PR #29030 完整报告

sgl-project/sglang

[NPU] use standalone group for moe ep

合并时间：2026-07-10 08:49

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/29030

执行摘要

- 一句话：NPU 使用独立 EP 通信组
- 推荐动作：值得合并。PR 聚焦、改动小、解决真实崩溃问题，且得到 reviewers 确认。建议后续补充 NPU 相关的单元测试。

功能与动机

NPU 硬件上需要独立的 EP 通信组，否则在长输入场景下会触发 dispatch 函数崩溃（见 Issue 评论中 Yuxin1999 的反馈）。

实现拆解

1. 导入依赖与检测 NPU：在 layer.py 中导入 get_moe_ep_group 和 is_npu，新增模块级变量 _is_npu = is_npu()。
2. 新增通信组选择函数 _get_deepep_comm_group：接收 a2a_backend 参数，默认返回 get_tp_group().device_group；若为 MORI 后端则返回 get_tp_group()（对象而非 device_group）；若为 NPU 则返回 get_moe_ep_group().device_group。
3. 更新 dispatcher 创建逻辑：create_moe_dispatcher 中构建 MaybeTboDeepEPDispatcher 时，将内联的通信组选择替换为调用 _get_deepep_comm_group(a2a_backend)。
4. 修改 EP 组初始化条件：在 parallel_state.py 中，将 _MOE_EP = _TP 的条件从 moe_ep_size == tensor_model_parallel_size 改为 moe_ep_size == tensor_model_parallel_size and not _is_npu，使 NPU 始终走 else 分支创建独立 EP 组。

关键文件：

- python/sglang/srt/layers/moe/fused_moe_triton/layer.py（模块 MoE 层；类别 source；类型 core-logic；符号 _get_deepep_comm_group）：核心变更：新增 _get_deepep_comm_group 函数，根据硬件选择 EP 通信组；修改 dispatcher 创建逻辑。
- python/sglang/srt/distributed/parallel_state.py（模块 分布式；类别 source；类型 core-logic）：修改 EP 组初始化条件，使 NPU 上始终创建独立 EP 组而非复用 TP 组。

关键符号：_get_deepep_comm_group

关键源码片段

python/sglang/srt/layers/moe/fused_moe_triton/layer.py

核心变更：新增 _get_deepep_comm_group 函数，根据硬件选择 EP 通信组；修改 dispatcher 创建逻辑。

```
# python/sglang/srt/layers/moe/fused_moe_triton/layer.py

from sglang.srt.distributed import (
    get_moe_ep_group, # 新增导入，用于 NPU 独立 EP 组
    get_tp_group,
    ...
)
from sglang.srt.utils import (
    is_npu, # 新增导入，检测 NPU 硬件
    ...
)

_is_hip = is_hip()
_is_cpu_amx_available = cpu_has_amx_support()
_is_cpu = is_cpu()
_is_npu = is_npu() # 模块级标志，用于后续分支判断
_use_aiter = get_bool_env_var("SGLANG_USE_AITER") and _is_hip


def _get_deepep_comm_group(a2a_backend):
    # 默认使用 TP 组的 device_group
    group = get_tp_group().device_group

    if a2a_backend.is_mori():
        # MORI 后端需要 TP 组对象本身
        group = get_tp_group()
    elif _is_npu:
        # NPU 要求独立的 EP 通信组
        group = get_moe_ep_group().device_group

    return group


def create_moe_dispatcher(moe_runner_config: MoeRunnerConfig) -> BaseDispatcher:
    a2a_backend = get_moe_a2a_backend()
    ...
    elif ...:
        return MaybeTboDeepEPDispatcher(
            group=_get_deepep_comm_group(a2a_backend), # 原内联逻辑提取为函数调用
            ...
        )
```

python/sglang/srt/distributed/parallel_state.py

修改 EP 组初始化条件，使 NPU 上始终创建独立 EP 组而非复用 TP 组。

```
# python/sglang/srt/distributed/parallel_state.py

    global _MOE_EP
    assert _MOE_EP is None, "expert model parallel group is already initialized"
    # NPU 要求独立的 EP 组，即使 moe_ep_size 等于 tp_size
    if moe_ep_size == tensor_model_parallel_size and not _is_npu:
        _MOE_EP = _TP # 非 NPU 时复用 TP 组
    else:
        # NPU 或其他情况：创建独立 EP 组
        group_ranks = []
        ...
        _MOE_EP = init_model_parallel_group(
            group_ranks,
            ...
            group_name="moe_ep",
        )
```

评论区精华

无实质性 review 讨论。whybeyoung 批准 PR，Yuxin1999 在 Issue 中反馈该 PR 修复了 910B3 上长输入 dispatch 崩溃。

- 暂无高价值评论线程

风险与影响

- 风险：风险较低：仅影响 NPU 硬件路径，且原有 TP 组复用逻辑在非 NPU 上保持不变。但未添加对应单元测试，回归依赖 CI。
- 影响：影响 NPU 上的 MoE 模型推理，修复长输入场景下的崩溃问题。对其他硬件（CUDA、CPU 等）无影响。
- 风险标记：缺少测试覆盖

关联脉络

- PR #30716 [Diffusion] Revert CPU AMX optimizations: 同为硬件适配（NPU vs CPU），共享硬件检测语境。