PR #33661 完整报告

sgl-project/sglang

[BCG][5/N] MLA Fully Support

合并时间：2026-08-10 14:52

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/33661

执行摘要

- 一句话：BCG 全量支持 MLA：移除禁用规则，修复捕获期缺陷
- 推荐动作：值得精读。核心看点：一是三个架构代理标志收敛为 is_cuda() 平台不变量的契约演进方式；二是三个捕获期缺陷（trtllm_mla 形状断言、qv 核 split-KV、dispose_tensor 释放）的根因定位逻辑；三是「拆分只是并行化策略，单 split 永远正确」的保守修复策略。建议结合 commit message 阅读，比 PR body 信息量大得多。

功能与动机

BCG 此前对 MLA 模型整体禁用：server_args._disable_breakable_cudagraph_if_incompatible 中有「MLA attention (non-DSA)」规则，理由是 MLA prefill 走 forward_mha、没有 eager break。但当 attention handler 在 BCG 下统一钉死 AttnForwardMethod.MLA 后，MHA companion 不再被捕获，该理由不再成立。PR 的目标是移除规则与 Kimi-K3 白名单，让所有 MLA 架构在 CUDA 上默认获得 BCG prefill 加速。提交信息还记录了三个必须修复的实测缺陷：trtllm_mla 的 ragged 核只接受 MHA 形状头维（192/128/256），吸收式 MLA 的 (576, 576, 512) 张量落入会直接 assert；flash_attn.cute 的 qv 核（head_dim 64、head_dim_v 512）没有 split-KV 变体；dispose_tensor 在 breakable 捕获期释放了图已记录地址的存储，重放时静默写坏 KV。

实现拆解

1. 删除 MLA-BCG 禁用规则与白名单契约（python/sglang/srt/server_args.py、python/sglang/srt/configs/model_config.py）：_disable_breakable_cudagraph_if_incompatible 移除「MLA attention (non-DSA)」规则及 is_deepseek_dsa 导入；model_config.py 删除 mla_breakable_cuda_graph_supported_model_archs（Kimi-K3）列表、is_mla_breakable_cuda_graph_supported() 函数及其在 ModelConfig.__init__ 中的属性赋值。注意 DeepSeek-V4 因捕获池内存压力仍单独禁用，不随本 PR 放开。
2. 收敛代理标志为平台判断（python/sglang/srt/model_executor/runner/prefill_cuda_graph_runner.py）：删除 dsa_sparse_prefill_forced 与 mla_pinned_under_bcg 字段及 is_deepseek_dsa 调用；can_replay_locally 中 MHA-companion 前缀禁令从「非 DSA 且非白名单且带前缀」改为「非 CUDA 且带前缀」，_restore_mha_capture_state 同样改用 not is_cuda()。语义：CUDA 上 BCG 钉死吸收式 MLA，MHA companion 从不被捕获，前缀可被图服务；非 CUDA 平台 BCG 仍走 MHA companion，前缀路径不可捕获，退回

eager。

3. 修复 BCG 捕获 / 重放期三个内核缺陷：
   - trtllm_mla_backend.py：init_mha_chunk_metadata 与 init_forward_metadata 的 flashinfer MLA 回退条件加入 is_in_breakable_cuda_graph()，避免吸收式 MLA 张量落入只接受 MHA 头维的 ragged 核。
   - python/sglang/kernels/ops/attention/flash_attn/cute/interface.py：_flash_attn_fwd 在 qv is not None 时强制 num_splits = 1。此前 diff-head-dim 守卫的「page_table is not None and q_stage == 1」例外让 paged 吸收式 MLA extend 仍走 split-KV 而断言；拆分只是并行化策略，单 split 路径永远正确。
   - python/sglang/srt/utils/common.py：dispose_tensor 增加 is_in_breakable_cuda_graph() 检查，与 tc_piecewise 一致地跳过释放。prefill runner 既不进 tc_piecewise 保护、也不进 decode 的 model_capture_mode() 保护，正是此前释放存储导致重放 KV 损坏的缺口。
4. 测试配套（test/registered/unit/configs/test_multimodal_piecewise_cuda_graph.py）：test_trtllm_mla_stays_on_breakable_and_is_disabled_by_compatibility 改名为 test_trtllm_mla_stays_on_breakable，期望从 Backend.DISABLED 改为 Backend.BREAKABLE，补充 is_multimodal=False、is_multimodal_breakable_cuda_graph_supported=False 桩字段；test_breakable_prefill_rejects_nonzero_prefix 改为 test_breakable_prefill_takes_nonzero_prefix_on_cuda_only，分别 patch is_cuda() 为 True/False 断言前缀接受 / 拒绝；_make_prefill_runner 删除 mla_pinned_under_bcg 桩。单测覆盖配置解析与 replay 资格判定，未覆盖 kernel 层。

关键文件：

- python/sglang/srt/model_executor/runner/prefill_cuda_graph_runner.py（模块 预填图执行；类别 source；类型 core-logic；符号 can_replay_locally, _restore_mha_capture_state）：BCG 重放资格判定 can_replay_locally 的核心改造点：删除 dsa_sparse_prefill_forced 与 mla_pinned_under_bcg 两个代理标志，将 MHA-companion 前缀禁令收敛为 not is_cuda()，同时 _restore_mha_capture_state 同步改用平台判断。这是整个 PR 行为语义变化最集中的文件。
- python/sglang/srt/configs/model_config.py（模块 模型配置；类别 source；类型 data-contract；符号 is_mla_breakable_cuda_graph_supported）：删除 MLA-BCG 白名单 mla_breakable_cuda_graph_supported_model_archs（Kimi-K3）、is_mla_breakable_cuda_graph_supported() 函数及 ModelConfig.__init__ 中的属性赋值，是数据契约层面的移除，影响所有读取这些符号的模块。
- python/sglang/srt/server_args.py（模块 服务参数；类别 source；类型 core-logic；符号 _disable_breakable_cudagraph_if_incompatible, _apply_cuda_graph_compatibility）：_disable_breakable_cudagraph_if_incompatible 删除「MLA attention (non-DSA)」禁用规则及 is_deepseek_dsa 导入，并更新 trtllm_mla 保持在 breakable 路径上的注释语义；这是行为默认值变化的总开关。
- python/sglang/srt/utils/common.py（模块 通用工具；类别 source；类型 bugfix；符号 dispose_tensor）：dispose_tensor 修复 breakable 捕获期的存储释放缺口：prefill

runner 既不进 tc_piecewise 保护也不进 decode 的 model_capture_mode 保护，释放图已记录地址的存储会在重放时静默写坏 KV。

- python/sglang/srt/layers/attention/trtllm_mla_backend.py（模块 注意力后端；类别 source；类型 bugfix；符号 init_mha_chunk_metadata, init_forward_metadata）：init_mha_chunk_metadata 与 init_forward_metadata 的 flashinfer MLA 回退条件加入 is_in_breakable_cuda_graph()，避免吸收式 (576, 576, 512) 张量落入只接受 MHA 头维的 ragged 核导致断言。
- python/sglang/kernels/ops/attention/flash_attn/cute/interface.py（模块 注意力内核；类别 infra；类型 bugfix；符号 _flash_attn_fwd）：_flash_attn_fwd 在 qv is not None 时强制 num_splits = 1，绕开 qv 核无 split-KV 变体的断言；修复了 diff-head-dim 守卫的「page_table is not None and q_stage == 1」例外让 paged 吸收式 MLA extend 仍走 split-KV 的漏洞。
- test/registered/unit/configs/test_multimodal_piecewise_cuda_graph.py（模块 配置测试；类别 test；类型 test-coverage；符号 test_trtllm_mla_stays_on_breakable, test_breakable_prefill_takes_nonzero_prefix_on_cuda_only）：测试语义随行为变更同步更新：trtllm_mla 期望从 DISABLED 改为 BREAKABLE；前缀测试改为按 is_cuda() 分支断言 CUDA 接受、非 CUDA 拒绝；删除 mla_pinned_under_bcg 桩字段。

关键符号：can_replay_locally, _restore_mha_capture_state, _disable_breakable_cudagraph_if_incompatible, dispose_tensor, init_forward_metadata, init_mha_chunk_metadata, _flash_attn_fwd, is_mla_breakable_cuda_graph_supported

关键源码片段

python/sglang/srt/model_executor/runner/prefill_cuda_graph_runner.py

BCG 重放资格判定 can_replay_locally 的核心改造点：删除 dsa_sparse_prefill_forced 与 mla_pinned_under_bcg 两个代理标志，将 MHA-companion 前缀禁令收敛为 not is_cuda()，同时 _restore_mha_capture_state 同步改用平台判断。这是整个 PR 行为语义变化最集中的文件。

can_replay_locally 是 BCG 重放资格的唯一事实来源，本 PR 将「调度已钉死」的三个代理（DSA 豁免、MLA 白名单、Kimi-K3 架构）收敛为 is_cuda() 平台判断：

```
def can_replay_locally(
    self,
    *,
    batch_size: int,
    num_tokens: Optional[int],
    input_embeds,
    replace_embeds,
    prefix_lens,
    is_target_verify: bool,
    capture_hidden_mode,
    return_logprob: bool,
    lora_ineligible: bool = False,
    chunked_prefix_uncapturable: bool = False,
```

```
) -> bool:
    """Rank-local replay eligibility: the single source of truth for
    ``can_run_graph`` (ForwardBatch, forward time) and the dp mlp-sync
    vote (ScheduleBatch, schedule time) — all dp ranks must reach the
    same replay-vs-eager decision or their collectives mismatch.
    """
    if self._is_full_backend and batch_size > self._capture_req_slots:
        return False
    # LoRA 重放需要 prepare_lora_batch 的静态元数据，LoRA prefill 在
    # DP attention 下保持 eager，因此这里仅凭 enable_lora 推导即可。
    if lora_ineligible:
        return False
    if input_embeds is not None:
        return False
    if replace_embeds is not None:
        return False
    # 关键收敛点：CUDA 上 BCG 对所有 MLA 架构钉死吸收式 MLA 路径
    # （attention_backend_handler 系列），MHA companion 从不被捕获，
    # 所以带前缀批次可以被图服务；非 CUDA 平台 BCG 仍走 MHA companion，
    # 其前缀路径不可捕获，必须退回 eager。
    if (
        self.prefill_backend_name == Backend.BREAKABLE
        and self.has_mha_companion_layers
        and not is_cuda()
        and prefix_lens is not None
        and any(prefix_lens)
    ):
        return False
    # FullCG 的 chunked-prefix 拓扑只覆盖有界前缀，对 breakable 投票路径无效。
    if chunked_prefix_uncapturable:
        return False
    # tc_piecewise 以 ForwardMode.EXTEND 且 spec_info=None 捕获，验证态不可重放。
    if is_target_verify:
        return False
    if (
        capture_hidden_mode is not None
        and self.capture_hidden_mode < capture_hidden_mode
    ):
        return False
    if return_logprob and not self._uses_eager_prefill_tail():
        return False
    if num_tokens is None:
        return True
    if num_tokens > self.max_num_tokens:
        return False
    # 不做形状精确匹配：load_batch 按 bucket 填充，只拒绝填充浪费过大的情况。
    padded_num_tokens = self._pad_to_bucket(num_tokens, self.capture_num_tokens)
    if padded_num_tokens > num_tokens * _MAX_PREFILL_CUDA_GRAPH_PADDING_FACTOR:
        return False
```

```
    return True
```

python/sglang/srt/utils/common.py

dispose_tensor 修复 breakable 捕获期的存储释放缺口：prefill runner 既不进 tc_piecewise 保护也不进 decode 的 model_capture_mode 保护，释放图已记录地址的存储会在重放时静默写坏 KV。

dispose_tensor 修复了 breakable 捕获期的存储释放缺口，避免重放时静默 KV 损坏：

```
def dispose_tensor(x: torch.Tensor):
    """
    Dispose a tensor by freeing its memory.
    During piecewise CUDA graph capture/replay, we skip disposal to avoid
    interfering with torch.compile's memory tracking and graph recording.
    """

    # 在捕获型 prefill 图（tc_piecewise 或 breakable）下跳过释放：
    # 释放底层存储会使图中已记录的地址失效，重放时会写坏 KV；
    # 本地导入避免循环依赖。
    from sglang.srt.model_executor.runner_backend_utils.breakable_cuda_graph import (
        is_in_breakable_cuda_graph,
    )
    from sglang.srt.model_executor.runner_backend_utils.tc_piecewise_cuda_graph import (
        is_in_tc_piecewise_cuda_graph,
    )

    if is_in_tc_piecewise_cuda_graph() or is_in_breakable_cuda_graph():
        return

    from sglang.srt.runtime_context import get_flags

    if get_flags().capture.disable_dispose_tensor:
        return

    x.set_(torch.empty((0,), device=x.device, dtype=x.dtype))
```

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

init_mha_chunk_metadata 与 init_forward_metadata 的 flashinfer MLA 回退条件加入 is_in_breakable_cuda_graph()，避免吸收式 (576, 576, 512) 张量落入只接受 MHA 头维的 ragged 核导致断言。

TRTLLMMLAAttentionBackend.init_forward_metadata 在扩展批次上决定是否回退 flashinfer MLA 实现，本 PR 把 breakable 图纳入回退条件：

```
# Eager path: no capture-stable dense write loc; the pool's _full_translate
# hook translates the write loc (safe out of a cuda graph).
self._decode_dense_loc = None
# Delegate to parent for non-decode modes.
if (
    forward_batch.forward_mode.is_extend()
```

```
        and not forward_batch.forward_mode.is_target_verify()
        and not forward_batch.forward_mode.is_draft_extend_v2()
    ):
        # extend 批次带前缀且关闭 chunked prefix cache 时，回退到
        # flashinfer MLA backend 的 ragged kernel；捕获型 prefill 图
        # （tc_piecewise 或 breakable）同样回退，因为图的 forward mode
        # 被钉在吸收式 MLA 上，而本后端的 ragged 核只接受 MHA 形状头维。
        has_prefix = any(forward_batch.extend_prefix_lens_cpu)
        fallback_to_flashinfer_impl = (
            (self.disable_chunked_prefix_cache and has_prefix)
            or is_in_tc_piecewise_cuda_graph()
            or is_in_breakable_cuda_graph()
        )
        if fallback_to_flashinfer_impl:
            super().init_forward_metadata(forward_batch)

        seq_lens = forward_batch.seq_lens - forward_batch.extend_prefix_lens
        cum_seq_lens_q = torch.cat(
            (
                torch.zeros(1, dtype=torch.int32, device=forward_batch.seq_lens.device),
                torch.cumsum(seq_lens, dim=0),
            )
        ).int()
        max_seq_len = max(forward_batch.extend_seq_lens_cpu)
        self.forward_prefill_metadata = TRTLLMMLAPrefillMetadata(
            max_seq_len,
            cum_seq_lens_q,
            seq_lens,
            fallback_to_flashinfer_impl,
        )
```

评论区精华

该 PR 没有 review 评论（review_comments_count=0），设计决策与权衡全部沉淀在 commit message 中，最值得注意的几点：

- 核心前提：所有 CUDA attention handler 在 BCG 下已钉死 AttnForwardMethod.MLA（_handle_attention_backend、handle_attention_trtllm_mla、handle_attention_triton），因此 MHA companion 从不被捕获，旧禁用规则的理由不再成立。
- 收敛设计：把三个「调度已钉死」的架构代理（dsa_sparse_prefill_forced、mla_pinned_under_bcg、Kimi-K3 白名单）折叠为一个平台判断 is_cuda()，消除了同类语义的多处重复表达。
- 权衡取舍：flash_attn split-KV 只是并行化策略，强制 num_splits = 1 永远正确，代价仅是长 KV 场景的并行度；trtllm_mla 在 BCG 下由「被禁用」改为「回退 flashinfer MLA 实现保持启用」，属于纯收益变更。
- 隐患说明：dispose_tensor 的修复点揭示了 prefill runner 既不在 tc_piecewise 保护内、也不在 decode 的 model_capture_mode() 保护内的保护缺口。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. 正确性回归风险：删除禁用规则后，所有 MLA 架构在 CUDA 上默认走 BCG，依赖「所有 CUDA attention handler 都钉死 MLA」这一前提；若未来新增 MLA 架构或新 attention backend 未覆盖该钉死逻辑，MHA companion 会被捕获且带前缀放行，重放结果可能错误。本轮单测仅覆盖 trtllm_mla + DeepseekV2 桩场景，PR body 的 Accuracy Tests 为空，缺少 E2E 精度数据。
  2. 性能回退风险：_flash_attn_fwd 对 qv is not None 强制 num_splits = 1，对所有 qv 路径生效（不只 BCG），长 KV 场景失去 split-KV 并行度。
  3. 平台判定不确定性：is_cuda() 在 AMD ROCm 下通常也返回 True，因此「非 CUDA 拒绝前缀」实际覆盖的主要是 NPU/XPU/CPU；若 ROCm 上 BCG 未钉死 MLA，AMD 会带前缀放行，需后续验证。
  4. 兼容性风险：is_mla_breakable_cuda_graph_supported 属性与函数被删除，任何外部引用（其他 runner 分支或用户脚本）会触发 AttributeError。
  5. 内存压力：BCG 覆盖范围扩大，prefill 捕获池内存占用上升（DSV4 正是因此被单独禁用，本 PR 保留了该例外）。 - 影响：用户影响：CUDA 上 DeepSeek、Kimi、MiniMax 等 MLA 模型 prefill 默认启用 BCG，无需显式 --cuda-graph-backend-prefill=breakable；同时修复了三个会导致崩溃或静默 KV 损坏的缺陷。非 CUDA 平台带前缀行为保持保守不变。系统影响：prefill 图路径覆盖面扩大，图捕获内存与 KV 池压力上升；消除了 breakable 捕获期释放存储导致的重放损坏隐患。团队影响：移除架构白名单，新 MLA 模型无需维护 allowlist；确立了「BCG 下 MLA 被钉死」的统一契约，为后续 BCG 系列扩展（如 diffusion）铺路。影响程度：中高。涉及核心 prefill 路径、配置契约与 kernel 层，但改动量可控（+42/-78，7 文件）。
- 风险标记：核心路径变更，默认行为变更，内核层断言修复，缺少 E2E 精度数据，split-KV 性能回退

关联脉络

- PR #34184 Fix stale track rows corrupting conv checkpoints under the prefill graph: 同改 prefill_cuda_graph_runner.py，同属「prefill 图重放静默损坏」高危区修复，与 dispose_tensor 释放存储问题互相印证该文件是 BCG 系列反复加固的核心。
- PR #34191 [PD] Skip speculative verify scratch on prefill servers (saves num_draft_tokens x mamba pool per rank): 同属 prefill 服务器 CUDA graph 内存与资源优化线，减少 prefill 图路径的额外 KV pool 开销，与 BCG 覆盖扩大后的内存压力形成互补。
- PR #34174 [diffusion] BCG: auto-capture the default warmup resolution instead of hard-requiring --warmup-resolutions: 同一 BCG 功能线的后续扩展，说明 BCG 正从 MLA 逐步覆盖 diffusion 等领域，本 PR 确立的「平台判断 + 钉死路径」契约是后续扩展的基础。