PR #21531 完整报告

sgl-project/sglang

[JIT Kernel] Migrate dsv3_router_gemm from AOT sgl-kernel to JIT kernel

合并时间：2026-06-27 02:52

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/21531

执行摘要

- 一句话：迁移 dsv3_router_gemm 到 JIT 内核，缩减 wheel 体积
- 推荐动作：值得精读。本 PR 是 AOT 到 JIT 迁移的完整范例，涵盖 CUDA 模板设计、Python 包装器、torch.compile 兼容、性能验证、测试覆盖。特别关注 RouterGemmDispatcher 减少编译模块数的优化，以及如何在不影响性能的前提下移除 AOT 依赖。

功能与动机

本 PR 是 sgl-kernel wheel 瘦身计划（issue #17865）的一部分。dsv3_router_gemm 内核的预编译版本在 H100 上占用约 13.7 MB，迁移到 JIT 后可从 sgl-kernel wheel 中移除，显著减少分发包体积，降低用户下载和安装成本。

实现拆解

1. 新增 JIT 内核源文件：创建 python/sglang/jit_kernel/dsv3_router_gemm.py 作为 Python 入口，通过 *cache_once* 和 load_jit 动态编译 CUDA 内核，并用 @register_custom_op 包装为 custom op 以兼容 torch.compile。对应的 CUDA 模板头文件放在 csrc/gemm/dsv3_router_gemm.cuh。
2. 引入 RouterGemmDispatcher 优化编译：将原本针对每种 token 数单独生成的内核合并为一个模板，利用参数展开和条件编译，将 JIT 编译模块数从 O(64) 降至 O(4)，减少首次编译时间。
3. 移除 AOT 内核：从 sgl-kernel 中彻底删除 dsv3_router_gemm 的 BIND、Python 接口、测试和基准文件，清理 CMakeLists.txt 和头文件中的注册。
4. 更新模型调用点：在 deepseek_v2.py 中，将导入从 sgl_kernel.dsv3_router_gemm 和 flashinfer.gemm 切换为 sglang.jit_kernel.dsv3_router_gemm，放宽 hidden_dim 条件为 1024 的倍数，移除 flashinfer 特殊路径，简化条件判断。
5. 添加 CI 测试：在 test/registered/jit/ 下新建单元测试（参数化 256/384 专家数、多 hidden_dim、1-16 token、两种输出 dtype）和基准测试（对比 JIT/AOT/Torch 三种方案），并通过 register_cuda_ci 注册到 CI 套件。

关键文件：

- python/sglang/jit_kernel/dsv3_router_gemm.py（模块 JIT 内核；类别 source；类型 entrypoint；符号 _jit_dsv3_router_gemm_module, _dsv3_router_gemm_custom_op,

dsv3_router_gemm）：JIT 内核的 Python 入口文件，是迁移的核心，包含模块缓存、custom op 注册和公共 API。

- python/sglang/srt/models/deepseek_v2.py（模块 模型层；类别 source；类型 data-contract；符号 flashinfer_dsv3_router_gemm, _jit_dsv3_router_gemm）：模型前向中路由器 GEMM 的调用点被更新，移除旧的 flashinfer 路径和 AOT 导入，统一使用 JIT 内核，并放宽 hidden_dim 限制。
- python/sglang/jit_kernel/csrc/gemm/dsv3_router_gemm.cuh（模块 CUDA 核心；类别 source；类型 core-logic）：新增 CUDA 模板头文件，实现具体的 RouterGemmDispatcher 内核，包含 LoopUnroller 和多种精度支持，是 JIT 编译的核心。
- test/registered/jit/test_dsv3_router_gemm.py（模块 单元测试；类别 test；类型 test-coverage；符号 _ref, test_dsv3_router_gemm）：新增单元测试，覆盖所有合法参数组合，验证数值正确性。
- test/registered/jit/benchmark/bench_dsv3_router_gemm.py（模块 基准测试；类别 test；类型 test-coverage；符号 _torch, benchmark）：新增基准测试，支持 JIT、AOT（若存在）、torch 三种方案对比，注册到 CI。
- sgl-kernel/tests/test_dsv3_router_gemm.py（模块 预编译内核；类别 test；类型 deletion；符号 test_dsv3_router_gemm）：被移除的旧单元测试，原依赖 sgl_kernel.dsv3_router_gemm。

关键符号：dsv3_router_gemm, _jit_dsv3_router_gemm_module, _dsv3_router_gemm_custom_op, benchmark, test_dsv3_router_gemm, DeepseekV2MLP.forward

关键源码片段

python/sglang/jit_kernel/dsv3_router_gemm.py

JIT 内核的 Python 入口文件，是迁移的核心，包含模块缓存、custom op 注册和公共 API。

```
"""
JIT kernel for DeepSeek V3 router GEMM.
Replaces the AOT sgl_kernel.dsv3_router_gemm for SM90+ (Hopper) GPUs.
"""

from __future__ import annotations
from typing import TYPE_CHECKING, Optional

import torch

from sglang.jit_kernel.utils import (
    cache_once,
    is_arch_support_pdl,
    load_jit,
    make_cpp_args,
)
from sglang.kernel_api_logging import debug_kernel_api
from sglang.srt.utils.custom_op import register_custom_op
```

```
if TYPE_CHECKING:
    from tvm_ffi.module import Module


@cache_once
# 通过 make_cpp_args 将 Python 参数序列化为 C++ 编译标志
# load_jit 会在第一次调用时编译 CUDA 模板，后续缓存 Module 对象
def _jit_dsv3_router_gemm_module(
    num_experts: int,
    hidden_dim: int,
    use_pdl: bool,
    out_float: bool,
) -> Module:
    args = make_cpp_args(num_experts, hidden_dim, use_pdl, out_float)
    return load_jit(
        "dsv3_router_gemm",
        *args,
        cuda_files=["gemm/dsv3_router_gemm.cuh"],
        cuda_wrappers=[
            ("dsv3_router_gemm", f"DSV3RouterGemmKernel<{args}>::run"),
        ],
    )


@register_custom_op(
    op_name="dsv3_router_gemm",
    mutates_args=["output"], # 此 custom op 会修改 output 张量
)
# 将 tvm_ffi 调用包装为 custom op，避免 torch.compile 追踪时报错
def _dsv3_router_gemm_custom_op(
    hidden_states: torch.Tensor,
    router_weights: torch.Tensor,
    output: torch.Tensor,
) -> None:
    num_experts = router_weights.shape[0]
    hidden_dim = hidden_states.shape[1]
    out_float = output.dtype == torch.float32
    module = _jit_dsv3_router_gemm_module(
        num_experts, hidden_dim, is_arch_support_pdl(), out_float
    )
    module.dsv3_router_gemm(hidden_states, router_weights, output)
    return None


@debug_kernel_api
# 公共 API，支持自动分配输出张量
def dsv3_router_gemm(
    hidden_states: torch.Tensor,
    router_weights: torch.Tensor,
```

```
    out_dtype: torch.dtype = torch.bfloat16,
    output: Optional[torch.Tensor] = None,
) -> torch.Tensor:
    if output is None:
        output = torch.empty(
            hidden_states.shape[0],
            router_weights.shape[0],
            device=hidden_states.device,
            dtype=out_dtype,
        )
    _dsv3_router_gemm_custom_op(hidden_states, router_weights, output)
    return output
```

python/sglang/srt/models/deepseek_v2.py

模型前向中路由器 GEMM 的调用点被更新，移除旧的 flashinfer 路径和 AOT 导入，统一使用 JIT 内核，并放宽 hidden_dim 限制。

```
# 导入部分（head 版本）：
if _is_cuda:
    from sgl_kernel import dsv3_fused_a_gemm
    # 从 sgl_kernel.dsv3_router_gemm 切换到 JIT 版本
    from sglang.jit_kernel.dsv3_router_gemm import (
        dsv3_router_gemm as _jit_dsv3_router_gemm,
    )
elif _is_musa:
    from sgl_kernel import dsv3_fused_a_gemm
    # MUSA 平台不再需要 dsv3_router_gemm（已在 CMake 中移除）
else:
    pass

# forward 中的条件分支 （head 版本）：
if (
    _is_cuda
    and hidden_states.shape[0] <= 16
    and hidden_states.shape[1] % 1024 == 0 # 从硬编码 7168 放宽为任意 1024 倍数
    and (self.weight.shape[0] == 256 or self.weight.shape[0] == 384)
    and _device_sm >= 90
):
    # 统一调用 JIT 内核，不再区分 flashinfer 和 sgl_kernel AOT
    logits = _jit_dsv3_router_gemm(
        hidden_states, self.weight, out_dtype=torch.float32
    )
elif _use_aiter:
    logits = aiter_dsv3_router_gemm(hidden_states, self.weight)
elif _is_npu:
    logits = F.linear(hidden_states, self.weight, None)
else:
    # 回退到 F.linear
    ...
```

评论区精华

讨论主要集中在与 #18892 的重合及性能差异（@DarkSharpness 指出可能重复，@meinie0826 解释本 PR 完全移除 AOT 而 #18892 保留 fallback，且内核逻辑一致）。@b8zhong 提供了详细的基准测试结果，证明 JIT 与 AOT 在 SM100 上性能对齐，并新增 hidden_dim=6144 支持。代码质量上，@gemini-code-assist 提出避免 const_cast、使用 std::array 替代 VLA 等建议；@Copilot 建议添加输入验证和 SM90+ 守卫。@b8zhong 多次要求修复 lint 和 CI 问题，最终批准。

- 与 #18892 重复及性能对齐 (correctness): 双方确认内核逻辑一致，且性能 benchmark 对齐。
- JIT 与 AOT 性能对齐 (performance): 性能无回归，可合并。
- 代码质量建议（CUDA） (style): 未在最终版本中完全解决，但审查者接受风险。
- 添加输入验证和 SM90+ 守卫 (correctness): 未采纳，开发者认为由调用者负责。
- CI 失败和合并冲突修复 (other): 修复后通过 CI，获得批准。

风险与影响

- 风险：
  1. 性能回归：基准测试表明性能一致，但未覆盖所有可能配置（如 num_tokens 超出 16），若调用者传入非法参数，可能触发 CUDA panic。
  2. 平台兼容：非 SM90+ GPU（如 SM80）不受影响，但 deepseek_v2.py 中条件未严格限制 _is_cuda+SM90，可能在其他 CUDA 设备上意外调用导致错误。
  3. 依赖移除：sgl-kernel 中删除 dsv3_router_gemm 后，依赖此符号的外部代码将不可用，但 sgl-kernel 发布版本应同步更新。
  4. JIT 首次编译：首次调用时有编译开销，但 cache_once 已缓存。 - 影响：对用户：功能无变化，但 sgl-kernel 包体积减少约 13.7 MB。对系统：减少 sgl-kernel 构建负担，JIT 编译在首次调用后无性能损失。对团队：JIT 内核更易于调试和迭代，CUDA 源码集中管理。 - 风险标记：核心路径变更 (deepseek_v2.py)，JIT 平台限制 (SM90+)，缺少输入验证，AOT 接口移除可能影响外部用户

关联脉络

- PR #18892 [JIT Kernel] Add dsv3_router_gemm JIT kernel: 并行工作，相同功能，本 PR 基于其代码并完全替换 AOT。
- PR #17865 [Feature] sgl-kernel wheel slimming plan tracking: 动机源头，本 PR 是该计划的一部分。