PR #27988 完整报告

sgl-project/sglang

[Experimental] Full Cuda Graph Support for Prefill

合并时间：2026-07-07 09:13

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/27988

执行摘要

- 一句话：为预填充阶段引入全 CUDA 图后端
- 推荐动作：值得精读源代码，尤其是 flashinfer_backend.py 中的 workspace 计算和 prefill_cuda_graph_runner.py 中 slot 管理逻辑，展示了在变长输入下 CUDA 图捕获的经典填充策略。对于关心推理引擎极致性能的工程师，这是重要的参考实现。

功能与动机

在快速 GPU 上，小型预填充（例如几十个 token）的每次前向启动和 Python 调度开销占比过高，导致 GPU 利用率不足。通过将整个 prefill 前向捕获为 CUDA 图并复用，可以消除这些开销。PR 描述提到 'Targets small prefills on fast GPUs where per-forward launch/python overhead dominates'。

实现拆解

1. 配置扩展：在 cuda_graph_config.py 中将 full 加入预填充后端可选值，并新增 full_prefill_max_req 配置项（默认从 chunked_prefill_size 自动推导），控制每个捕获图承载的最大请求数。
2. Runner 集成：在 prefill_cuda_graph_runner.py 中导入 FullCudaGraphBackend，初始化时根据后端类型设置 _capture_req_slots 和 _full_cg_seq_lens_cpu 缓冲区；仅在 Full 后端时执行 body-only 捕获（LM 头与 logits 处理器留在图外运行）。
3. 注意力后端适配：在 flashinfer_backend.py 中添加 _full_cg_prefill_workspace_bytes（计算 split-kv 最坏情况下的 workspace 需求）和 _create_full_cg_prefill_wrappers（创建共享的 BatchPrefillWithPagedKVCacheWrapper 集合）；在 flashattention_backend.py 中新增 _init_full_cg_prefill_metadata 和 _init_full_cg_decode_metadata 方法，遵循 decode 风格的两阶段元数据契约。
4. 兼容性检查：在 server_args.py 中添加 _disable_full_prefill_cudagraph_if_incompatible 方法（当前为空规则，仅输出实验性警告），并在配置验证流程中串联调用。
5. 集成测试：新增 test/registered/cuda_graph/full_prefill/test_full_cuda_graph_prefill.py，启动 Qwen3-8B 模型并指定 --cuda-graph-backend-prefill=full 和 --attention-backend=flashinfer，验证 mgsm_en 准确率不低于 0.80。

关键文件：

- python/sglang/srt/layers/attention/flashinfer_backend.py（模块 注意力层；类别 source；类型 core-logic；符号 _full_cg_prefill_workspace_bytes, _create_full_cg_prefill_wrappers）：核心变更之一：新增 full-CG 预填充的 workspace 计算方法、wrapper 创建逻辑，以及 init_forward_metadata_out_graph 中 extend 分支的调度。
- python/sglang/srt/model_executor/runner/prefill_cuda_graph_runner.py（模块 执行器；类别 source；类型 data-contract）：Runner 主要集成点：模块文档更新、导入 FullCudaGraphBackend、初始化时处理 full 后端的 capture_req_slots 和 seq_lens_cpu 缓冲区。
- python/sglang/srt/layers/attention/flashattention_backend.py（模块 注意力层；类别 source；类型 core-logic；符号 _init_full_cg_decode_metadata, _init_full_cg_prefill_metadata）：FlashAttention 后端适配：新增 _init_full_cg_prefill_metadata 和 _init_full_cg_decode_metadata，实现 capture-stable 元数据分配和填充逻辑。
- test/registered/cuda_graph/full_prefill/test_full_cuda_graph_prefill.py（模块 测试；类别 test；类型 test-coverage；符号 TestFullCudaGraphPrefill, setUpClass, tearDownClass, test_gsm8k_accuracy）：新增集成测试，验证 full 后端在 Qwen3-8B 上的准确率，确保无回归。
- python/sglang/srt/server_args.py（模块 配置；类别 source；类型 core-logic；符号 _disable_full_prefill_cudagraph_if_incompatible）：添加 full 后端的兼容性检查方法 _disable_full_prefill_cudagraph_if_incompatible 及实验性警告。
- python/sglang/srt/model_executor/cuda_graph_config.py（模块 配置；类别 source；类型 data-contract）：将 full 加入预填充后端选项并新增 full_prefill_max_req 配置键。
- python/sglang/srt/model_executor/runner_backend/utils.py（模块 执行器；类别 source；类型 data-contract）：调整 resolve_prefill_backend 逻辑以适配 full 后端。

关键符号：_full_cg_prefill_workspace_bytes, _create_full_cg_prefill_wrappers, _init_full_cg_decode_metadata, _init_full_cg_prefill_metadata, _disable_full_prefill_cudagraph_if_incompatible, resolve_prefill_backend, PrefillCudaGraphRunner.init

关键源码片段

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

核心变更之一：新增 full-CG 预填充的 workspace 计算方法、wrapper 创建逻辑，以及 init_forward_metadata_out_graph 中 extend 分支的调度。

```
# 位于 FlashInferAttnBackend.__init__ 末尾，延迟创建 full-CG prefill wrapper 集合
self.full_cg_prefill_wrappers: Optional[
    List[BatchPrefillWithPagedKVCacheWrapper]
] = None

# 在 init_forward_metadata_out_graph 中新增 extend 分支（2024-03-20 新增）
elif forward_mode.is_extend():
```

```
    # plain EXTEND 模式：捕获时 plan() 在图外执行，重用 capture-stable wrapper；
    # 重放时捕获的内核读取刷新后的状态。必须位于 target-verify / draft-extend / dllm 分支之后。
    # split-kv 必须保持启用 —— 其 block_valid_mask 是填充 tile 提前退出的唯一机制。
    self.indices_updater_prefill.update(
        req_pool_indices[:bs],
        seq_lens[:bs],
        seq_lens_cpu[:bs] if seq_lens_cpu is not None else None,
        seq_lens_sum,
        prefix_lens=forward_batch.extend_prefix_lens[:bs],
        prefill_wrappers=self.full_cg_prefill_wrappers,
        use_ragged=False,
        encoder_lens=encoder_lens[:bs] if encoder_lens is not None else None,
        spec_info=None,
    )
```

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

Runner 主要集成点：模块文档更新、导入 FullCudaGraphBackend、初始化时处理 full 后端的 capture_req_slots 和 seq_lens_cpu 缓冲区。

```
# PrefillCudaGraphRunner.__init__ 中新增的 full 后端处理片段
self._is_full_backend = isinstance(self.backend, FullCudaGraphBackend)
if self._is_full_backend:
    max_req = model_runner.server_args.cuda_graph_config.prefill.full_prefill_max_req
    if max_req is None:
        # 自动推导：根据 chunked_prefill_size 缩放请求槽数
        max_req = max(model_runner.server_args.chunked_prefill_size // 512, 1)
    self._capture_req_slots = min(max_req, self.max_bs)
    self._full_cg_seq_lens_cpu = (
        torch.zeros((self._capture_req_slots,), dtype=torch.int64, device="cpu")
        if self._is_full_backend
        else None
    )
```

评论区精华

- 配置命名（reviewer merrymercy）：建议使用更具体的名称 full_prefill_max_req 代替初始的 req_slots，以明确其作用域。作者采纳并重命名。
- 默认请求槽数（reviewer merrymercy）：质疑默认值 64 是否过大。作者解释初始硬编码为 16，后改为从 chunked_prefill_size // 512 自动推导，并在提交中确认已通过 cuda_graph_config 暴露为可调节参数。
- 两个讨论均已解决，无未关闭的疑虑。
- 配置命名建议：req_slots → full_prefill_max_req (design): 作者接受并重命名，最终提交中已修改。
- 默认请求槽数大小质疑 (performance): 最终方案采用自动推导（max(chunked_prefill_size // 512, 1)），并暴露配置项供用户调节。

风险与影响

- 风险：
  - 实验性功能：标记为 *experimental*，在生产环境中可能不稳定，需充分测试。
  - split-kv 强制依赖：FlashInfer 后端下必须启用 split-kv（*enable_split_kv=True*），否则捕获的图无法正确退出填充 tile。若用户禁用 split-kv，图捕获将失败。
  - FlashAttention *page_size* 限制：仅支持 *page_size=1*，遇到其他 *page* 大小会抛出 ValueError 并降级为 *eager*。
  - 内存填充开销：每个捕获图固定请求槽会 pad 未使用的槽位，可能导致内存浪费和 batch 大小受限（默认自动推导值可能不优）。
  - workspace 计算偏差：_full_cg_prefill_workspace_bytes 依赖 flashinfer 内部算法对齐，若库版本升级可能导致尺寸不足或浪费。
  - 后端覆盖不全：其他注意力后端（如 trtllm_mla、DSA）未适配，会在捕获时直接报错，缺少自动降级机制。
- 影响：
  - 用户：新功能默认关闭，需显式指定 *--cuda-graph-backend-prefill=full* 启用。对小预填充（≤128 tokens）和低延迟敏感场景有明显加速（实测 1.7x）。准确率经验证无回归。
  - 系统：增加额外的 GPU workspace 分配（flashinfer 约 2GB 专用缓冲区），但仅在启用该后端时消耗。split-kv 强制开启可能对解码阶段无影响，但需注意闪存模板兼容性。
  - 团队：新增一个实验性后端，增加了配置选项和测试覆盖要求。后续需维护对齐 flashinfer/flash_attention 的元数据契约。
  - 风险标记：实验性功能，split-kv 强制依赖，fa4 *page_size=1* 限制，内存填充开销，后端覆盖不全无降级

关联脉络

- 暂无明显关联 PR