PR #34508 完整报告

sgl-project/sglang

[Diffusion][LTX-2] Allocate AdaLN outputs from one contiguous slab

合并时间：2026-08-12 16:28

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/34508

执行摘要

- 一句话：LTX-2 AdaLN 九输出合并为单 slab 分配，提速 1.63x
- 推荐动作：值得快速阅读。核心价值不在改动规模，而在“多输出 Triton kernel 用单 slab + unbind 视图减少 allocator 往返”这一通用优化模式，以及用 torch.compile(fullgraph=True) 做回归保护的测试思路。生产路径上如存在同类多输出内核，可直接套用该模式。

功能与动机

PR body 明确说明：ltx2_ada_values9 在每次 transformer block 调用时分配 9 个独立的 CUDA tensor，而 Triton kernel 写入的输出形状、dtype 完全相同且互不相交，这些 allocator 往返是不必要的。优化目标是在保持对外九元组 API 不变的前提下，把每次调用的分配次数从 9 次降到 1 次。

实现拆解

1. 定位分配热点：python/sglang/kernels/ops/diffusion/triton/ltx2_ada_values.py 的 ltx2_ada_values9 原实现通过生成器创建 9 个 (batch, seq, hidden) 独立张量，每次 transformer block 调用都产生 9 次 CUDA allocator 往返。
2. 合并为单一 slab：改为分配一个 (9, batch, seq, hidden) 的连续 output_storage，再 unbind(dim=0) 得到 9 个不相交的连续视图；_ltx2_ada_values9_kernel 的调度与写入逻辑不变，返回的九元组 API 保持不变，对上层模型完全透明。
3. 测试契约强化：在 test/registered/kernels/ops/diffusion/test_ltx2_ada_values.py 中给 test_ltx2_ada_values9 增加 is_contiguous() 断言；新增 test_ltx2_ada_values9_torch_compile_fullgraph，用 torch.compile(fullgraph=True) 验证整个调用可被单一计算图捕获、无 graph break。
4. 验证与量化：B300 上以生产形状 B=1, S=1, D=4096、BF16 做 9 轮 × 2000 次调用，wrapper 中位延迟由 24.40us 降至 14.98us（约 1.63x），aten::empty 由 9 次降为 1 次；9 个输出与 eager scale/shift 参考实现对比 atol=0、rtol=0，逐位一致。

关键文件：

- python/sglang/kernels/ops/diffusion/triton/ltx2_ada_values.py（模块 内核算子；类别 source；类型 performance-optimization；符号 ltx2_ada_values9）：核心变更文件：将 ltx2_ada_values9 的 9 次独立 torch.empty 分配合并为单次连续 slab 分配，是本 PR 性

能提升的来源。

- test/registered/kernels/ops/diffusion/test_ltx2_ada_values.py（模块 内核测试；类别 test；类型 test-coverage；符号 test_ltx2_ada_values9, test_ltx2_ada_values9_torch_compile_fullgraph）：配套测试：为原测试补充连续性断言，并新增 torch.compile(fullgraph=True) 回归，确保 slab 视图路径可被完整编译。

关键符号：ltx2_ada_values9, test_ltx2_ada_values9, test_ltx2_ada_values9_torch_compile_fullgraph

关键源码片段

python/sglang/kernels/ops/diffusion/triton/ltx2_ada_values.py

核心变更文件：将 ltx2_ada_values9 的 9 次独立 torch.empty 分配合并为单次连续 slab 分配，是本 PR 性能提升的来源。

```
# python/sglang/kernels/ops/diffusion/triton/ltx2_ada_values.py
def ltx2_ada_values9(scale_shift_table, timestep):
    """为一个 transformer block 计算 9 组 AdaLN scale/shift 输出，返回 9 个连续视图。"""
    batch, seq, _ = timestep.shape
    hidden = scale_shift_table.shape[1]
    rows = int(batch * seq)

    # 旧实现为每个输出单独调用 torch.empty，一次函数调用产生 9 次
    # CUDA allocator 往返；新实现把 9 个输出合并进一个连续 slab，
    # 再用 unbind(0) 切出不相交的视图，分配次数降为 1 次。
    output_storage = torch.empty(
        (9, batch, seq, hidden),
        device=timestep.device,
        dtype=timestep.dtype,
    )
    outs = tuple(output_storage.unbind(dim=0))

    # Triton kernel 仍按行写入这 9 个视图；kernel 参数与调度保持不变，
    # 仅输出存储从 9 个独立张量换成 slab 视图，对上层 API 完全透明。
    _ltx2_ada_values9_kernel[(rows,)](
        timestep,
        scale_shift_table,
        # ... 其余参数与原实现一致
    )
    return outs
```

test/registered/kernels/ops/diffusion/test_ltx2_ada_values.py

配套测试：为原测试补充连续性断言，并新增 torch.compile(fullgraph=True) 回归，确保 slab 视图路径可被完整编译。

```
# test/registered/kernels/ops/diffusion/test_ltx2_ada_values.py
@torch.no_grad()
def test_ltx2_ada_values9_torch_compile_fullgraph() -> None:
```

```
hidden = 4096
scale_shift_table = torch.randn(
    9, hidden, device=DEVICE, dtype=torch.bfloat16
).contiguous()
timestep = torch.randn(
    1, 1, 9 * hidden, device=DEVICE, dtype=torch.bfloat16
).contiguous()

# fullgraph=True 强制整个函数被捕获进单一计算图，
# 任何隐式分支、额外分配或不可编译操作都会触发 graph break；
# 这能确保 slab + unbind 的返回路径可被 torch.compile 完整编译。
actual = torch.compile(ltx2_ada_values9, fullgraph=True)(
    scale_shift_table, timestep
)
expected = _reference(scale_shift_table, timestep)

assert len(actual) == 9
for actual_value, expected_value in zip(actual, expected):
    # 每个返回视图必须保持连续，避免上层按视图访问时发生
    # 非连续内存拷贝或性能回退
    assert actual_value.is_contiguous()
    torch.testing.assert_close(actual_value, expected_value, atol=0, rtol=0)
```

评论区精华

该 PR 没有收到任何 review 评论；Issue 评论区仅有作者 BBuf 的 /tag-and-rerun-ci、/rerun-failed-ci 指令和一条 CI 链接，属于流水线操作。PR body 自带的性能与精度数据（1.63x 提速、9 个输出逐位一致、fullgraph 编译通过）是唯一的决策依据，未出现技术争议或未解决疑虑。

- 暂无高价值评论线程

风险与影响

- 风险：
  - 内存布局语义变化：9 个输出从独立张量变为同一 slab 的视图，若 Triton kernel 对第 0 维 stride 的偏移计算有误，越界写会直接污染相邻视图；原实现中越界只影响单个张量。当前形状、dtype 一致且偏移简单，风险较低，但属于隐蔽的语义变化。
  - 视图连续性依赖：unbind(dim=0) 返回的视图连续与否取决于 slab 与 hidden 维的内存排布，测试中已有 is_contiguous() 断言兜底；若未来支持非连续布局需显式处理。
  - 测试环境局限：性能数据与 fullgraph 编译验证均在 B300 上完成，且 Extra CI 曾显示失败，普通 CI 环境未必覆盖该内核，回归保护强度依赖 B300 本地验证。
  - 影响：影响范围仅限 LTX-2 diffusion 模型推理路径：每个 transformer block 的内存分配开销显著下降，B300 场景 wrapper 提速约 1.63x，aten::empty 从 9 次降为 1 次，对批量、序列更大的形状同样受益；数值逐位不变，无精度影响。改动集中在 1 个内核文件与 1 个测试文件，对外 API 无变化，对团队协作影响很小，且新增的 fullgraph 编

译回归为后续 torch.compile 适配提供了保护。

- 风险标记：内存布局语义变更，依赖视图连续性，回归依赖 B300 验证

关联脉络

- PR #34349 [Diffusion] Tune QK head LayerNorm for SM120: 同属 diffusion Triton 内核性能优化，聚焦 SM120 平台内核启动开销，与本次分配优化目标一致。
- PR #34350 [Diffusion] Avoid slow cuBLASLt GELU epilogue on SM120: 同属 diffusion 内核性能优化，通过禁用慢速 cuBLASLt epilogue 减小算子开销，与本 PR 同处 diffusion 性能优化主线。
- PR #34314 [diffusion] Ideogram-4: fuse Qwen3-style RoPE and SwiGLU silu-mul (denoise -5.1% H100 / -4.7% H200, bit-exact): 同属 diffusion jit-kernel 融合优化，通过算子融合减少中间分配与内核启动，与本 PR 的减少 allocator 往返思路一致。