PR #35002 完整报告

sgl-project/sglang

Support model-defined prefill input embedding width

合并时间：2026-08-17 06:08

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/35002

执行摘要

- 一句话：prefill CUDA graph 支持模型自定义 input_embeds 宽度
- 推荐动作：值得快速精读，重点看 getattr 默认回退与双处宽度统一的设计决策；建议合并后补一个针对 _input_embeds_hidden_size 分支的单元测试，并在模型接入文档中明确 input_embeds_hidden_size 的语义，避免未来接入方拼错属性名时静默回退。

功能与动机

PR body 明确指出：Some multimodal models merge text and media embeddings at a width that differs from config.hidden_size, so prefill CUDA graph buffers sized only from the configuration can be too narrow. 即 general_mm_embed_routine 会把合并后的文本 + 媒体 embedding 拷贝进按 model_config.hidden_size 分配的 input_embeds 缓冲区，若模型在残差流宽度上做合并，配置推算的宽度将小于实际写入宽度，导致捕获阶段缓冲区越界或过窄。

实现拆解

1. 新增模型可选契约：在 prefill_cuda_graph_runner.py 中新增 _input_embeds_hidden_size() 方法，通过 getattr(self.model_runner.model, "input_embeds_hidden_size", self.model_runner.model_config.hidden_size) 读取模型类上可选属性，未声明时回退到 config.hidden_size。
2. 输入缓冲区使用声明宽度：__init__ 的 buffers 构建段先计算 input_embeds_hidden_size，再把 PrefillInputBuffers.create 的 hidden_size 参数从 model_config.hidden_size 替换为该值，决定 input_embeds 缓冲区的实际宽度。
3. 共享注册表宽度同步：build_prefill_registry 同样接收 input_embeds_hidden_size，保证 token 轴共享缓冲注册表与输入缓冲区使用同一物理宽度，capture 与 replay 阶段的 data_ptr 语义保持一致，避免两处尺寸分叉。
4. 测试与配套：未新增测试文件，依赖既有 test_prefill_cuda_graph_runner*.py；PR body 说明作者通过直接 helper smoke check 验证了模型声明宽度与配置回退两条路径。无文档、无配置、无部署配套改动。

关键文件：

- python/sglang/srt/model_executor/runner/prefill_cuda_graph_runner.py（模块 预填充执行器；类别 source；类型 data-contract；符号 _input_embeds_hidden_size）：唯一的

变更文件，也是 prefill CUDA graph 执行的核心入口。新增 _input_embeds_hidden_size() 可选契约，并让 PrefillInputBuffers.create 与 build_prefill_registry 一致使用模型声明宽度，直接决定 input_embeds 输入缓冲区的物理尺寸。

关键符号：_input_embeds_hidden_size

关键源码片段

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

唯一的变更文件，也是 prefill CUDA graph 执行的核心入口。新增 _input_embeds_hidden_size() 可选契约，并让 PrefillInputBuffers.create 与 build_prefill_registry 一致使用模型声明宽度，直接决定 input_embeds 输入缓冲区的物理尺寸。

```
# 缓冲区构建入口（__init__ 内）：`hidden_size` 只负责多模态
# `input_embeds` 缓冲区的宽度，`general_mm_embed_routine` 会把合并后的
# 文本 + 媒体 embedding 拷贝进来。若模型在高于 `config.hidden_size` 的
# 宽度上合并（例如残差流合并），写入张量会更宽，所以允许模型声明宽度。
input_embeds_hidden_size = self._input_embeds_hidden_size()

self.buffers: PrefillInputBuffers = PrefillInputBuffers.create(
    device=self.device,
    max_bs=self.max_bs,
    max_num_tokens=self.max_num_tokens,
    cache_loc_dtype=self._cache_loc_dtype(),
    is_multimodal=self.is_multimodal,
    hidden_size=input_embeds_hidden_size, # 用模型声明宽度替换裸的 config.hidden_size
    dtype=self.model_runner.dtype,
    enable_mamba_track=self.mamba_track_enabled,
)
self.buffers.share_buffers()

# 共享缓冲区注册表使用同一宽度，保证 capture 与 replay 阶段的
# data_ptr 语义和输入缓冲区保持一致，避免两处尺寸分叉。
self.buffer_registry: CudaGraphBufferRegistry = build_prefill_registry(
    device=self.device,
    max_bs=self.max_bs,
    max_num_token=self.max_num_tokens,
    cache_loc_dtype=self._cache_loc_dtype(),
    is_multimodal=self.is_multimodal,
    hidden_size=input_embeds_hidden_size,
    embed_dtype=self.model_runner.dtype,
    enable_mamba_track=self.mamba_track_enabled,
    enable_num_token_non_padded=enable_num_token_non_padded(),
    require_gathered_buffer=require_gathered_buffer(model_runner.server_args),
    enable_prefill_cp=(is_dsa_enable_prefill_cp() or is_mla_prefill_cp_enabled()),
    source=self.buffers,
)
```

```
def _input_embeds_hidden_size(self) -> int:
    """模型在 CUDA graph 内写入的 `input_embeds` 张量宽度。

    默认回退到 `config.hidden_size`；只有合并宽度更大的多模态
    模型才需要声明 `input_embeds_hidden_size` 属性来覆盖。

    刻意用 `getattr` 加默认值而不是给所有模型类都补字段：
    该属性只存在于少数合并宽度不同的模型类上，为一次 `None`
    检查给注册表里每个模型加字段并不划算。这与 `base_runner`
    和 `decode_cuda_graph_runner` 里已有的 `hc_hidden_size`
    可选注入是同一套路。
    """
    return getattr(
        self.model_runner.model,
        "input_embeds_hidden_size",
        self.model_runner.model_config.hidden_size,
    )
```

评论区精华

本 PR 没有任何 review 评审评论，唯一一条 issue 评论是作者发出的 /tag-and-rerun-ci（触发 CI 重跑的指令）。因此不存在公开的设计争议或权衡交锋；实现决策（getattr 默认回退、双处宽度统一）均由 commit 内注释说明。

- CI 重跑指令，无实质 review 讨论 (other): 无可提炼的设计结论；相关设计权衡全部通过 commit 内注释说明。

风险与影响

- 风险：
  - 缺少直接配套单测：_input_embeds_hidden_size 的模型声明与配置回退两条分支没有测试文件直接覆盖，回归时属性名拼错会静默回退到 config.hidden_size，问题会延迟到 CUDA graph 捕获期才暴露。
  - 契约不校验实际写入宽度：该 PR 只提供模型“声明的权利”，没有运行时校验声明宽度是否真正容纳得下 input_embeds 写入；若声明值仍小于实际合并宽度，捕获阶段依然会越界。
  - 核心路径影响：改动位于 prefill_cuda_graph_runner.py 的 __init__ 捕获初始化段，属于 prefill 执行关键路径，但默认回退保证存量模型行为完全不变，风险集中在新增的宽合并模型接入场景。
- 影响：
  - 对用户 / 系统：现有模型无任何行为变化（默认值保留）；为“宽合并”类多模态模型（如残差流合并）解锁 prefill CUDA graph 路径，避免缓冲区过窄导致的捕获失败。
  - 对团队：为后续多模态模型接入提供了一个标准接入口，契约形式与解码侧已有的 hc_hidden_size 可选注入对齐，形成统一的模型自定义宽度模式。

- 影响范围：仅限 prefill runner 的缓冲区分配逻辑，不触及调度、KV cache 或模型前向逻辑。
- 风险标记：缺少直接配套单测，宽度契约不校验实际写入，影响 CUDA graph 捕获核心路径

关联脉络

- PR #34995 [VLM] Avoid synchronizing multimodal placeholder counts: 同属多模态 prefill 热路径优化，本 PR 为其打通了更宽输入缓冲区的能力，共同构成多模态 prefill 链路的性能与契约演进。
- PR #34404 [VLM] Cache Kimi-K3 per-image processor artifacts: 同一多模态 prefill 链路的前置处理与契约优化，说明团队在多模态输入进入 prefill 前的缓冲区与缓存契约上持续投入。
- PR #35000 Support unified SWA page mapping in attention metadata: 同属 prefill/decode 缓冲区与元数据契约系列调整，标志 SRT 在执行器侧持续扩展模型自定义数据契约。