PR #35758 完整报告

sgl-project/sglang

qwen 3.8 rebase

合并时间：2026-08-29 11:41

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/35758

执行摘要

- 一句话：引入 Qwen3.5 模型支持，升级 FlashInfer 0.6.18 集成
- 推荐动作：值得精读。这是理解 SGLang 如何与 FlashInfer 深度集成的重要样例，尤其是： 1) FlashInferMNNVLCuteDSLARFusion 工作区的图稳定性设计； 2) gdn_flashinfer.py 中 32 字节对齐与 Triton 回退的权衡； 3) _finish_mlp_output 的 deferred-finalize 数据契约； 4) overlap_utils.py 中一次初始化缺陷的发现与修复过程。

功能与动机

PR body 明确说明：initial pr 为 #34585，后续工作是升级 flashinfer 到 0.6.18、移除所有 flashinfer patches；并注明相比 day 0 image，本 PR 不使用 flashinfer gdn prefill cp kernel。核心动机是让 Qwen3.5（含 MoE 文本模型）在 SGLang 上完整落地，并把 FlashInfer 依赖收敛到上游 0.6.18，删除树内 patch，统一内核路径。

实现拆解

1. 新增 MNNVL CuTe DSL AllReduce 融合基础设施：新建 python/sglang/srt/layers/flashinfer_mnnvl_cutedsl.py，提供 _import_kernel_backend 懒加载后端、_with_early_finalize_shared_load 改写 FlashInfer finalize 路由（让共享专家在 PDL 前加载）、FlashInferMNNVLCuteDSLARFusion 进程级图稳定工作区，同时服务 moe_finalize_all_reduce_rms_norm 和 all_reduce_residual_rms_norm 两种融合模式。
2. 新增 Qwen3.5 专用融合服务：新建 python/sglang/srt/layers/moe/qwen35_flashinfer_fusion.py，定义 Qwen35FlashInferFusionService（工作区准备与 supports 检查）、Qwen35MoeFinalizeHandoff（承载未 finalize 的 routed 输出与独立 gated 的 shared 输出）和 resolve_max_m（取框架 token 上界作为工作区容量）。
3. 模型层接线与数据契约改造：python/sglang/srt/models/qwen3_5.py 新增 _use_mnnvl_cutedsl_fusion、_layer_communicator_class、_finish_mlp_output，通过 SGLANG_FLASHINFER_MNNVL_CUTEDSL_AR_FUSION 环境开关启用融合，并在 MLP 输出处校验 deferred-finalize handoff；qwen2_moe.py 增加 enable_qwen35_fp8_deferred_finalize、_gate_shared_output_out_of_place 和 DeepEP v2 兼容分支。
4. GDN 内核对齐与回退：gdn_flashinfer.py 新增 32 字节对齐缓冲管理（_empty_aligned_like、_prepare_dynamic_input、_prepare_parameter），并将对齐失败场景回退到 TritonGDNKernel，避免 FlashInfer 写穿可变指针导致错误。

5. MoE dispatch 与量化配套：token_dispatcher/flashinfer.py 调整 workspace 大小命名空间（按 speculative 与否加偏移）、支持 prefill allgather 分派；unquant.py 增加 FlashInfer PR 4266 BF16 GEMM 分派；logits_processor.py、radix_linear_attention.py、disaggregation/utils.py 均做了配套接线。
6. 测试与配置配套：至少 18 个测试文件与源码同步改动，覆盖融合服务、dispatch、GDN 对齐、CUDA graph 捕获等；依赖升级 FlashInfer 0.6.18 并删除原 patches，同时修复 overlap_utils.py 中 _lazy_init_forward_buf 的一次性初始化缺陷。

关键文件：

- python/sglang/srt/layers/flashinfer_mnnvl_cutedsl.py（模块 融合工作区；类别 source；类型 dependency-wiring；符号 _import_kernel_backend, _with_early_finalize_shared_load, _WorkspaceSignature, FlashInferMNNVLCuteDSLARFusion）：新增的 FlashInfer MNNVL CuTe DSL AllReduce 融合工作区管理模块，是整个 Qwen3.5 融合能力的底层基础设施。
- python/sglang/srt/layers/moe/qwen35_flashinfer_fusion.py（模块 模型融合服务；类别 source；类型 dependency-wiring；符号 is_supported_forward_mode, resolve_max_m, Qwen35MoeFinalizeHandoff, from_flashinfer）：Qwen3.5 专用的 FlashInfer 融合服务与 deferred-finalize handoff 数据契约，是模型与内核之间的桥接层。
- python/sglang/srt/models/qwen3_5.py（模块 模型入口；类别 source；类型 data-contract；符号 _use_mnnvl_cutedsl_fusion, _layer_communicator_class, _finish_mlp_output, prepare_before_cuda_graph_capture）：Qwen3.5 模型入口，定义融合开关、LayerCommunicator 选择与 MLP 输出契约检查。
- python/sglang/srt/layers/attention/linear/kernels/gdn_flashinfer.py（模块 GDN 内核；类别 source；类型 dependency-wiring；符号 _empty_aligned_like, _prepare_dynamic_input, _prepare_parameter, _mutable_inputs_are_aligned）：GDN 线性注意力内核升级，新增 32 字节对齐缓冲管理与对齐失败时的 Triton 回退。
- python/sglang/srt/models/qwen2_moe.py（模块 MoE 层；类别 source；类型 data-contract；符号 trace_sync, _forward_router_experts, supports_deferred_finalize, _gate_shared_output_out_of_place）：MoE 层支持 deferred-finalize ABI 与 DeepEP v2，是 Qwen3.5 FP8 deferred 路径的关键改造。
- python/sglang/srt/layers/moe/token_dispatcher/flashinfer.py（模块 Token 分发；类别 source；类型 dependency-wiring；符号 _max_tokens_per_scattered_source, _scattered_source_token_counts, _workspace_size_for_namespace, make_moe_a2a）：FlashInfer EP dispatch 调整 workspace 命名空间与 prefill allgather 分派，影响所有使用 flashinfer 后端的 MoE 模型。

关键符号：_import_kernel_backend, _with_early_finalize_shared_load, FlashInferMNNVLCuteDSLARFusion, Qwen35FlashInferFusionService, Qwen35MoeFinalizeHandoff.from_flashinfer, resolve_max_m, _finish_mlp_output, _layer_communicator_class, _use_mnnvl_cutedsl_fusion, _disable_shared_experts_fusion, _empty_aligned_like, _prepare_dynamic_input, _prepare_parameter, _forward_deepep, _lazy_init_forward_buf

关键源码片段

python/sglang/srt/layers/moe/qwen35_flashinfer_fusion.py

Qwen3.5 专用的 FlashInfer 融合服务与 deferred-finalize handoff 数据契约，是模型与内核之间的桥接层。

```
# qwen35_flashinfer_fusion.py —— Qwen3.5 对 FlashInfer MNNVL CuTe DSL 融合的接入

from dataclasses import dataclass

import torch

from sglang.srt.arg_groups.overrides import resolving_view
from sglang.srt.model_executor.forward_batch_info import ForwardMode


def is_supported_forward_mode(forward_mode: ForwardMode) -> bool:
    return forward_mode in (
        ForwardMode.DECODE,
        ForwardMode.EXTEND,
        ForwardMode.TARGET_VERIFY,
    )


def resolve_max_m(model_runner) -> int:
    """以框架 token 上界作为工作区容量（M）的真值来源。

    把 server_args 中的 cutedsl_moe_max_num_tokens、max_running_requests
    和 CUDA graph 配置里的各 batch size 都作为候选，取最大值；一个正数
    都没有时说明框架配置异常，直接抛错。
    """
    server_args = resolving_view(model_runner.server_args)
    decode_config = server_args.cuda_graph_config.decode
    prefill_config = server_args.cuda_graph_config.prefill
    candidates = [
        server_args.cutedsl_moe_max_num_tokens(),
        model_runner.max_running_requests,
        decode_config.max_bs,
        prefill_config.max_bs,
        *(decode_config.bs or []),
        *(prefill_config.bs or []),
    ]
    positive = [
        int(value) for value in candidates if value is not None and int(value) > 0
    ]
    if not positive:
        raise RuntimeError("framework reported no positive fusion workspace M bound")
    return max(positive)
```

```
@dataclass(frozen=True)
class Qwen35MoeFinalizeHandoff:
    """未 finalize 的 routed 输出 + 独立 gated 的 shared 贡献。"""

    routed_output: torch.Tensor
    expert_weights: torch.Tensor
    permuted_indices: torch.Tensor
    gated_shared_output: torch.Tensor
    m: int

    @classmethod
    def from_flashinfer(cls, deferred_output, *, gated_shared_output, m):
        # 从 FlashInfer 的 deferred 输出中取出 top_k、gemm2 结果和 token 置换
        # 索引，统一裁剪到当前 batch 的 m 行，交给后续 finalize 使用。
        top_k = int(deferred_output.top_k)
        return cls(
            routed_output=deferred_output.gemm2_out.view(
                -1, deferred_output.gemm2_out.shape[-1]
            ),
            expert_weights=deferred_output.expert_weights.view(-1, top_k)[:m],
            permuted_indices=deferred_output.expanded_idx_to_permuted_idx.view(
                -1, top_k
            )[:m],
            gated_shared_output=gated_shared_output,
            m=int(m),
        )
```

评论区精华

核心讨论围绕 python/sglang/srt/managers/overlap_utils.py 中 _lazy_init_forward_buf 的一次性初始化问题：

- Qiaolin-Yu 提出："it seems _lazy_init_forward_buf will only be initialized once. if the first payload has not topk_p but the following payload has, will this be an issue?"
- YAMY1234 确认这是一个真实缺陷："Good catch. The first non-empty relay can be a prefill payload without top-k, so the one-shot initialization could permanently leave those buffers disabled." 并说明修复方案：改为按字段首次出现时用 FutureMap 动态初始化，补充了回归测试，修复提交为 c31b024790。
- _lazy_init_forward_buf 一次性初始化导致后续 payload 缺少 top-k 缓冲 (correctness): YAMY1234 修改为 FutureMap 按字段首次出现时初始化，并增加回归测试覆盖无 top-k prefill 后接有 top-k decode 的场景，修复提交为 c31b024790。

风险与影响

- 风险：
  1. 依赖升级风险：FlashInfer 从带 patch 版本升级到 0.6.18 并移除全部 patches，若上游行为有细微差异（如 workspace 大小、对齐要求、prefill cp kernel 缺省），可能引

发量化或 MoE 路径回归，需重点验证 unquant.py 的 BF16 GEMM 分派和 token_dispatcher/flashinfer.py 的 workspace 分配。

1. CUDA graph 捕获风险：gdn_flashinfer.py 引入对齐缓冲缓存和 Triton 回退路径，若缓冲地址在 graph capture 后变化或回退分支触发时机不稳定，可能导致捕获失败或静默错误；qwen3_5.py 的 prepare_before_cuda_graph_capture 相关改动也需关注。
2. 一次性初始化缺陷：overlap_utils.py 的修复依赖 FutureMap 按字段首次出现初始化，但若后续出现新的 payload 类型顺序组合（如首个 payload 无 top-k、随后出现带 top-k 的 decode），仍可能存在覆盖不全的边界情况，需持续观察 PP 重叠场景。
3. 多后端兼容风险：改动涉及 CUDA、AMD/aiter、CPU 等后端分支，qwen3_5.py 中 _GDN_FUSED_QKVZBA_RATIOS 按后端分流，新增的 _gdn_decode_fused_proj_conv 仅在 CUDA 开启，其他后端行为变化需测试覆盖。 - 影响：用户影响：新增 Qwen3.5 系列模型（含 MoE 文本模型）的完整推理支持，需要使用 FlashInfer 0.6.18 及以上版本；通过环境变量 SGLANG_FLASHINFER_MNNVL_CUTEDSL_AR_FUSION 可启用 MNNVL CuTe DSL AllReduce 融合。

系统影响：FlashInfer 依赖统一升级并移除 patches，FlashInfer EP dispatch 的 workspace 命名空间调整会影响所有使用 flashinfer 后端的 MoE 模型；GDN 内核新增对齐要求与 Triton 回退，会影响线性注意力模型在 SM90/SM100 上的运行路径。

团队影响：该 PR 标记为 release-highlight，属于发布重点，后续需维护 Qwen3.5 专属融合链路与 upstream FlashInfer 的同步；修复的 overlap_utils.py 初始化问题对 PP 重叠调度的稳定性有长期价值。

- 风险标记：大型模型集成，依赖升级，CUDA graph 路径，对齐假设，一次性初始化修复

关联脉络

- PR #34585 initial qwen3.5 support (PR body 引用): PR body 明确说明本 PR 是 initial pr #34585 的 rebase 和后续工作。
- PR #36929 Update CUDA 13.4 image to flashinfer 0.6.18rc10, cutedsl 4.8. Fix sgl-wheel unpinning: 与本次升级 FlashInfer 0.6.18 并移除 patches 直接相关，同属依赖收敛线。
- PR #36914 [Fix] Lazy-import aiter in DSv4 paged_decode to unbreak CPU CI: 与本 PR 中 GDN/FlashInfer 内核的懒加载模式一致，均是为了避免非 CUDA 环境导入失败。
- PR #35453 [Fix] Support LSE on the RadixAttention extra-kwargs graph path: 本 PR 同时修改了 radix_linear_attention.py，与该修复位于同一注意力组件路径。