespace Routing
} // namespace trtllmgen_moe
} // namespace kernels
} // namespace tensorrt_llm
```

python/sglang/kernels/ops/moe/trtllm_lora_temp/data/include/flashinfer/trtllm/fused_moe/runner.h

trtllm_lora_temp 的 MoE 核心头文件新位置，是 Stage 2 中最大搬迁单元，内容与原位置一致但路径更新。

```
/*
 * 新位置：sglang/kernels/ops/moe/trtllm_lora_temp/data/include/flashinfer/trtllm/fused_moe/
runner.h
 * 该文件由旧位置整体搬迁而来，内容几乎不变，仅 include 路径因相对位置变化可能调整。
 * 展示了 MoE 路由枚举在新结构中的位置。
 */

#pragma once

#include "DevKernel.h"
#include "flashinfer/trtllm/fused_moe/RoutingKernel.h"
#include <string>
#include "flashinfer/trtllm/batched_gemm/KernelRunner.h"
#include "flashinfer/trtllm/batched_gemm/trtllmGen_bmm_export/trtllm/gen/DtypeDecl.h"
#include "flashinfer/trtllm/common/cudaUtils.h"

namespace tensorrt_llm {
namespace kernels {
namespace trtllmgen_moe {

namespace MoE {
class Runner;
} // namespace MoE

namespace Routing {

enum class RoutingMethodType : int64_t {
  Default = 0, // Softmax -> TopK
  Renormalize = 1, // TopK -> Softmax
  DeepSeekV3 = 2, // Sigmoid -> Group -> Expert
  Llama4 = 3, // Top1 -> Sigmoid
  RenormalizeNaive = 4, // Softmax -> TopK -> Renormalize
  TopK = 5, // TopK only
  SigmoidRenorm = 6, // Sigmoid -> TopK -> Renormalize
  MiniMax2 = 7, // Sigmoid + Bias -> TopK -> ScaledSumNormalize
  Sigmoid = 8, // Sigmoid -> TopK
  Unspecified = 9,
};
```

```
inline int32_t maybeGetMinTokenCount(int32_t numPaddedTokens, int32_t hiddenSize, int32_t
dtypeSizeBits) {
  int32_t minNumTokensRequired = common::divUp(128 * 1024 * 8, hiddenSize * dtypeSizeBits);
  return std::max(numPaddedTokens, minNumTokensRequired);
}

} // namespace Routing
} // namespace trtllmgen_moe
} // namespace kernels
} // namespace tensorrt_llm
```

评论区精华

该 PR 未收到人工 review 评论，仅有一条 Gemini Code Assist 自动停止提示和一条来自 BBuf 的 CI 链接。BBuf 提供的 CI 链接（Run #29912754833）用于验证变更，推测为内部验证通过。68 个 CPU 内核测试全部通过，lint 检查通过。

风险与影响

风险：

- 大规模路径重写（389 文件变更）可能导致遗漏导入错误，但测试覆盖基本保障。
- trtllm_lora_temp 的 C++ 代码搬迁可能因 include 路径或符号冲突导致编译问题。
- 本 PR 不包含 shim 的完整清理（由 batch-1 处理），存在新旧路径并存期间的维护成本。
- 缺乏人工 review 覆盖（仅有自动 CI 验证）。

影响：

- 所有直接 import sglang.jit_kernel 的代码需迁移到 sglang.kernels.ops，但短期内 shim 保持兼容。
- 算子组织更清晰，便于后续按领域进行优化。
- RFC #29630 进入收尾阶段，该 PR 完成后 jit_kernel 包仅剩 8 个兼容 shim。

关联脉络

- 本 PR 是 RFC #29630 的 Phase 4 batch-3，上一批 batch-2 已完成主要算子迁移。
- #32072 是本系列 finale PR，将彻底删除剩余 shim 和 jit_kernel 空包，实现命名空间最终统一。
- 近期历史 PR 中的 #32072（删除 jit_kernel 包）和本 PR 直接相关，构成完整的迁移链路。