PR #25391 完整报告

sgl-project/sglang

Support DeepSeek V4 DeepEP Waterfill

合并时间：2026-05-26 12:04

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25391

执行摘要

- 一句话：DeepSeek V4 整合 DeepEP Waterfill 负载均衡
- 推荐动作：值得精读。本 PR 展示了如何将 DeepEP Waterfill 负载均衡集成到 DeepSeek V4 的 HashTopK 路由中，设计上保持了 shared-expert fusion 并扩展了 balancer 接口，对其他 MoE 模型的类似集成有参考价值。

功能与动机

DeepEP Waterfill 通过将共享专家负载均衡到最空闲的 EP rank 上，可以减少通信瓶颈并提升吞吐。DeepSeek V4 原有的 shared-expert fusion 因 clamp 差异被禁用，但 Waterfill 需要保持 fusion 以利用共享专家分发。为此需要适配 DeepSeek V4 的 HashTopK 路由逻辑和 expert distribution 记录。

实现拆解

1. HashTopK 增加 Waterfill 支持(python/sglang/srt/layers/moe/hash_topk.py): 在 __init__ 中根据 num_fused_shared_experts > 0 和 enable_deepep_waterfill 标志设置 enable_deepep_waterfill，并保留 balancer 属性；在 forward 和 empty_topk_output 末尾调用 _apply_deepep_waterfill 方法，通过 DeepEPWaterfillBalancer.expand_topk 将共享专家追加到 topk 输出中。
2. DeepSeek V4 保留 shared-expert fusion(python/sglang/srt/models/deepseek_v4.py): 在 determine_num_fused_shared_experts 中，当 enable_deepep_waterfill 为 True 且 n_shared_experts==1 时，不禁用 fusion，而是保留 num_fused_shared_experts = 1，使 shared expert 可通过路由参与 Waterfill 均衡。
3. ModelRunner 准备 HashTopK 的 Waterfill(python/sglang/srt/model_executor/model_runner.py): 在 _prepare_moe_topk 中新增对 HashTopK 实例的识别，统一提取 routed_scaling_factor（从 module.routed_scaling_factor 或 module.topk_config.routed_scaling_factor），并创建 DeepEPWaterfillBalancer 分配给模块。
4. TopK 简化 Waterfill 判定(python/sglang/srt/layers/moe/topk.py): 移除不必要的 try-except，直接使用 num_fused_shared_experts > 0 and get_global_server_args().enable_deepep_waterfill 设置标志。

关键文件：

- python/sglang/srt/layers/moe/hash_topk.py（模块 MoE 路由；类别 source；类型 dependency-wiring；符号 _apply_deepep_waterfill）：核心修改：新增 Waterfill 支持，包括初始化、_apply_deepep_waterfill 方法，以及 forward/empty_topk_output 的出口扩展。
- python/sglang/srt/models/deepseek_v4.py（模块 DeepSeek V4；类别 source；类型 data-contract）：决策关键：在 determine_num_fused_shared_experts 中为 Waterfill 保留 shared-expert fusion，打破原有限制。
- python/sglang/srt/model_executor/model_runner.py（模块 模型运行；类别 source；类型 data-contract）：初始化 WaterfillBalancer 的入口，新增对 HashTopK 实例的处理。
- python/sglang/srt/layers/moe/topk.py（模块 MoE 路由；类别 source；类型 dependency-wiring）：简化 Waterfill 判定逻辑，移除 try-except，与 HashTopK 保持一致。

关键符号：_apply_deepep_waterfill, determine_num_fused_shared_experts, _prepare_moe_topk

关键源码片段

python/sglang/srt/layers/moe/hash_topk.py

核心修改：新增 Waterfill 支持，包括初始化、_apply_deepep_waterfill 方法，以及 forward/empty_topk_output 的出口扩展。

```
# python/sglang/srt/layers/moe/hash_topk.py
# 在 __init__ 中根据配置开启 Waterfill 模式
class HashTopK(nn.Module):
    def __init__(self, topk, num_experts, num_fused_shared_experts, vocab_size, ...):
        super().__init__()
        self.layer_id = None
        from sglang.srt.server_args import get_global_server_args

        # 仅当存在 fused shared expert 且全局 waterfill 启用时生效
        self.enable_deepep_waterfill = (
            num_fused_shared_experts > 0
            and get_global_server_args().enable_deepep_waterfill
        )
        self.deepep_waterfill_balancer = None

        if self.enable_deepep_waterfill:
            # Waterfill 将 shared expert 作为额外路由专家，因此从 topk 中扣除
            topk -= num_fused_shared_experts
            num_fused_shared_experts = 0
        # ... 其余初始化

    # 在 empty_topk_output 和 forward 末尾调用此方法
    def _apply_deepep_waterfill(
        self, topk_output: StandardTopKOutput, num_tokens: int
    ) -> StandardTopKOutput:
```

```
        # 若启用但未在 ModelRunner 中准备 balancer，则报错
        if self.enable_deepep_waterfill and self.deepep_waterfill_balancer is None:
            raise RuntimeError(
                "DeepEP waterfill HashTopK must be prepared by ModelRunner before forward."
            )
        if self.deepep_waterfill_balancer is None:
            return topk_output
        # 通过 balancer 将 shared expert 追加到 topk 输出中
        return self.deepep_waterfill_balancer.expand_topk(topk_output, num_tokens)
```

python/sglang/srt/models/deepseek_v4.py

决策关键：在 determine_num_fused_shared_experts 中为 Waterfill 保留 shared-expert fusion，打破原有限制。

```
    # python/sglang/srt/models/deepseek_v4.py
    # 在 determine_num_fused_shared_experts 中增加 Waterfill 分支
        def determine_num_fused_shared_experts(self):
            self.num_fused_shared_experts = 0
            if get_global_server_args().disable_shared_experts_fusion:
                return

            # Waterfill 需要 shared-expert fusion 以将 shared expert 分发给空闲 EP
            if get_global_server_args().enable_deepep_waterfill:
                if self.config.n_shared_experts != 1:
                    raise ValueError(
                        "DeepEP Waterfill for DeepSeek V4 expects exactly one shared "
                        f"expert, but got n_shared_experts={self.config.n_shared_experts}."
                    )
                # 保留 fusion，设置 num_fused_shared_experts 为 1
                self.num_fused_shared_experts = self.config.n_shared_experts
                log_info_on_rank0(logger,
                    "DeepSeek V4: --enable-deepep-waterfill set; KEEP shared-experts "
                    "fusion enabled so waterfill can rebalance shared expert dispatch.")
                return

            # 默认行为：V4 因 clamp 差异禁用 fusion
            get_global_server_args().disable_shared_experts_fusion = True
            log_info_on_rank0(logger, "Shared experts fusion optimization is disabled.")
```

评论区精华

- try-except 简化(ch-wan 提问，xutizhou 建议)：ch-wan 质疑 HashTopK 中是否需要 try-except，xutizhou 建议直接使用 num_fused_shared_experts > 0 and get_global_server_args().enable_deepep_waterfill 简化逻辑，最终采纳该方案。
- 整数除法安全性(gemini-code-assist[bot] 建议)：在 topk.py 的 _remap_topk_for_deepep 中，将 n_routed_experts // ep_size 改为 num_physical_routed_experts // ep_size，但未添加整除断言；reviewer 建议添加断言以确保安全。

- HashTopK 中 try-except 的简化 (design): 简化为 self.enable_deepep_waterfill = (num_fused_shared_experts > 0 and get_global_server_args().enable_deepep_waterfill)，移除了 try-except。
- 整数除法安全性与断言 (correctness): 已改为使用 num_physical_routed_experts 计算，但未添加断言；讨论未进一步处理。

风险与影响

- 风险:
  1. 异常路径风险: 若 n_shared_experts != 1 但启用了 waterfill，DeepSeek V4 会抛出 ValueError（已处理），但用户可能未预期此行为。
  2. 整数整除风险: 在 topk.py 的 _remap_topk_for_deepep 中，num_physical_routed_experts // ep_size 假设整除，当 redundant experts 导致不能整除时可能产生错误；review 建议添加断言但未落实。
  3. 性能影响: Waterfill 增加一次 balancer 调用（expand_topk），但吞吐测试显示正向收益 2-4%，风险可控。 - 影响：-用户影响: 仅影响使用 --enable-deepep-waterfill 的 DeepSeek V4 用户，提供 2-4% 吞吐提升，正确性经过 MMLU 验证（分数一致）。 - 系统影响: 修改 MoE 路由核心路径（HashTopK 和 TopK），需确保其他模型（如 DeepSeek V3、Qwen 等）不受影响，因为 enable_deepep_waterfill 仅在新参数下生效。 - 团队协作: 依赖两个 bugfix PR（#25285, #25367），需依次合并。 - 风险标记：核心 MoE 路由变更，缺少兼容性测试，整数整除假设

关联脉络

- PR #25285 [bugfix] Fix ...: 本 PR 依赖的 bugfix PR，修复了 waterfill 相关的前置问题。
- PR #25367 [bugfix] Fix ...: 本 PR 依赖的另一个 bugfix PR，需要先合并以确保正确性。