PR #24562 完整报告

sgl-project/sglang

Fix performance regression on Deepseek V3 on `moe-runner-backend=triton` on SM90

合并时间：2026-05-09 18:49

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/24562

执行摘要

- 一句话：修复 DeepSeek V3 Triton MoE 版本降级回归
- 推荐动作：建议合并，该 PR 修复了一个由 PyTorch/Triton 版本升级引起的隐性性能回归，改动小而精准，风险可控。合并后可考虑在相关测试中覆盖 Triton 3.6.0 环境以验证 fallback 效果。

功能与动机

PyTorch 升级到 2.11 后 Triton 3.6.0 尚未包含已调优的 fused_moe 配置，原有硬编码 fallback 列表只支持到 3.4.0，导致意外回退到 3.3.1（而非更优的 3.5.1），引发 DeepSeek V3 在 moe-runner-backend=triton 模式下的性能回归。PR 标题明确指向此修复。

实现拆解

1. 移除静态 fallback 版本列表（文件 fused_moe_triton_config.py）：删除第 60-61 行硬编码的 supported_triton_versions = ['3.4.0', '3.3.1', '3.2.0', '3.1.0']，消除列表不全导致的回退异常。
2. 引入动态版本发现机制：在 get_moe_configs 函数中，新增从 configs/ 目录动态扫描 triton_* 子目录的逻辑，提取版本号并按照语义版本降序排序，得到 available_versions 列表。
3. 替换 fallback 循环：将原先对 supported_triton_versions 的 for 循环替换为对 available_versions 的循环，路径构建改用 configs_root 变量（复用已拼接的 configs/ 路径）。此改动使 fallback 总是选择磁盘上存在的最高可用版本配置，而非固化的次优版本。
4. 无测试或配置配套变更：本次改动仅涉及 1 个源码文件，无配套测试或配置更新；但变更逻辑简单且风险较低。

关键文件：

- python/sglang/srt/layers/moe/moe_runner/triton_utils/fused_moe_triton_config.py（模块 MoE 调度；类别 source；类型 core-logic；符号 get_moe_configs）：PR 中唯一被修改的文件，核心逻辑变更：将 hardcoded fallback 版本列表替换为动态磁盘目录扫描。

关键符号：get_moe_configs

关键源码片段

python/sglang/srt/layers/moe/moe_runner/triton_utils/fused_moe_triton_config.py

PR 中唯一被修改的文件，核心逻辑变更：将 hardcoded fallback 版本列表替换为动态磁盘目录扫描。

```
@functools.lru_cache
def get_moe_configs(
    E: int,
    N: int,
    dtype: Optional[str],
    block_n: Optional[int] = 0,
    block_k: Optional[int] = 0,
    per_channel_quant: bool = False,
    down_moe: bool = False,
) -> Optional[Dict[int, Any]]:
    # ... 前置逻辑省略 ...

    # 原本此处存在硬编码列表：
    # supported_triton_versions = ["3.4.0", "3.3.1", "3.2.0", "3.1.0"]
    # 该列表缺少 3.6.0 导致 PyTorch 2.11 下回退错误；现已移除。

    # 新逻辑：动态扫描磁盘上的 triton_* 目录，按语义版本降序排列
    configs_root = os.path.join(config_dir, "configs")
    available_versions = sorted(
        (
            d.removeprefix("triton_").replace("_", ".")
            for d in os.listdir(configs_root)
            if d.startswith("triton_")
        ),
        key=lambda v: tuple(int(x) for x in v.split(".")),
        reverse=True, # 降序，优先选最新版本
    )

    # fallback 循环，优先使用磁盘上存在的最高版本配置
    for try_triton_version in available_versions:
        if try_triton_version == triton_version:
            continue
        try_config_file_path = os.path.join(
            configs_root,
            f"triton_{try_triton_version.replace('.', '_')}",
            json_file_name,
        )
        if os.path.exists(try_config_file_path):
            with open(try_config_file_path) as f:
                logger.warning(
                    f"Config file not found at {config_file_path}. "
                    f"Fallback to triton version {try_triton_version} and use MoE kernel config from
                    {try_config_file_path}. "
```

```
            "Performance might be sub-optimal!"
        )
        return {int(key): val for key, val in json.load(f).items()}
```

评论区精华

PR 代码审查评论数量为 0，合并者直接审批通过，未出现公开技术讨论。但 PR body 明确指出了问题根因和修复动机，且合并者 Fridge003 给予了批准。

- 暂无高价值评论线程

风险与影响

- 风险：
  - 回归风险低：变更仅影响 fallback 路径，核心逻辑（精确匹配当前 Triton 版本的配置）保持不变。动态扫描目录可能因 os.listdir 失败而提前退出，但原始代码已有 os.path.exists 检查作为保护，且最终 fallback 仍会退化到默认配置（return None），不会崩溃。
  - 性能影响正面：修复后的 fallback 会选择磁盘上可用的最高版本（如 3.5.1），避免非必要地回退到老旧版本，预期恢复因版本降级造成的性能损失。
  - 目录结构依赖：新逻辑要求 configs/ 目录存在且至少包含一个 triton_* 子目录；若目录为空或不存在，available_versions 为空列表，循环不会执行，随后会执行原有的 down_moe/ 默认配置回退，行为与旧代码一致。
- 影响：
  - 用户：使用 moe-runner-backend=triton 且 Triton 版本 ≥ 3.4.0（尤其是 3.6.0）的环境将自动采用更优的 3.5.1 配置，消除性能下降。对 Triton 版本低于 3.1.0 或使用非 Triton 后端的用户无影响。
  - 系统：单文件 14 行净增，无外部依赖或配置变更，可直接合并后生效。
  - 团队：修复无需迁移或改动其他模块，维护成本极低。
  - 风险标记：依赖目录结构，缺少测试覆盖

关联脉络

- PR #24793 [DSV4] Cherry pick missing commits from deepseek_v4 branch and enhance tests: 同为 DeepSeek 系列模型相关的 MoE 配置修复，体现持续性演进。