PR #36330 完整报告

sgl-project/sglang

[AMD] Optimize Qwen3.5 MTP unified attention on gfx950

合并时间：2026-08-27 15:13

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/36330

执行摘要

- 一句话：gfx950 新增 MTP 专用注意力内核，提速约 2 倍
- 推荐动作：值得精读。该 PR 展示了如何针对特定硬件和形状手写 Triton 内核并通过形状门控安全接入，review 中关于 TP1/TP2 兼容性的讨论也体现了对通用性的思考。关注 unified_attention_3d_mtp_func 的 launch 配置和 aiter_backend.py 的门控条件，对理解 AMD 内核优化和投机解码加速有参考价值。

功能与动机

Qwen3.5 target verification on MI355X uses a short multi-token query (max_seqlen_q <= 4) with 16 query heads, one KV head, head size 256, page size 16, and FP8 KV cache. AITER's generic 3D unified-attention configuration processes one query token per workgroup and reloads the same paged KV data for each speculative token, leaving this latency-sensitive path substantially slower than the comparison implementation.

实现拆解

1. 新增 Triton 内核：在 python/sglang/kernels/ops/attention/unified_attention_3d_mtp.py 中实现 unified_attention_3d_mtp_kernel 和 unified_attention_3d_mtp_reduce_segments_kernel，并封装 unified_attention_3d_mtp_func。核心优化是 BLOCK_M=32、BLOCK_Q=2，每个 workgroup 打包两个 speculative token，复用已加载的 KV tile；TILE_SIZE=32，分段数在 8-64 之间，利用 CU 并行和 per-segment KV work。
2. 接入 dispatch：在 python/sglang/srt/layers/attention/aiter_backend.py 的 forward_extend 中新增 use_unified_attention_3d_mtp 形状门控，条件包括 gfx950、max_q_len 在 2-4、max_kv_len > 512、GQA 比例 16:1、head_dim 256、page size 16、BF16 query、FP8 KV、无 sliding window、无 softcap 等。命中时调用新内核，否则保持原 AITER 路径。
3. 注册导出：在 python/sglang/kernels/ops/attention/__init__.py 中注册 unified_attention_3d_mtp，导出 unified_attention_3d_mtp_func。
4. 新增单元测试：在 test/registered/amd/test_unified_attention_3d_mtp.py 中添加对比测试，覆盖 TP1/TP2 两种 GQA 配置（16:1 和 32:2）以及混合 query 长度 [4, 2]，验证新内核与 AITER 输出一致。

5. 性能验证：MI355X 微基准显示 1.97x-2.29x 加速，decode 阶段每层注意力耗时降低 37-38%，端到端吞吐和 TTFT 均有改善，但 concurrency 16 时 TPOT 有 5.4% 回退且原因未隔离。

关键文件：

- python/sglang/kernels/ops/attention/unified_attention_3d_mtp.py（模块 注意力内核；类别 infra；类型 infrastructure；符号 cdiv_fn, apply_softcap, find_seq_idx, unified_attention_3d_mtp_kernel）：新增 Triton 内核核心文件，实现 MTP 专用 unified attention 及 segment reduction，是性能优化的主体。
- python/sglang/srt/layers/attention/aiter_backend.py（模块 注意力后端；类别 source；类型 dependency-wiring）：在 target_verify 路径接入形状门控，决定何时使用新内核，是正确性保障的关键。
- test/registered/amd/test_unified_attention_3d_mtp.py（模块 单元测试；类别 test；类型 test-coverage；符号 TestUnifiedAttention3dMtp, test_matches_aiter, test_matches_aiter_multi_kv_head, _check_matches_aiter）：新增单元测试，对比新内核与 AITER，覆盖 TP1/TP2 配置，是确保正确性的重要配套。
- python/sglang/kernels/ops/attention/__init__.py（模块 内核导出；类别 infra；类型 infrastructure）：注册新内核导出，供其他模块导入使用。

关键符号：unified_attention_3d_mtp_kernel, unified_attention_3d_mtp_reduce_segments_kernel, unified_attention_3d_mtp_func, forward_extend

关键源码片段

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

在 target_verify 路径接入形状门控，决定何时使用新内核，是正确性保障的关键。

```
# 形状门控：只对符合条件的分发到 MTP 专用内核，
# 其他 shape 继续走 AITER 原路径，避免影响其他模型。
num_queries_per_kv = layer.tp_q_head_num // layer.tp_k_head_num
use_unified_attention_3d_mtp = (
    is_gfx95_supported() # 仅 gfx950
    and 1 < self.forward_metadata.max_q_len <= 4 # MTP 短 query
    and max_kv_len > 512 # 长上下文才值得走专用内核
    and num_queries_per_kv == 16 # 16:1 GQA，BLOCK_M=32 的前提
    and layer.tp_k_head_num == layer.tp_v_head_num # K/V 头数一致
    and layer.qk_head_dim == 256 # head_dim 仅验证过 256
    and layer.v_head_dim == 256
    and self.page_size == 16
    and q_unified.dtype == torch.bfloat16
    and k_unified.dtype == fp8_dtype
    and window_size == (-1, -1) # 不支持 sliding window
    and not layer.logit_cap # 不支持 softcap
    and sinks is None
)
if use_unified_attention_3d_mtp:
    unified_attention_3d_mtp_func(
```

```
            q=q_unified,
            k=k_unified,
            v=v_unified,
            out=o.view(-1, layer.tp_q_head_num, layer.v_head_dim),
            cu_seqlens_q=self.forward_metadata.qo_indptr,
            seqused_k=forward_batch.seq_lens + self.forward_metadata.max_q_len,
            max_seqlen_q=self.forward_metadata.max_q_len,
            max_seqlen_k=max_kv_len,
            softmax_scale=layer.scaling,
            block_table=page_table,
            k_descale=k_descale,
            v_descale=v_descale,
        )
        return o.view(-1, layer.tp_q_head_num * layer.v_head_dim)
```

test/registered/amd/test_unified_attention_3d_mtp.py

新增单元测试，对比新内核与 AITER，覆盖 TP1/TP2 配置，是确保正确性的重要配套。

```
@unittest.skipUnless(_RUNNABLE, "requires HIP gfx950 with aiter")
class TestUnifiedAttention3dMtp(CustomTestCase):
    def test_matches_aiter(self):
        # TP2 shard of Qwen3.5-397B-A17B: 32 q / 2 kv heads split over two ranks。
        self._check_matches_aiter(num_query_heads=16, num_kv_heads=1)

    def test_matches_aiter_multi_kv_head(self):
        # TP1 同一模型未分片：仍为 16:1，但有两个 kv heads。
        self._check_matches_aiter(num_query_heads=32, num_kv_heads=2)

    def _check_matches_aiter(self, num_query_heads: int, num_kv_heads: int):
        # 构造带两个序列、混合 query 长度 [4, 2] 的输入，
        # 分别用 AITER 原实现和新内核计算，再以 1e-2 容差对比。
        torch.manual_seed(0)
        device = "cuda"
        query_lens = [4, 2]
        kv_lens_list = [1024, 769]
        head_size = 256
        block_size = 16
        max_kv_len = max(kv_lens_list)
        max_blocks_per_seq = math.ceil(max_kv_len / block_size)
        num_blocks = len(query_lens) * max_blocks_per_seq

        query = torch.randn(
            sum(query_lens), num_query_heads, head_size,
            device=device, dtype=torch.bfloat16,
        )
        key = torch.randn(
            num_blocks, block_size, num_kv_heads, head_size,
            device=device, dtype=torch.bfloat16,
        ).to(e4m3_dtype)
```

```
value = torch.randn_like(key, dtype=torch.bfloat16).to(e4m3_dtype)
cu_seqlens_q = torch.tensor([0, query_lens[0], sum(query_lens)], device=device, dtype=
torch.int32)
seqused_k = torch.tensor(kv_lens_list, device=device, dtype=torch.int32)
block_table = torch.arange(num_blocks, device=device, dtype=torch.int32).view(len(query_
lens), max_blocks_per_seq)
k_descale = torch.ones(1, device=device, dtype=torch.float32)
v_descale = torch.ones(1, device=device, dtype=torch.float32)
expected = torch.empty_like(query)
actual = torch.empty_like(query)

unified_attention(
    q=query, k=key, v=value, out=expected,
    cu_seqlens_q=cu_seqlens_q, max_seqlen_q=max(query_lens),
    seqused_k=seqused_k, max_seqlen_k=max_kv_len,
    softmax_scale=head_size**-0.5, causal=True,
    window_size=(-1, -1), block_table=block_table,
    softcap=0.0, q_descale=None, k_descale=k_descale, v_descale=v_descale,
)
unified_attention_3d_mtp_func(
    q=query, k=key, v=value, out=actual,
    cu_seqlens_q=cu_seqlens_q, seqused_k=seqused_k,
    max_seqlen_q=max(query_lens), max_seqlen_k=max_kv_len,
    softmax_scale=head_size**-0.5, block_table=block_table,
    k_descale=k_descale, v_descale=v_descale,
)

torch.testing.assert_close(actual, expected, atol=1e-2, rtol=1e-2)
```

评论区精华

reviewer 1am9trash 在 aiter_backend.py 第 2312 行提出疑问：门控使用 num_queries_per_kv == 16，但 wrapper 仍断言 num_query_heads == 16 and num_kv_heads == 1，担心 TP1（32/2）配置会进入但断言失败。作者 yichiche 回应将断言改为通用形式：assert num_query_heads % num_kv_heads == 0 和 assert num_queries_per_kv == 16，并补充了 32/2 的测试。该线程已解决。

- MTP 内核门控对 TP1/TP2 的兼容性 (correctness): 已解决。提交 f62ee5d 将门控改为基于 GQA 比例，并新增 32:2 测试用例。

风险与影响

- 风险：
  1. 正确性风险：新内核门控条件复杂（形状、硬件、dtype 等），若条件设置不当可能误用路径导致数值错误。测试容差 1e-2 相对宽松，可能无法捕捉精细差异。
  2. 性能回退：concurrency 16 时 TPOT 回退 5.4%，原因未隔离，可能影响高并发场景。

3. 硬件限定：仅 gfx950 生效，其他 AMD GPU 或 NVIDIA 不受影响，但维护时需注意分支扩散。
4. 依赖 AITER：内核模块直接导入 aiter 的 e4m3_dtype 等类型，若 AITER 接口变动会导致兼容性问题。
5. 维护成本：743 行 Triton 内核与上游 AITER 有差异，未来合入上游更新时需手动同步。
- 影响：影响范围限定在 AMD gfx950 上运行的 Qwen3.5 MTP target verification 路径（TP1/TP2）。对用户而言，长上下文场景下端到端延迟和吞吐有明确提升（吞吐 +3%~+9.6%，TTFT -7.6%~-13.5%），但需注意高并发下 TPOT 可能回退。对团队而言，新增内核和测试已纳入 AMD CI（stage-b-test-1-gpu-small-amd-mi35x），风险可控。
- 风险标记：形状门控复杂，性能回退未隔离，依赖 AITER 类型，仅 gfx950 生效

关联脉络

- PR #34296 [AMD] Use fast exponentials in C4 and C128 ROCm kernels: 同为 AMD 平台 Triton/JIT 内核性能优化，展示团队在 gfx 平台上的持续调优模式。
- PR #36636 [AMD][CI] Add targeted Mori test labels: 本 PR 新增的 AMD 单元测试注册了 CI suite，与 AMD CI 基础设施演进相关。