PR #24949 完整报告

sgl-project/sglang

Deepseek-v4-Pro share expert tp1

合并时间：2026-05-12 14:19

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/24949

执行摘要

- 一句话：为 DeepSeek-V4 共享专家添加 TP1 部署支持
- 推荐动作：推荐阅读，特别是关注共享专家 TP1 部署时前向传播中 all-reduce 与共享输出叠加的顺序设计。这个模式值得在类似大模型部署中参考。

功能与动机

官方 DeepSeek-V4 共享专家权重（如 scales）的某些维度无法被全局 TP 大小（如 16）整除，导致 TP16 模式下根本无法加载模型。此 PR 让共享专家可以使用 TP1 部署（每卡完整副本），从而绕过整除限制，同时保留路由专家的 TP 并行以维持性能。

实现拆解

1. 注册环境变量 SGLANG_SHARED_EXPERT_TP1 - 在 python/sglang/srt/environ.py 中添加 SGLANG_SHARED_EXPERT_TP1 = EnvBool(False)，默认关闭，作为门控开关（文件 environ.py）。
2. 重构共享专家实例化条件 - 在 python/sglang/srt/models/deepseek_v2.py 的 __init__ 中，将原来内联到 dict(tp_rank=0, tp_size=1) 的条件抽取为独立变量 _shared_expert_use_tp1，新增envs.SGLANG_SHARED_EXPERT_TP1.get() 作为可选项。 - 同时新增实例属性 _shared_expert_tp1 记录该决策，供前向传播使用。
3. 调整前向传播中的共享专家输出合并方式 - 在 forward_normal_dual_stream 和 _post_combine_hook 中：
   - 调用 maybe_fuse_routed_scale_and_shared_add 时，若 _shared_expert_tp1 为 True，则传入 None 作为 shared_output，避免在 all-reduce 前融合。
   - 在 tensor_model_parallel_all_reduce 之后，若 _shared_expert_tp1，再将 shared_output 加到 final_hidden_states 上。
   - 这是因为 TP1 下每个 rank 都有完整共享专家输出，若在 all-reduce 前融合会导致每份输出被累加 tp_size 次。
4. 放宽模型加载时的整除兼容性检查-在python/sglang/srt/model_executor/model_runner.py 的 check_quantized_moe_compatibility 中，当 SGLANG_SHARED_EXPERT_TP1 启用时，跳过 moe_intermediate_size // moe_tp_size 对 weight_block_size_n 的整除检查，因为共享专家已不参与 TP 切分。

关键文件：

- python/sglang/srt/models/deepseek_v2.py（模块 模型层；类别 source；类型 core-logic）：核心变更文件：修改共享专家实例化的条件，新增 _shared_expert_tp1 属性，并调整前向传播中共享输出的合并时机。+26/-14。
- python/sglang/srt/environ.py（模块 配置层；类别 source；类型 configuration）：定义新环境变量 SGLANG_SHARED_EXPERT_TP1，作为 TP1 部署的门控开关。+1/-1。
- python/sglang/srt/model_executor/model_runner.py（模块 模型运行器；类别 source；类型 data-contract）：修改 check_quantized_moe_compatibility，当 SGLANG_SHARED_EXPERT_TP1 启用时跳过整除检查。+4/-2。

关键符号：forward_normal_dual_stream, _post_combine_hook, check_quantized_moe_compatibility

关键源码片段

python/sglang/srt/models/deepseek_v2.py

核心变更文件：修改共享专家实例化的条件，新增 _shared_expert_tp1 属性，并调整前向传播中共享输出的合并时机。+26/-14。

```
# python/sglang/srt/models/deepseek_v2.py ( 关键片段 )

# 在 __init__ 中提取共享专家 TP1 决策逻辑
self._shared_expert_tp1 = False # 新增属性
...
_shared_expert_use_tp1 = (
    get_moe_a2a_backend().is_deepep()
    or get_moe_a2a_backend().is_mooncake()
    or get_moe_a2a_backend().is_nixl()
    or get_moe_a2a_backend().is_mori()
    or get_moe_a2a_backend().is_ascend_fuseep()
    or get_moe_a2a_backend().is_flashinfer()
    or should_use_flashinfer_cutlass_moe_fp4_allgather()
    or envs.SGLANG_SHARED_EXPERT_TP1.get() # 新增：环境变量扩展 TP1 范围
)
self.shared_experts = DeepseekV2MLP(
    ...,
    **(dict(tp_rank=0, tp_size=1) if _shared_expert_use_tp1 else {}),
)
self._shared_expert_tp1 = _shared_expert_use_tp1 # 记录决策

# 前向传播：TP1 下共享输出在 all-reduce 后叠加
# forward_normal_dual_stream 中的关键调整
final_hidden_states = maybe_fuse_routed_scale_and_shared_add(
    self.experts, final_hidden_states,
    None if self._shared_expert_tp1 else shared_output, # TP1 时跳过融合
    self.routed_scaling_factor,
)
...# 之后
final_hidden_states = tensor_model_parallel_all_reduce(final_hidden_states)
```

```
if self._shared_expert_tp1: # TP1 共享输出需在 all-reduce 之后叠加
    final_hidden_states += shared_output

# 类似的逻辑也存在于 `_post_combine_hook` 中
```

python/sglang/srt/environ.py

定义新环境变量 SGLANG_SHARED_EXPERT_TP1，作为 TP1 部署的门控开关。+1/-1。

```
# python/sglang/srt/environ.py ( 新增环境变量 )
class Envs:
    ...
    # Distributed
    SGLANG_DSV4_FIX_TP_ATTN_A2A_SCATTER = EnvBool(True)
    SGLANG_SHARED_EXPERT_TP1 = EnvBool(False) # 新增：强制共享专家使用 TP1 部署
    # Symmetric Memory
    ...
```

python/sglang/srt/model_executor/model_runner.py

修改 check_quantized_moe_compatibility，当 SGLANG_SHARED_EXPERT_TP1 启用时跳过整除检查。+4/-2。

```
# python/sglang/srt/model_executor/model_runner.py (check_quantized_moe_compatibility)
if (
    not envs.SGLANG_SHARED_EXPERT_TP1.get() # 新增：TP1 模式下跳过整除检查
    and (moe_intermediate_size // moe_tp_size) % weight_block_size_n != 0
    and not _use_aiter
):
    raise ValueError(
        f"For quantized MoE models, please make sure ({moe_intermediate_size=} / {moe_tp_size=})
        % {weight_block_size_n=} == 0 "
        ...
    )
```

评论区精华

只有一条 Review 评论来自 gemini-code-assist[bot]：建议在 forward_normal_dual_stream 的 if self._shared_expert_tp1: final_hidden_states += shared_output 之前增加 if shared_output is not None 判断，以保持与 _post_combine_hook 风格一致并防止 shared_output 为 None 时出错。该评论未被采纳（可能因为调用处已确保 shared_output 有效），最终 reviewer Fridge003 直接 APPROVED 且未要求修改。

- 前向传播中 shared_output None 检查 (correctness): 未被采纳；reviewer Fridge003 直接 APPROVED，未要求修改。可能是因为调用处已确保 shared_output 有效（如 shared_experts 未初始化时不会进入该路径）。

风险与影响

- 风险：低风险。变更仅在 SGLANG_SHARED_EXPERT_TP1 环境变量为 True 时生效，默认行为完全不变。主要风险点：若 forward_normal 路径（非 dual_stream）未同步修改，但

实际代码中 forward_normal 调用 _post_combine_hook，已包含该修改。此外，若未来引入新的前向路径，需确保重复正确的条件判断。

- 影响：影响范围局限于 DeepSeek-V4 模型在 TP>1 且共享专家 weight_block_size 不整除时的场景。对用户而言，通过设置环境变量即可部署之前无法加载的模型。对系统无性能负面影响，因为 TP1 共享专家增加了每卡计算量但省去了 all-reduce 通信。
- 风险标记：核心路径变更，缺少测试覆盖

关联脉络

- PR #23911 DeepSeekV4 branch PR: PR body 中提及此为其 DeepSeekV4 分支 PR，功能相同但针对不同分支。