PR #24906 完整报告

sgl-project/sglang

Support Qwen3.5 NVFP4 MTP DeepEP

合并时间：2026-05-15 10:49

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/24906

执行摘要

- 一句话：支持 Qwen3.5 NVFP4 MTP 与 DeepEP 低延迟模式
- 推荐动作：此 PR 涉及 DeepEP 低延迟模式与 MTP 的集成，以及 GPU 架构感知的 verify 内核选择，设计取舍值得关注。建议负责 DeepEP 和推测解码的工程师精读，特别是 forward_unquantized_deepep_ll 的 fallback 实现和 bf16_dispatch 的配置传播。

功能与动机

Qwen3.5 NVFP4 disaggregated serving with DeepEP and NEXTN/MTP currently fails in the MTP draft path on GB300/SM100+. Three issues:

1) MTP draft MoE can have quant_config=None; 2) DeepEP low-latency dispatch needs kNumMaxTopK >= 10; 3) FlashInfer GDN target verify is not supported on SM100+. This PR adds Python-side fixes for issues 1 and 3.

实现拆解

1. 处理未量化 draft MoE 层：在 ep_moe/layer.py 的 __init__ 中，当 quant_config is None 时通过 dispatcher.set_quant_config({"bf16_dispatch": True}) 强制 BF16 分发；在 run_moe_core 中增加 quant_config is None 分支以避免调用 quant_config.get_name()；新增 forward_unquantized_deepep_ll 方法执行 w13 -> silu(gate)*up -> w2 并屏蔽 padding 行。
2. 支持显式 BF16 分发配置：在 token_dispatcher/deepep.py 的 _dispatch_core 中读取 bf16_dispatch 键并优先于其他条件判断，确保 draft 层以 BF16 格式分发隐藏状态。
3. 限定 FlashInfer GDN verify 支持范围：在 kernels/gdn_flashinfer.py 中添加 supports_target_verify 属性，基于 SM major 决定（SM90 支持，SM100+ 不支持）。
4. 更新 verify kernel 选择逻辑：在 gdn_backend.py 中增加 flashinfer_kernel.supports_target_verify 检查，仅在确认支持时使用 FlashInfer 验证内核，否则回退到 Triton。
5. 验证：通过 Qwen3.5 NVFP4 GPQA 端到端测试确认正确性，但未添加新的单元测试。

关键文件：

- python/sglang/srt/layers/moe/ep_moe/layer.py（模块 MoE 层；类别 source；类型 core-logic；符号 forward_unquantized_deepep_ll）：核心改动：处理

quant_config=None，新增 forward_unquantized_deepep_ll 方法，实现未量化 DeepEP 低延迟 MoE 前向。

- python/sglang/srt/layers/attention/linear/gdn_backend.py（模块 注意力层；类别 source；类型 core-logic）：修改 verify kernel 选择逻辑，添加 supports_target_verify 检查以在 SM100+ 上回退到 Triton。
- python/sglang/srt/layers/moe/token_dispatcher/deepep.py（模块 分发器；类别 source；类型 core-logic）：支持 bf16_dispatch 配置键，使得 draft 层能被 DeepEP 以 BF16 分发。
- python/sglang/srt/layers/attention/linear/kernels/gdn_flashinfer.py（模块 注意力层；类别 source；类型 core-logic）：添加 supports_target_verify 属性，基于 SM major 决定 FlashInfer 是否可用于 MTP verify。

关键符号：forward_unquantized_deepep_ll, GDNBackend.init, _dispatch_core, FlashInferGDNKernel.init

关键源码片段

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

核心改动：处理 quant_config=None，新增 forward_unquantized_deepep_ll 方法，实现未量化 DeepEP 低延迟 MoE 前向。

```
# Inside run_moe_core - handling DeepEP low-latency dispatch output
elif DispatchOutputChecker.format_is_deepep_ll(dispatch_output):
    if self.quant_config is None:
        # MTP draft MoE layer without quant config: use BF16 fallback
        output = self.forward_unquantized_deepep_ll(dispatch_output)
    elif (
        get_moe_runner_backend().is_flashinfer_cutedsl()
        and self.quant_config is not None
        and self.quant_config.get_name() == "modelopt_fp4"
    ):
        output = self.forward_flashinfer_cutedsl(dispatch_output)
    else:
        # fallback to other forward paths
        ...


def forward_unquantized_deepep_ll(
    self,
    dispatch_output: DeepEPLLDispatchOutput,
):
    # BF16 fallback for DeepEP low-latency mode when no quant config is present
    hidden_states, hidden_states_scale, _, _, masked_m, _ = dispatch_output
    assert hidden_states_scale is None
    assert self.moe_runner_config.activation == "silu"
    assert self.moe_runner_config.is_gated
    assert hidden_states.dim() == 3

    num_experts, max_tokens, _ = hidden_states.shape
```

```
    token_offsets = torch.arange(max_tokens, device=hidden_states.device)
    # Create mask for valid tokens per expert (ignore padding rows)
    valid_mask = (
        token_offsets.unsqueeze(0) < masked_m[:num_experts].unsqueeze(1)
    ).unsqueeze(-1)
    hidden_states = hidden_states.masked_fill(~valid_mask, 0)

    # Gate-up projection: (E, T, dim) -> (E, T, 2 * inter_dim)
    gate_up = torch.bmm(hidden_states, self.w13_weight.transpose(1, 2))
    w13_bias = getattr(self, "w13_weight_bias", None)
    if w13_bias is not None:
        gate_up += w13_bias.unsqueeze(1)

    gate, up = gate_up.chunk(2, dim=-1)
    hidden_states = F.silu(gate) * up

    # Down projection
    output = torch.bmm(hidden_states, self.w2_weight.transpose(1, 2))
    w2_bias = getattr(self, "w2_weight_bias", None)
    if w2_bias is not None:
        output += w2_bias.unsqueeze(1)

    # Zero out padding rows in output
    return output.masked_fill(~valid_mask, 0)
```

python/sglang/srt/layers/attention/linear/gdn_backend.py

修改 verify kernel 选择逻辑，添加 supports_target_verify 检查以在 SM100+ 上回退到 Triton。

```
# Verify kernel: use FlashInfer only when the selected FlashInfer kernel
# supports MTP verify. On SM100+ FlashInfer GDN decode is supported, but
# its MTP verify path is not, so keep Triton as the verify fallback.
if (
    decode_backend.is_flashinfer() or prefill_backend.is_flashinfer()
) and flashinfer_kernel.supports_target_verify:
    self.verify_kernel = flashinfer_kernel
else:
    self.verify_kernel = triton_kernel
```

评论区精华

> gemini-code-assist[bot]建议优化 forward_unquantized_deepep_ll 中的掩码计算，预计算 valid_mask 以减少冗余 unsqueeze 和取反操作（性能 / 风格建议）。 ispobock询问是否可直接使用 flashinfer_kernel.supports_target_verify 属性替代复杂条件，作者回复已解决。 ch-wan指出 ep_moe/layer.py 中无条件设置 bf16_dispatch 可能过于宽泛，影响其他配置；作者承认并计划参考 PR#17392 改进。ch-wan 曾建议回退以避免阻塞 PR#22822，但后来 PR#22822 已合并。 ch-wan与 YAMY1234讨论了回退此 PR 以加速 PR#22822 的可能性，作者表示可接受并计划后续重新引入。

- 优化 forward_unquantized_deepep_ll 中的掩码计算 (performance): 建议未被合并者采纳或拒绝，但代码中保留了原始实现。
- 简化 GDN verify kernel 的选择条件 (question): YAMY1234 回复 "Resolved!"，最终使用该属性作为判断依据，但保留了 decode/prefill 的 OR 条件。
- bf16_dispatch 设置过于宽泛的设计担忧 (design): PR#22822 合并后该担忧部分缓解，代码中保留了 bf16_dispatch 设置，但作者计划后续优化。
- 回退 PR 以加速 PR#22822 合并的讨论 (other): 由于 PR#22822 在讨论结束后不久合并，回退未实际执行。

风险与影响

- 风险：
  1. 性能风险：新增的 forward_unquantized_deepep_ll 是纯 Python 手写 BF16 前向，未使用 Triton 或 CUTLASS 优化，在高吞吐场景下可能成为瓶颈。
  2. 配置传播风险：bf16_dispatch 的全局设置可能无意中影响其他量化配置的路径，虽然当前 quant_config is None 条件已限制范围，但仍可能干扰未来新增的 unquantized 场景。
  3. 测试覆盖不足：本 PR 仅依赖 GPQA 端到端验证，缺少针对 forward_unquantized_deepep_ll 和 supports_target_verify 的单元测试。
  4. 兼容性风险：GDN verify 回退到 Triton 可能在某些边缘 case 下产生数值差异，但 FlashInfer 和 Triton 应该在数学上等价。 - 影响：仅影响 --speculative-algorithm NEXTN 且 --speculative-moe-runner-backend flashinfer_cutedsl 的 NVFP4 量化模型（如 Qwen3.5-397B-A17B-NVFP4）在 GB300/SM100+ 上的 MTP draft 路径。对其他模型、量化配置、GPU 架构无影响。系统吞吐经 GPQA 验证为 296.8 token/s。 - 风险标记：未量化 MoE fallback 路径，过于宽泛的 BF16 dispatch 设置，缺少测试覆盖

关联脉络

- PR #17392 相关 Dispatcher refactor: YAMY1234 在讨论中提及可以参考该 PR 中的 set_quant_config 模式来改进 bf16_dispatch 的设置方式。
- PR #22822 另一项改动，与 bf16_dispatch 冲突：ch-wan 指出 bf16_dispatch 的全局设置可能妨碍 #22822 的合并，建议回退，但最终 #22822 先合并。