PR #27166 完整报告

sgl-project/sglang

Reland "Support NextN = 2/4 in DSV32"

合并时间：2026-06-06 04:43

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/27166

执行摘要

- 一句话：支持 DSV32 中 NextN = 2/4 的 DG 原生路径
- 推荐动作：值得精读。该 PR 展示了如何利用 DeepGEMM 原生多 token 接口优化计算密集型 kernel，尤其是 _build_paged_mqa_schedule_2d_ctx_lens 的布局选择逻辑和 use_dg_native 的 fallback 设计，对类似 speculative decoding 加速场景有参考价值。

功能与动机

为了利用 DeepGEMM 原生接口 [B, next_n, H, D] 的更大 MMA tile 优势，减少 atom 数量，从而降低 target_verify 延迟。PR body 提到分支已过时，需要 rebase 并确保测试通过。

实现拆解

1. 导入与数据结构扩展（dsa_backend.py）：条件导入 deep_gemm，新增 is_sm100_supported 导入；在 DSAMetadata 和 DSAIndexerMetadata 中加入 paged_mqa_ctx_lens_2d 字段，用于缓存预计算的 2D context lengths。
2. 新增 ctx_lens 构建方法（dsa_backend.py）：新增 _build_paged_mqa_schedule_2d_ctx_lens 方法，根据 forward_mode 和 speculative_num_draft_tokens 生成适当形状的 2D 张量（[B, next_n] 用于 target_verify + SM100，否则为 per-token 或其他布局）。
3. 预计算调度 metadata（dsa_backend.py）：在 init_forward_metadata 中，当处于 decode/target_verify/draft_extend 模式时，调用上述方法并赋值给 metadata.paged_mqa_ctx_lens_2d，避免后续每层重复计算。
4. 索引器路径选择（dsa_indexer.py）：在 _get_topk_paged 中，新增判断逻辑 use_dg_native，当满足 CUDA、target_verify、next_n>=2 且 paged_mqa_ctx_lens_2d 形状匹配时，走 DeepGEMM 原生 [B, next_n] 路径；否则保持原有 per-token unsqueeze 路径。
5. 原生路径适配（dsa_indexer.py）：在 use_dg_native 分支中，q_fp8 直接 view 为 [B, next_n, H, D] 而非 unsqueeze，block_tables 取 [::next_n] 去重，利用 DeepGEMM 的 stride 检查避免数据拷贝。AMD/ 非 SM100 路径保持不变。

关键文件：

- python/sglang/srt/layers/attention/dsa_backend.py（模块 DSA 后端；类别 source；类型 dependency-wiring；符号 _build_paged_mqa_schedule_2d_ctx_lens）：核心后端文件

：添加了 DeepGEMM 导入、新 metadata 字段、_build_paged_mqa_schedule_2d_ctx_lens 方法，以及在 init_forward_metadata 中预计算 2D ctx_lens。

- python/sglang/srt/layers/attention/dsa/dsa_indexer.py（模块 DSA 索引器；类别 source；类型 core-logic）：索引器核心：新增 use_dg_native 路径判断，允许在 target_verify 且 next_n>=2 时使用 DeepGEMM 原生 [B, next_n] 接口，并调整 q_fp8 和 block_tables 的传入方式。

关键符号：_build_paged_mqa_schedule_2d_ctx_lens, _get_topk_paged, init_forward_metadata

关键源码片段

python/sglang/srt/layers/attention/dsa_backend.py

核心后端文件：添加了 DeepGEMM 导入、新 metadata 字段、_build_paged_mqa_schedule_2d_ctx_lens 方法，以及在 init_forward_metadata 中预计算 2D ctx_lens。

```
def _build_paged_mqa_schedule_2d_ctx_lens(
    self,
    forward_mode: ForwardMode,
    cache_seqlens_int32: torch.Tensor,
    seqlens_expanded: torch.Tensor,
    batch_size: int,
) -> torch.Tensor:
    # target_verify 且 next_n>=2 且 SM100+ 时使用 [B, next_n] 布局
    next_n = self.speculative_num_draft_tokens
    if (
        forward_mode.is_target_verify()
        and next_n
        and next_n >= 2
        and is_sm100_supported()
    ):
        return cache_seqlens_int32.view(-1, 1).expand(-1, next_n).contiguous()
    # 其他 target_verify/draft_extend 使用 expanded 方式
    if forward_mode.is_target_verify() or forward_mode.is_draft_extend(
        include_v2=True
    ):
        return _to_2d_context_lens(seqlens_expanded, batch_size)
    # 默认 decode 使用原始 seqlens
    return _to_2d_context_lens(cache_seqlens_int32, batch_size)
```

python/sglang/srt/layers/attention/dsa/dsa_indexer.py

索引器核心：新增 use_dg_native 路径判断，允许在 target_verify 且 next_n>=2 时使用 DeepGEMM 原生 [B, next_n] 接口，并调整 q_fp8 和 block_tables 的传入方式。

```
# 在 _get_topk_paged 中，判断是否使用 DG-native 路径
B = metadata.get_seqlens_int32().shape[0]
next_n = q_offset // B if B > 0 else 0
```

```
ctx_2d = getattr(metadata, "paged_mqa_ctx_lens_2d", None)
use_dg_native = (
    _is_cuda
    and forward_batch.forward_mode.is_target_verify()
    and next_n >= 2
    and ctx_2d is not None
    and ctx_2d.shape == (B, next_n)
)

# 根据 use_dg_native 选择 2D ctx_lens
if use_dg_native:
    seqlens_32_2d = ctx_2d
elif seqlens_32.dim() == 2:
    seqlens_32_2d = seqlens_32
else:
    seqlens_32_2d = seqlens_32.unsqueeze(-1)

# ... 后续逻辑中，use_dg_native 分支直接 view q_fp8 为 [B, next_n, H, D]
elif use_dg_native:
    logits = deep_gemm.fp8_paged_mqa_logits(
        q_fp8[:q_offset].view(B, next_n, q_fp8.shape[1], q_fp8.shape[2]),
        kv_cache_fp8,
        weights[:q_offset],
        seqlens_32_2d,
        block_tables[::next_n], # 去掉重复的 page table
        schedule_metadata,
        max_seq_len,
        clean_logits=False,
    )
```

评论区精华

该 PR 的 review 评论较少，主要由作者通过 /rerun-test 命令触发 CI 验证指定测试（registered/attention/unittests/dsa/test_dsa.py），结果 15 passed, 2 skipped。审核人 Fridge003 已批准。

- 暂无高价值评论线程

风险与影响

- 风险：
  - 兼容性风险：新路径仅在 SM100+ 架构上启用，AMD 和其他 GPU 回退到原 unsqueeze 路径，无影响。
  - 正确性风险：use_dg_native 的判断条件依赖于 paged_mqa_ctx_lens_2d 的 shape 是否正确，若因元数据问题未设置或形状不匹配，将自动走原有路径（fallback），不会出错。
  - 性能风险：block_tables[::next_n] 仅为 view 操作，无额外拷贝；但若 next_n 计算有误（q_offset // B），可能导致错误的去重索引。

    - 导入风险：deep_gemm 仅为 CUDA 环境时导入，但若在非 CUDA 环境下错误配置可能导致导入失败（已用 is_cuda() 保护）。
- 影响：
    - 用户：使用 DeepSeek V3 稀疏注意力（DSV32）且处于 SM100+（Blackwell）硬件的用户，在 Target Verify 阶段（speculative decoding 中的 NextN=2/4）将获得性能提升；其他用户无变化。
    - 系统：新增约 1.4KB 代码，仅影响 DSA Attention 路径，对框架其他模块无侵入。
    - 团队：需在 SM100 上补充目标验证测试以确保新路径质量，现有 CI（1-gpu-h100，4-gpu-b200）已覆盖。
    - 风险标记：核心路径变更（target_verify），新硬件依赖（SM100），条件导入风险

关联脉络

- PR #27138 未知（PR body 引用）：PR body 中引用，可能是该功能的原始尝试。
- PR #24870 未知（PR body 引用）：PR body 中引用，可能是该功能的原始尝试。