

PR #6244 完整报告

verl-project/verl

[perf, hardware] feat: NPU supports Liger-Kernel

合并时间: 2026-05-20 17:41

原文链接: <http://prhub.com.cn/verl-project/verl/pull/6244>

执行摘要

- 一句话: 为 Ascend NPU 集成 Liger-Kernel 加速
- 推荐动作: 该 PR 展示了如何在 NPU 后端集成第三方内核并优雅地管理算子冲突, 设计上优先复用已有配置字段, 值得推荐精读。建议增加 CI 测试覆盖 NPU 环境下的 Liger-Kernel 回归验证。

功能与动机

Liger-Kernel v0.8.0 首次支持 Ascend NPU, 但它与 Verl 默认的 `npu_patch` 算子替换冲突。用户需要一种方式在启用 Liger 时禁用 `npu_patch`, 同时保持向后兼容。

实现拆解

- 步骤 1: 在 `verl/workers/engine/fsdp/utils.py` 中修改 `apply_npu_fsdp_patches` 函数, 增加 `model_config` 参数; 当 `model_config` 中 `use_liger` 为 `True` 时提前返回, 跳过 `npu_patch` 导入。
- 步骤 2: 在 `verl/workers/engine/fsdp/transformer_impl.py` 中将调用 `apply_npu_fsdp_patches()` 改为 `apply_npu_fsdp_patches(self.model_config)`, 使配置判断生效。
- 步骤 3: 根据 review 建议, 移除最初引入的 `use_npu_patch_kernels` 配置字段, 直接复用 `model.use_liger` 控制。
- 步骤 4: 更新 Ascend 安装文档, 移除 `liger-kernel` 不支持条目, 添加新版本支持说明及与 `npu_patch` 互斥的提示。

关键文件:

- `verl/workers/engine/fsdp/utils.py` (模块 FSDP 引擎; 类别 source; 类型 core-logic; 符号 `apply_npu_fsdp_patches`): 核心逻辑变更: 修改 `apply_npu_fsdp_patches` 函数, 增加 `model_config` 参数; 当 `use_liger=True` 时跳过 `npu_patch`, 避免算子冲突。
- `verl/workers/engine/fsdp/transformer_impl.py` (模块 FSDP 引擎; 类别 source; 类型 core-logic): 调用点: 将 `self.model_config` 传递给 `apply_npu_fsdp_patches`, 使跳过逻辑生效。
- `docs/ascend_tutorial/get_start/install_guidance.rst` (模块 文档; 类别 docs; 类型 documentation): 文档更新: 移除 Liger-Kernel 不支持条目, 添加新版本支持说明以及与 `npu_patch` 的互斥关系。

关键符号: `apply_npu_fsdp_patches`

关键源码片段

`verl/workers/engine/fsdp/utils.py`

核心逻辑变更: 修改 `apply_npu_fsdp_patches` 函数, 增加 `model_config` 参数; 当 `use_liger=True` 时跳过 `npu_patch`, 避免算子冲突。

```
def apply_npu_fsdp_patches(model_config=None):
    """Apply NPU patches for FSDP backend if NPU is available.

    如果配置了 use_liger=True, 则跳过 npu_patch (因为 Liger-Kernel 会替换
    相同的算子, 同时启用会冲突)。
    """
    if is_npu_available:
        # 当 model_config 中存在 use_liger 且为 True 时, 跳过 npu_patch
        if model_config is not None and model_config.get("use_liger", False):
            return
        try:
            import verl.models.transformers.npu_patch # noqa

            if torch.distributed.is_initialized() and torch.distributed.get_rank() == 0:
                logger.info("Applied NPU patches for FSDP backend")
        except Exception as e:
            logger.warning(f"Failed to apply NPU patches: {e}")
```

评论区精华

- @wuxibin89 建议复用 `use_liger` 配置字段, 避免引入重复配置。最终 PR 采纳此建议, 删除了 `use_npu_patch_kernels`。
- @FightingZhen 询问默认参数设置, @zheliuyu 解释保持默认 `False` 以匹配原行为, 后续通过文档展示用法。
- gemini-code-assist[bot] 提醒需在文档中加入禁用 `npu_patch` 的步骤, 已完成。
- 文档应强调禁用 `npu_patch` (documentation): 作者随后提交了文档更新 (`install_guidance.rst`) 和代码变更 (`transformer_impl.py` 传递 `model_config`), 确保冲突可控。
- Liger-Kernel 默认参数设置讨论 (design): 维持默认 `False`, 通过配置选择启用, 确保向后兼容。
- 复用 `use_liger` 配置字段 (design): PR 最终采纳建议, 移除新增配置, 直接使用 `model.use_liger` 判断是否跳过 `npu_patch`。

风险与影响

- 风险:
 - 兼容性风险: 默认行为不变 (`use_liger=False`), 现有 NPU 用户无需修改配置。启用 Liger 的用户需确保安装正确版本的依赖 (`liger-kernel>=0.8.0`, `torch>=2.8.0`),

triton-ascend>=3.2.0) 。

- 冲突风险已规避：通过函数内的条件返回，确保 Liger 与 npu_patch 不会同时生效。
- 测试覆盖不足：PR 未包含自动化测试验证 Liger-Kernel 的集成，仅提供了示例脚本和实验数据。
- 影响：影响范围仅限 Ascend NPU + FSDP 引擎的训练路径。用户可通过 `model.use_liger=true` 开启加速，默认不开启，无向后兼容问题。变更涉及 3 个文件，改动量小，风险可控。
- 风险标记：默认 npu_patch 与 Liger-Kernel 冲突，依赖版本 (`liger-kernel>=0.8.0`)，配置项 `use_liger` 默认 `False`，无向后兼容风险

关联脉络

- 暂无明显关联 PR