PR #27382 完整报告

sgl-project/sglang

[AMD][Perf] Split-KV flash-decode attention for EAGLE target-verify (Triton backend)

合并时间：2026-06-19 10:11

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/27382

执行摘要

- 一句话：AMD 推测验证分块 KV 注意力，加速 ~11x
- 推荐动作：该 PR 值得精读，尤其关注：如何将成熟的 flash-decode 技术跨场景迁移、多层平台门控模式的运用、以及缓存 key 设计对显存稳定性的影响。对 TritonAttnBackend 中条件调度逻辑的封装方式值得后续类似扩展复用。

功能与动机

修复 issue #23123 揭示的 AMD 长上下文推测解码无加速的问题。PR body 指出原验证路径在 16k 上下文时 KV 带宽仅达 HBM 峰值的约 8%，严重限制推测解码收益。采用 *decode* 路径中已成熟的 flash-decode / split-KV 技术适配到验证场景，以恢复内存带宽利用率。

实现拆解

1. 新增 split-KV 验证内核：在 verify_splitkv.py 中实现两个 Triton 内核：_verify_prefix_stage1 分块处理前缀 KV 并应用 fp8 缩放，_verify_combine_stage2 通过 log-sum-exp 合并各分块的注意力结果并加入小规模 causal draft-draft 块。公共入口 verify_splitkv_fwd 接受与 extend_attention_fwd 完全相同的参数，并返回是否成功执行。
2. 调度集成：在 TritonAttnBackend.forward_extend 中，当满足 use_verify_splitkv（由 is_gfx95_supported、环境变量 SGLANG_ENABLE_SPLITKV_VERIFY 和 topk==1 共同决定）且当前模式是 is_target_verify 时，首先尝试调用 verify_splitkv_fwd；若返回 False（不支持的场景）则自动 fallback 到原有的 extend_attention_fwd。
3. 门控与配置：在 environ.py 添加 SGLANG_ENABLE_SPLITKV_VERIFY 环境变量（默认开启）以允许用户 opt-out。通过 is_gfx95_supported() 确保内核仅在有 CDNA 启动参数（waves_per_eu, matrix_instr_nonkdim）的 gfx950 上启用；NVIDIA 和其他 AMD 架构自动使用 fallback。
4. 测试与基准：新增 test_verify_splitkv.py 测试数值一致性（与 extend_attention_fwd 对比）和 can_handle 回退场景；新增基准脚本 bench_verify_splitkv.py 用于衡量不同上下文长度下的延迟加速。

关键文件：

- python/sglang/srt/layers/attention/triton_ops/verify_splitkv.py（模块 注意力内核；类别 source；类型 core-logic；符号 block_config, choose_n_splits, _verify_prefix_stage1, _verify_combine_stage2）：核心新增，包含 split-KV 验证内核的两个 Triton kernel 和门

控逻辑 can_handle。是整个 PR 的性能关键。

- python/sglang/srt/layers/attention/triton_backend.py（模块 调度层；类别 source；类型 dependency-wiring；符号 init, use_verify_splitkv, forward_extend）：修改调度入口，在 forward_extend 中添加条件判断，是启用新内核的控制点。
- python/sglang/srt/environ.py（模块 配置；类别 source；类型 configuration；符号 SGLANG_ENABLE_SPLITKV_VERIFY）：新增环境变量开关，提供 opt-out 能力，是配置入口。
- test/registered/attention/test_verify_splitkv.py（模块 测试；类别 test；类型 test-coverage；符号 _build_verify_inputs, TestVerifySplitKV, _run_parity, test_numerics_head_dim_256）：提供数值一致性和 fallback 测试，确保正确性。
- benchmark/kernels/verify_splitkv_triton/bench_verify_splitkv.py（模块 基准测试；类别 source；类型 benchmark；符号 build_inputs, main）：提供微基准测试，量化性能收益，辅助验证。

关键符号：verify_splitkv_fwd, _verify_prefix_stage1, _verify_combine_stage2, can_handle, VerifySplitKV.init, VerifySplitKV._alloc, VerifySplitKV.grow_buffers, TritonAttnBackend.init, TritonAttnBackend.forward_extend

关键源码片段

python/sglang/srt/layers/attention/triton_ops/verify_splitkv.py

核心新增，包含 split-KV 验证内核的两个 Triton kernel 和门控逻辑 can_handle。是整个 PR 的性能关键。

```
def can_handle(
    custom_mask, is_causal, mask_indptr, max_len_extend,
    sliding_window_size, sinks, logit_cap, xai_temperature_len
):
    """
    判断当前 verify 配置是否支持 split-KV 路径（位精度等价于 extend_attention_fwd）。
    只支持纯因果（topk=1）、无滑动窗口、无 sink、无 logit cap、无 xai_temperature 的场景。
    不支持 ragged extend（即所有序列的 extend 长度相同）。
    """
    # 非因果或自定义 mask 时无法处理（topk>1 时 custom_mask 非 None）
    if custom_mask is not None or not is_causal:
        return False
    # 滑动窗口、sink、logit cap、xai_temperature 等复杂场景直接 fallback
    if sliding_window_size > 0 or sinks is not None or logit_cap != 0.0 or xai_temperature_len > 0:
        return False
    # Ragged extend（不同序列 extend 长度不同）不支持
    if mask_indptr is not None:
        return False
    return True

def verify_splitkv_fwd(
    q_extend, k_extend, v_extend, o_extend,
    k_buffer, v_buffer,
```

```
    qo_indptr, kv_indptr, kv_indices,
    custom_mask, is_causal, mask_indptr,
    max_len_extend, k_scale, v_scale,
    sm_scale=None, logit_cap=0.0, skip_prefix_custom_mask=True,
    sliding_window_size=-1, sinks=None,
    window_kv_offsets=None, xai_temperature_len=-1,
    max_bs=None,
):
    """
    公共入口：先调用 can_handle 检查，若不支持则返回 False（无操作）。
    若支持则通过 _get_vk 获取或创建 VerifySplitKV 实例，执行 split-KV 的前缀分块和合并阶段，
    最后写入 o_extend。返回 True 表示已执行。
    """
    if not can_handle(custom_mask, is_causal, mask_indptr, max_len_extend,
                      sliding_window_size, sinks, logit_cap, xai_temperature_len):
        return False
    # 确定 batch size 和 n_splits
    bs = qo_indptr.shape[0] - 1
    n_splits = choose_n_splits(prefix_len=kv_indptr[-1].item(), ...)
    # 使用稳定的 max_bs 缓存实例
    if max_bs is None:
        max_bs = bs
    vk = _get_vk(max_bs, ...)
    vk.run(...) # 内含 _verify_prefix_stage1 和 _verify_combine_stage2 调用
    return True
```

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

修改调度入口，在 forward_extend 中添加条件判断，是启用新内核的控制点。

```
class TritonAttnBackend(AttentionBackend):
    def __init__(self, model_runner, ...):
        # ... 原有初始化代码 ...
        # 设置 topk 和 use_verify_splitkv 条件
        self.topk = model_runner.server_args.speculative_eagle_topk or 0
        self.use_verify_splitkv = (
            is_gfx95_supported()
            and envs.SGLANG_ENABLE_SPLITKV_VERIFY.get()
            and self.topk == 1
        )

    def forward_extend(self, forward_batch, ...):
        # ... 构建 forward 参数 ...
        # Split-KV 快速路径：仅在 target_verify 且 use_verify_splitkv 时尝试
        if (self.use_verify_splitkv
            and forward_batch.forward_mode.is_target_verify()
            and self.verify_splitkv_fwd(
                q_extend, k_extend, v_extend, o_extend,
                k_buffer, v_buffer,
                qo_indptr, kv_indptr, kv_indices,
```

```
                custom_mask, is_causal, mask_indptr,
                max_len_extend, k_descale, v_descale,
                sm_scale=sm_scale, logit_cap=logits_soft_cap,
                sliding_window_size=sliding_window_size,
                sinks=sinks, window_kv_offsets=window_kv_offsets,
                xai_temperature_len=layer.xai_temperature_len,
                max_bs=self.req_to_token_pool.size,
            )
        ):
            return # 新内核已写入 o_extend，直接返回
        # 否则 fallback 到原有 extend_attention_fwd
        # ... 原有代码 ...
```

test/registered/attention/test_verify_splitkv.py

提供数值一致性和 fallback 测试，确保正确性。

```
class TestVerifySplitKV(CustomTestCase):
    def _run_parity(self, prefix_lens, l_ext=4, h_q=16, h_kv=2, head_dim=256, dtype=torch.
    bfloat16):
        # 构造与 extend_attention_fwd 相同的输入
        q, k, v, kb, vb, qo, kvp, kvi, mle = _build_verify_inputs(prefix_lens, l_ext, h_q, h_kv, head_
        dim, head_dim, dtype, 'cuda')
        sm_scale = 1.0 / (head_dim ** 0.5)
        # 运行参考 extend_attention_fwd
        o_ref = torch.empty_like(q)
        extend_attention_fwd(q, k, v, o_ref, kb, vb, qo, kvp, kvi,
                             None, True, None, mle, 1.0, 1.0, sm_scale=sm_scale)
        # 运行 split-KV 内核
        o_split = torch.empty_like(o_ref)
        ran = verify_splitkv_fwd(q, k, v, o_split, kb, vb, qo, kvp, kvi,
                                 None, True, None, mle, 1.0, 1.0, sm_scale=sm_scale)
        # 验证内核确实执行了，并且输出与参考匹配
        self.assertTrue(ran)
        self.assert_allclose(o_split, o_ref, atol=2e-2, rtol=1e-2)
    # 测试用例遍历各种参数
    def test_numerics_head_dim_256(self):
        self._run_parity([4096] * 2, head_dim=256)
    def test_numerics_gqa_ratios(self):
        for h_q, h_kv in [(32,1), (16,2), (8,2)]:
            with self.subTest(h_q=h_q, h_kv=h_kv):
                self._run_parity([2048] * 2, h_q=h_q, h_kv=h_kv)
```

评论区精华

1. 缓存键优化：gemini-code-assist[bot] 指出原实现用动态 batch size bs 作为 _VK_CACHE 键会导致每个 batch size 分配独立 scratch buffer，引发显存膨胀。作者采纳建议，改为使用稳定的 max_bs（来自 req_to_token_pool.size）并添加 grow_buffers 方法动态扩大缓冲区。

2. 平台门控收紧：HaiShaw 要求将 use_verify_splitkv 的检查从 is_hip() 改为 is_gfx95_supported()，因为内核的块配置和 CDNA 启动提示仅针对 gfx950 调优。作者在 commit f7622c0 中先改为 is_hip()，后根据进一步 review 在 2561dc3 中改为 is_gfx95_supported()。
3. CI 测试注册：HaiShaw 要求将测试注册到 mi35x 测试组而非通用组。作者调整 register_amd_ci 的 suite 为 stage-b-test-1-gpu-small-amd-mi35x。
4. 基准脚本退出策略：HaiShaw 建议在非 gfx950 硬件上直接退出（而非打印警告后继续）。作者改为 SystemExit 并给出明确提示。

- VerifySplitKV 缓存键使用动态 batch size 导致显存膨胀 (performance): 作者采纳，重构 _get_vk 使用 max_bs 作为缓存键，并在请求超过当前大小时调用 grow_buffers。
- 平台门控从 is_hip 改为 is_gfx95_supported (correctness): 作者在 commit 2561dc3 中改为 is_gfx95_supported()，并相应调整测试注册到 mi35x 组。
- 测试注册到 mi35x 测试组 (testing): 作者将 register_amd_ci 的 suite 改为 stage-b-test-1-gpu-small-amd-mi35x。
- 基准脚本在非 gfx950 应退出而非警告 (other): 作者改为 SystemExit 并添加 docstring 说明。

风险与影响

- 风险：
  1. NVIDIA 兼容性：初始版本未门控导致 NVIDIA Triton 因 waves_per_eu 等参数崩溃，已在 commit f7622c0 和 2561dc3 中通过多层门控（is_hip → is_gfx95_supported）彻底隔离。当前 CUDA CI 已通过。
  2. 仅支持 topk=1：该内核仅适用于 EAGLE 树退化为纯因果链的情况。当 topk>1 时自动 fallback，无正确性风险，但可能难以覆盖所有推测配置。
  3. 块配置固化为 gfx950：block_config 和 choose_n_splits 当前仅针对 MI350X (gfx950) 调优。其他 AMD 架构（如 gfx942）自动禁用。未来若需支持更多架构，需要通用化或添加架构级配置。
  4. 环境变量默认开启：SGLANG_ENABLE_SPLITKV_VERIFY 默认 true，但因有多层硬件门控，在非 gfx950 上实际无影响，风险极低。 - 影响：AMD gfx950 (MI350X) 用户：推测解码验证阶段内核延迟大幅降低，端到端吞吐提升高达 30%，且不影响正确性和 accept length。其他硬件用户（NVIDIA、AMD gfx942 等）：无行为变化，自动 fallback。开发团队：需维护一个新的 Triton 内核实现及其与两个后端（CUDA/ROCm）的兼容性，增加了后续重构的复杂度。测试覆盖：单元测试验证了数值一致性，基准测试提供了性能标杆，但缺少端到端（含模型）的回归测试（依赖 nightly ROCm spec 测试）。 - 风险标记：仅支持 topk=1，平台固化为 gfx950，需维护双后端兼容，环境变量默认开启但风险低，NVIDIA 兼容已修复但需监控

关联脉络

- PR #27793 [AMD][Perf] Tune extend attention block sizes for gfx950 (head_dim > 128): 均属于 AMD gfx950 注意力性能优化系列，extend_attention 块大小调优影响本

PR 的基线性能。

- PR #28558 [AMD] register 2 spec tests to stage-b-test-1-gpu-large-amd (batch-5): 均为 AMD 推测解码测试注册，本 PR 的测试也注册到类似测试组，可参考其 CI 配置。