PR #25545 完整报告

sgl-project/sglang

[Spec] Add `trtllm_mha` support for Gemma 4 MTP draft attention backend

合并时间：2026-08-02 07:16

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25545

执行摘要

- 一句话：Gemma 4 MTP draft 后端支持 trtllm_mha，修复 topk>1 图尺寸越界
- 推荐动作：值得精读。核心价值不在改动量（4 文件 +19/-10），而在两处洞察：(1) 为什么 triton 一直掩盖 topk 展开后的序列数问题——TritonAttnBackend 从 max_num_tokens 推导一切，只有 trtllm_mha 把 max_bs 当序列数读；(2) SM100 守卫缺失导致整类测试在目标 runner 上静默跳过——测试守卫覆盖不足比没有测试更危险。建议跟进三件事：为 trtllm_mha + topk>1 补 paged tree-draft 夹具或参数化注册测试；修复 test_model_overrides.py 回归并加固属性访问；把 pyc96 提出的 server_args 层 gap 校验闭环。

功能与动机

PR body 一句话点明动机："Faster draft backend for Gemma 4 Frozen-KV MTP"。此前 Gemma 4 31B global-attn 层（headDim=512）在 num_draft_tokens >= 5 时无法使用 trtllm_mha：flashinfer issue#3343 指出 selectGqaGenerationKernel 把 numTokensHeadsQ（40~64）落到未预编译的 tile=64 KeepsMmaAb 桶，且降级启发式不会跨 Keeps/Swaps 家族边界；flashinfer#3393 修复后随 0.6.14 发布（本仓库已 pin）。作者在 B200 上重新验证 trtllm_batch_decode_with_kv_cache 在 headDim=512 下 q_len_per_req 全范围（1/2/3/4/5/6/8/16）可用，于是把更快的 trtllm_mha 引入 Gemma 4 的 MTP draft 阶段。

实现拆解

1. draft 后端选择逻辑扩展（python/sglang/srt/speculative/frozen_kv_mtp_worker_v2.py）：_init_draft_attn_backend 原先对 topk > 1 只接受 triton、否则抛 ValueError；改为 triton / trtllm_mha 双分支，并新增 _init_trtllm_mha_draft_attn_backend，以 skip_prefill=True 懒加载构造 TRTLLMHAAttnBackend。这是整个 PR 的入口变更。
2. CUDA-graph metadata 尺寸修复（python/sglang/srt/speculative/frozen_kv_mtp_cuda_graph_runner.py）：__init__ 中把 init_cuda_graph_state 的第一参数从 max_bs 改为 max_bs * topk。这是本 PR 最关键的 bugfix——没有它 trtllm_mha + topk>1 启动即崩。triton 未暴露是因为 TritonAttnBackend 的所有图 buffer 由 max_num_tokens 推导（该值已经是 max_bs * topk），其唯一读取 max_bs 的 maybe_create_verify_mask 对 draft 后端（skip_prefill=True）返回 None。

3. page_size 自动提升补全（python/sglang/srt/arg_groups/overrides.py）：_mla_backend_page_constraints 的 trtllm_mha 分支追加 view.speculative_draft_attention_backend == "trtllm_mha"，弥补此前只有 target / prefill / decode 后端触发提升、draft-only 被漏掉的缺口（trtllm_mha 要求 page_size >= 16）。
4. SM100 测试守卫修复（test/registered/attention/unittests/dense/test_trtllm_mha.py）：skip 条件加入 is_sm100_supported()，使该测试类在 B200 上真正执行，从 4 passed/5 skipped 变为 9 passed/0 skipped。
5. 已知缺口记录：PR body 明确承认 topk>1 没有单元测试（draft runner fixture 是线性 page_size=1 布局，而 trtllm_mha 要求 page_size >= 16），只靠手动 e2e 验证；并记录 target=trtllm_mha 仍需 topk == 1、SWA 与 topk>1 + page_size>1 组合正交受限。

关键文件：

- python/sglang/srt/speculative/frozen_kv_mtp_worker_v2.py（模块 投机解码；类别 source；类型 core-logic；符号 _init_draft_attn_backend, _init_trtllm_mha_draft_attn_backend）：PR 的入口变更：_init_draft_attn_backend 从仅支持 triton 扩展为 triton / trtllm_mha 双后端，新增 _init_trtllm_mha_draft_attn_backend（skip_prefill=True），使 Gemma 4 Frozen-KV MTP 的 topk>1 batch-expansion 路径首次获得 trtllm_mha 支持。
- python/sglang/srt/speculative/frozen_kv_mtp_cuda_graph_runner.py（模块 图执行器；类别 source；类型 bugfix；符号 FrozenKVMTPCudaGraphRunner.init）：最关键的 bugfix：__init__ 中 init_cuda_graph_state 传入 max_bs * topk，修复 topk>1 下 draft 图捕获阶段 cache_seqlens / page_table 越界导致的 cudaErrorIllegalAddress（700）；PR body 明确指出没有该修复 trtllm_mha + topk>1 完全无法启动。
- python/sglang/srt/arg_groups/overrides.py（模块 参数校验；类别 source；类型 configuration；符号 _mla_backend_page_constraints）：_mla_backend_page_constraints 把 speculative_draft_attention_backend == "trtllm_mha" 纳入 page_size 自动提升条件，弥补 draft-only 后端被漏掉的缺口；同时这个新增属性访问被报告破坏了 test_model_overrides.py，是本次合入的主要争议点。
- test/registered/attention/unittests/dense/test_trtllm_mha.py（模块 注意力测试；类别 test；类型 test-coverage；符号 TestTRTLLMMHADenseAttentionBackendCorrectness）：skip 守卫补上 sm100，使整类测试在 B200（4-gpu-b200 runner）上从 4 passed/5 skipped 变为 9 passed/0 skipped；同时清理了 AI 生成的冗余注释。

关键符号：_init_draft_attn_backend, _init_trtllm_mha_draft_attn_backend, _init_triton_draft_attn_backend, FrozenKVMTPCudaGraphRunner.init, _mla_backend_page_constraints

关键源码片段

python/sglang/srt/speculative/frozen_kv_mtp_worker_v2.py

PR 的入口变更：_init_draft_attn_backend 从仅支持 triton 扩展为 triton / trtllm_mha 双后端，新增 _init_trtllm_mha_draft_attn_backend（skip_prefill=True），使 Gemma 4

Frozen-KV MTP 的 topk>1 batch-expansion 路径首次获得 trtllm_mha 支持。

```
def _resolve_draft_backend_type(self) -> str:
    # 优先级：显式指定的 draft 后端 > decode 后端 > 全局 attention 后端
    return (
        self.server_args.speculative_draft_attention_backend
        or self.server_args.decode_attention_backend
        or self.server_args.attention_backend
    )


def _init_draft_attn_backend(self):
    # topk == 1 时走 chain 路径，不需要独立 draft 后端，
    # 直接复用 draft model runner 的 attention backend
    if self.topk == 1:
        return self.draft_model_runner.attn_backend

    # topk > 1 走 batch-expansion 路径：必须按用户指定的 draft
    # 后端显式构造。本 PR 新增 trtllm_mha 分支，解锁 Gemma 4
    # Frozen-KV MTP 在此路径上的 trtllm_mha 支持。
    backend_type = self._resolve_draft_backend_type()
    if backend_type == "triton":
        return self._init_triton_draft_attn_backend()
    if backend_type == "trtllm_mha":
        return self._init_trtllm_mha_draft_attn_backend()
    raise ValueError(
        "Frozen-KV MTP topk > 1 currently supports triton and trtllm_mha "
        f"attention backends, got {backend_type}."
    )


def _init_trtllm_mha_draft_attn_backend(self):
    # 懒加载 import：避免无 FlashInfer 环境下直接报错；
    # skip_prefill=True 表示该后端只面向 draft decode 阶段
    from sglang.srt.layers.attention.trtllm_mha_backend import TRTLLMHAAttnBackend

    return TRTLLMHAAttnBackend(self.draft_model_runner, skip_prefill=True)
```

python/sglang/srt/speculative/frozen_kv_mtp_cuda_graph_runner.py

最关键的 bugfix：__init__ 中 init_cuda_graph_state 传入 max_bs * topk，修复 topk>1 下 draft 图捕获阶段 cache_seqlens / page_table 越界导致的 cudaErrorIllegalAddress（700）；PR body 明确指出没有该修复 trtllm_mha + topk>1 完全无法启动。

```
# 静态捕获宽度：draft_decode 阶段每个请求的 token 数就是
# speculative_eagle_topk，因此 max_num_token 天然是展开后的总数
self.captured_req_width = resolve_num_tokens_per_req(
    phase="draft_decode", server_args=model_runner.server_args
)
self.capture_bs, _ = get_batch_sizes_to_capture(
```

```
    model_runner, self.captured_req_width
)
self.max_bs = max(self.capture_bs)
self.max_num_token = self.max_bs * self.captured_req_width

# 修复：expand_for_topk_draft 会把每个请求展开成 topk 个独立单 token
# 序列，draft attention backend 实际看到 max_bs * topk 个序列。此前传入
# 未展开的 max_bs，trtllm_mha 会据此分配 cache_seqlens(max_bs) 和
# page_table(max_bs, max_num_pages)，topk > 1 时图捕获阶段即越界，
# 报 cudaErrorIllegalAddress 错误（700）。
# triton 未暴露该问题：TritonAttnBackend 的图 buffer 全部由 max_num_tokens
# 推导（已是展开后数量），其唯一读取 max_bs 的 maybe_create_verify_mask
# 对 draft 后端（skip_prefill=True）返回 None。
self.draft_attn_backend.init_cuda_graph_state(
    self.max_bs * self.topk, self.max_num_token
)
```

评论区精华

1. pyc96 针对早期版本 _init_trtllm_mha_draft_attn_backend 的 TODO 注释提问，"do we want to add these asserts in server_args?"——已知 gap（target 侧 num_draft_tokens>=6 缺 kernel、target=triton + draft=trtllm_mha 需手动 page-size、SWA 组合受限）是否应下沉到 server_args 做显式校验。PR 最终以 body 记录 gap 收尾，该问题未闭环。
2. pyc96 再问 "Do we still need to use draft attn backend here?"，质疑 draft_forward 中显式 forward_context(attn_backend=self.draft_attn_backend) 包裹是否冗余。评论基于早期 frozen_kv_mtp_worker.py 实现，最终合入版本中相关逻辑演进至 v2，显式包裹按删除方向处理。
3. kpham-sgl 多次发起 "Remove AI gen comments" 自评并落地多个专门 commit（PR 含 Cursor / Claude Opus 5 co-author），体现 AI 辅助编码下对注释质量的纪律要求。
4. DevashishLal-CB 合入后报告 test_model_overrides.py::test_page_constraint_passes_at_callable_level 失败：_mla_backend_page_constraints 新增的 view.speculative_draft_attention_backend 属性访问在浅 view（_view(attention_backend="flashmla")）上抛错。该回归的修复不在本 PR 内。

- 是否应在 server_args 层对 trtllm_mha 已知 gap 加 assert (design): PR 以 body 记录已知 gap 收尾，作者未在 server_args 加 assert；该问题未闭环。
- draft_forward 是否还需要显式 forward_context 包裹 draft backend (design): 评论基于早期 frozen_kv_mtp_worker.py 的实现；最终合入版本中相关逻辑演进到 v2，显式 forward_context 包裹按删除方向处理。
- AI 生成注释的清理 (style): 已通过两次专用 commit 清理完毕。
- 合入后 test_model_overrides.py 回归 (correctness): 有报告记录，修复不在本 PR 内；需要在参数约束层做更健壮的属性访问或补 view 契约测试。

风险与影响

- 风险：
  1. 参数校验回归（已见报）：overrides.py 新增的 view.speculative_draft_attention_backend 属性访问隐式要求 view 对象带该属性，DevashishLal-CB 报告 test_model_overrides.py 因此失败；修复不在本 PR 内，需改为 getattr 式访问或补 view 契约测试。
  2. 组合覆盖缺口：trtllm_mha + topk>1 在 CI 中无用例，注册测试 test_frozen_kv_mtp.py 只扫 triton draft 后端，未来改动可能悄悄破坏该路径。
  3. 外部依赖：headDim=512 场景要求 flashinfer >= 0.6.14，旧 pin 会回到 "Trtllm-gen kernels not found" 的晦涩报错。
  4. 行为变化：仅指定 --speculative-draft-attention-backend trtllm_mha 也会自动把 page_size 提升到 64，改变 kv cache 页布局、显存占用与缓存行为。
  5. 测试转为执行：SM100 守卫修复后 B200 用例从静默跳过变为全量执行，属于暴露型变化，后续失败不应被误判为新回归。
- 影响：
  1. 用户影响：Gemma 4 系列（31B / E4B）用户在 Blackwell（sm100）上可通过 --speculative-draft-attention-backend trtllm_mha 获得约 23% 的 GSM8K eval 墙钟加速（17.3 s vs 22.6 s），精度持平；需 flashinfer >= 0.6.14。
  2. 系统影响：改动集中在 Frozen-KV MTP 的 draft 路径，默认 triton 路径行为不变；CUDA-graph 尺寸修复对所有走同一 runner 的后端生效，triton 不受影响（因其不按 max_bs 分配 buffer）。
  3. 团队 / 工程影响：暴露了 attention backend 对 max_bs / max_num_tokens 契约理解不一致的隐患，对后续接入新 draft backend 有直接指导意义；test_model_overrides.py 的 CI 破坏说明参数约束层的属性访问需要更健壮的写法。
  4. 测试体系影响：B200 上 trtllm_mha 相关测试从大量静默跳过变为全量执行。 - 风险标记：overrides 属性访问引发 CI 回归，trtllm_mha+topk>1 无 CI 覆盖，依赖 flashinfer >= 0.6.14，page_size 自动提升改变 kv 缓存布局

关联脉络

- PR #26521 [Spec] Fix EAGLE draft CUDA-graph capture-time NaN: test_trtllm_mha.py 注释明确引用：同一 draft-decode CUDA-graph 捕获路径上的历史修复（capture-time NaN fix）。
- PR #26655 [Spec] Fix EAGLE draft CUDA-graph replay-time slice rebind: test_trtllm_mha.py 注释明确引用：同一 init_forward_metadata_replay_cuda_graph 路径上的历史修复（replay-time slice rebind）。
- PR #31221 [AMD] Derive AITER verify tokens-per-req from input shape: 同类 speculative 路径的 attention backend 形状参数修复：修正后端对序列形状契约的假设，与本 PR 的 max_bs 契约问题同源。