PR #23996 完整报告

sgl-project/sglang

[parallel] Support moe_dense_tp_size == attn_tp_size to share the attention TP group

合并时间：2026-05-30 17:10

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/23996

执行摘要

- 一句话：支持 dense MLP 与 attention 共享 TP 组的并行
- 推荐动作：值得精读，尤其是 communicator.py 中新增的通信模式。设计上保持了与现有 _simple / _gather_* 分支的一致性，代码简洁。建议后续补充单元测试，并考虑在文档中明确 moe_dense_tp_size 的取值范围和限制。

功能与动机

SGLang 之前强制 moe_dense_tp_size 只能为 1 或 None，阻止了 dense MLP 沿着 attention TP 组进行张量并行（moe_dense_tp_size == attn_tp_size > 1）。对于 MoE 模型，dense 层如果能与 attention 共享 TP 组，可以减少通信开销、提高资源利用率。PR body 明确说明需要解除这三处守卫，使该配置开箱可用。

实现拆解

1. server_args.py：将 check_server_args 中 moe_dense_tp_size in {1, None} 的断言改为 (None, 1, self.tp_size)，允许合法值为 None、1 或 tp_size。
2. common.py：修改 require_attn_tp_gather 函数，移除 assert server_args.moe_dense_tp_size in [1, None]；并将条件 moe_dense_tp_size == 1 改为 moe_dense_tp_size is not None，使所有非 None 的值都触发 attention TP gather 路径。
3. communicator.py：在 CommunicateWithAllReduceAndLayerNormFn.get_fn 中新增一个条件分支，当输入 / 输出均为 TP_ATTN_FULL 且 attn_tp_size > 1 时，返回新静态方法 _tp_attn_all_reduce_and_layernorm。该方法在 attention TP 组内对 hidden states 做 all-reduce，然后执行 layernorm。

配置验证限于 moe_dense_tp_size == attn_tp_size，其他大于 1 的值走现有 require_mlp_tp_gather 分支，未在本 PR 中测试。

关键文件：

- python/sglang/srt/layers/communicator.py（模块 通信层；类别 source；类型 core-logic；符号 _tp_attn_all_reduce_and_layernorm, get_fn）：新增 _tp_attn_all_reduce_and_layernorm 方法，实现了 dense MLP 与 attention 共享 TP 组时的通信路径，是本 PR 的核心逻辑变更。
- python/sglang/srt/server_args.py（模块 服务参数；类别 source；类型 configuration；符号 check_server_args）：修改了 moe_dense_tp_size 的校验规则，从仅允许 1/None

扩展到允许 tp_size，是配置层面的关键变更。

- python/sglang/srt/utils/common.py（模块 公共工具；类别 source；类型 core-logic；符号 require_attn_tp_gather）：修改 require_attn_tp_gather 函数，移除断言并调整条件，使所有非 None 的 moe_dense_tp_size 都触发 attention TP gather。

关键符号：_tp_attn_all_reduce_and_layernorm, get_fn, require_attn_tp_gather, check_server_args

关键源码片段

python/sglang/srt/layers/communicator.py

新增 _tp_attn_all_reduce_and_layernorm 方法，实现了 dense MLP 与 attention 共享 TP 组时的通信路径，是本 PR 的核心逻辑变更。

```
# python/sglang/srt/layers/communicator.py, line ~929

def get_fn(...):
    # ... existing branches ...

    # 新增分支：当 hidden_states 和 residual 均为 TP_ATTN_FULL 且 attn_tp_size > 1 时，
    # 使用 all-reduce + layernorm 代替 gather，避免不必要的数据重排。
    # 这是 dense MLP 与 attention 共享 TP 组时的关键路径。
    if (
        (hidden_states_input_mode == ScatterMode.TP_ATTN_FULL)
        and (residual_input_mode in [ScatterMode.SCATTERED, ScatterMode.TP_ATTN_FULL])
        and (hidden_states_output_mode == ScatterMode.TP_ATTN_FULL)
        and (residual_output_mode == ScatterMode.TP_ATTN_FULL)
        and context.attn_tp_size > 1
    ):
        return (
            CommunicateWithAllReduceAndLayerNormFn._tp_attn_all_reduce_and_layernorm
        )

    raise NotImplementedError(...)

@staticmethod
def _tp_attn_all_reduce_and_layernorm(
    hidden_states: torch.Tensor,
    residual: torch.Tensor,
    forward_batch: ForwardBatch,
    layernorm: torch.nn.Module,
    context: CommunicateContext,
):
    """在 attention TP 组内对 hidden states 做 all-reduce，然后 layernorm。

    当 dense MLP 共享 attention TP 组（moe_dense_tp_size > 1）时使用：
    输入和输出都保持在 TP_ATTN_FULL 模式。
    """
```

```
    hidden_states = get_attention_tp_group().all_reduce(hidden_states)
    if hidden_states.shape[0] != 0:
        hidden_states, residual = layernorm(hidden_states, residual)
    return hidden_states, residual
```

python/sglang/srt/server_args.py

修改了 moe_dense_tp_size 的校验规则，从仅允许 1/None 扩展到允许 tp_size，是配置层面的关键变更。

```
# python/sglang/srt/server_args.py, line ~7112
# 在 check_server_args 方法中：
# 原断言：moe_dense_tp_size in {1, None}
# 改为允许 tp_size，使 dense MLP 可以与 attention 共享 TP 组。
assert self.moe_dense_tp_size in (
    None,
    1,
    self.tp_size,
), "moe_dense_tp_size only supports None, 1, or tp_size currently"
```

python/sglang/srt/utils/common.py

修改 require_attn_tp_gather 函数，移除断言并调整条件，使所有非 None 的 moe_dense_tp_size 都触发 attention TP gather。

```
# python/sglang/srt/utils/common.py, line ~3014
def require_attn_tp_gather(server_args: ServerArgs):
    """判断 attention 输入是否需要 gather（即是否分散）。"""
    from sglang.srt.layers.moe.utils import get_moe_a2a_backend

    # 原代码：assert server_args.moe_dense_tp_size in [1, None];
    # 然后 if not ... or server_args.moe_dense_tp_size == 1:
    # 修改后：移除断言，条件变为 is not None,
    # 使得 moe_dense_tp_size 为 tp_size 时也能正确进入 gather 路径。
    if not get_moe_a2a_backend().is_none() or server_args.moe_dense_tp_size is not None:
        if server_args.enable_dp_attention:
            return server_args.dp_size < server_args.tp_size
        else:
            return True
    else:
        return False
```

评论区精华

Reviewer ch-wan 确认了 PR 的意图——是允许任意 moe_dense_tp_size 还是仅 moe_dense_tp_size == attn_tp_size。作者 brucechanglongxu 澄清为后者，随后更新了标题和描述。ch-wan 也对 common.py 中 >=1 的注释提出修正意见，最终实现采用了 is not None 的精确表达。

- PR 意图确认：任意 moe_dense_tp_size 还是仅等于 attn_tp_size (design): 明确 PR 范围限于 moe_dense_tp_size == attn_tp_size，其他值未经测试但未完全禁止。

- common.py 中条件表达式正确性 (correctness): 代码最终采用 is not None 取代 >= 1。

风险与影响

- 风险：1）communicator.py 中的新分支仅在 context.attn_tp_size > 1 时激活，对原有路径（attn_tp_size==1）无影响。2）server_args 放宽后，用户可能误设 moe_dense_tp_size 为其他值（如 2 但 attn_tp_size=4），由于 require_mlp_tp_gather 未完全验证，可能产生未定义的通信行为。PR 已声明其他值未被测试。3）未添加单元测试，回归风险依赖手动验证。
- 影响：对使用 MoE 模型并希望 dense 层与 attention 共享 TP 组的用户是正向影响，可提升并行效率和内存利用率。影响范围限于设置 moe_dense_tp_size == attn_tp_size > 1 的场景，默认行为不变。团队需要关注后续若支持更多 moe_dense_tp_size 值时的测试覆盖。
- 风险标记：缺少测试覆盖，配置放宽但未充分验证

关联脉络

- 暂无明显关联 PR