

PR #36231 完整报告

sgl-project/sglang

[DeepGEMM] Deduplicate JIT precompile across local ranks

合并时间: 2026-08-25 13:58

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/36231>

执行摘要

- 一句话: 跨本地 rank 去重 DeepGEMM JIT 预编译, 减少重复 NVCC 编译。
- 推荐动作: 值得精读, 该 PR 展示了通过简单文件锁解决多进程并发编译去重的思路, 设计简洁, 对理解多 rank 启动优化有参考价值。建议关注锁粒度和异常处理, 未来可考虑使用更细粒度的锁或异步编译。

功能与动机

当多个本地 rank 共享 DG_JIT_CACHE_DIR 时, 它们会惰性遍历相同的 (kernel_type, n, k, num_groups) 预编译组, 可能并发编译完全相同的 DeepGEMM kernel, 导致大量重复的 NVCC 编译工作。本 PR 旨在通过跨 rank 的编译锁消除这种重复, 加快多 rank 启动速度。

实现拆解

1. 在 compile_utils.py 中新增 _local_rank_compile_lock 上下文管理器, 利用 fcntl.flock 实现跨进程文件锁。
2. 锁文件存放在 DG_JIT_CACHE_DIR/locks 目录下, 文件名为 kernel 组唯一标识。
3. 在 _maybe_compile_deep_gemm_one_type_all 中, 将编译调用包裹在锁内, 确保同一组 kernel 仅由一个 rank 编译, 其他 rank 等待后复用缓存。
4. 未修改任何测试文件, 但该改动不改变编译结果, 仅影响编译并发性。

关键文件:

- python/sglang/srt/layers/deep_gemm_wrapper/compile_utils.py (模块 编译工具; 类别 source; 类型 core-logic; 符号 _local_rank_compile_lock): 核心改动文件, 新增文件锁机制控制 JIT 预编译并发。

关键符号: _local_rank_compile_lock

关键源码片段

[python/sglang/srt/layers/deep_gemm_wrapper/compile_utils.py](#)

核心改动文件, 新增文件锁机制控制 JIT 预编译并发。

```
# python/sglang/srt/layers/deep_gemm_wrapper/compile_utils.py
# 新增上下文管理器, 用于序列化跨 rank 的编译组
@contextmanager
```

```
def local_rank_compile_lock(
    kernel_type: DeepGemmKernelType, n: int, k: int, num_groups: int
):
    """
    序列化一个预编译组，跨共享 DG_JIT_CACHE_DIR 的多个本地 rank。

    每个 rank 惰性遍历相同的 (kernel_type, n, k, num_groups) 组，
    不锁时 N 个 rank 会对每个 kernel 编译 N 份副本。
    锁持有者将编译产物写入共享缓存；
    等待者发现 cubin 已存在后，仅执行 M 列表的 warmup 过程。
    """
    # 锁文件放在缓存目录下的 locks 子目录，确保各 rank 可见
    lock_dir = Path(os.environ["DG_JIT_CACHE_DIR"]) / "locks"
    lock_dir.mkdir(parents=True, exist_ok=True)
    # 锁文件名包含 kernel 类型和形状参数，保证不同组之间互不阻塞
    lock_path = lock_dir / f"{kernel_type.name}_n{n}_k{k}_g{num_groups}.lock"
    with open(lock_path, "w") as lock_file:
        fcntl.flock(lock_file, fcntl.LOCK_EX) # 获取排他锁
        try:
            yield
        finally:
            fcntl.flock(lock_file, fcntl.LOCK_UN) # 释放排他锁
```

评论区精华

无 review 评论。Approver Fridge003 直接批准，未提出额外问题。

- 暂无高价值评论线程

风险与影响

- 风险：
 1. 锁机制依赖 DG_JIT_CACHE_DIR 环境变量，若未设置可能导致 KeyError；但该变量本应存在，风险较低。
 2. 锁粒度较粗，仅按 kernel 组加锁，多个 rank 仍可能同时执行 warmup，但编译部分已串行，整体争用有限。
 3. 如果编译过程中进程崩溃，锁文件可能残留，但 fcntl.flock 锁会在进程退出时自动释放，且下次启动会重新创建文件，风险可忽略。
 4. 无测试覆盖多 rank 并发场景，但该改动不改变编译结果，回归风险较低。- 影响：影响范围限于 DeepGEMM JIT 预编译启动路径，主要影响多 rank 场景下的启动时间，对单 rank 或多个不共享缓存目录的场景无影响。用户可感知到多卡部署时启动时间缩短，稳态推理性能不变。对团队而言，减少了并发编译的资源损耗，提升了集群利用率。- 风险标记：无测试覆盖，依赖环境变量，锁粒度粗

关联脉络

- 暂无明显关联 PR