PR #25795 完整报告

sgl-project/sglang

Enable breakable CUDA graph for eagle

合并时间：2026-05-21 09:02

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25795

执行摘要

- 一句话：为 Eagle 投机解码启用可中断 CUDA 图
- 推荐动作：该 PR 为 Eagle 投机解码解锁了 BCG 能力，是一个有价值的功能增强。虽然改动量不大，但设计上的一些决策（如 capture_hidden_mode 的三种模式、草稿工作器的延迟初始化）值得学习。建议架构相关工程师精读 breakable_cuda_graph_runner.py 和 eagle_utils.py 的改动。需要注意的是，当前缺少测试覆盖，合并后应补充针对草稿工作器 BCG 的集成测试。

功能与动机

启用可中断 CUDA 图（BCG）以支持 Eagle 投机解码，让草稿工作器也能受益于 BCG 的性能提升。PR 描述中明确说明：'Enable breakable CUDA graph (BCG) support for eagle speculative decoding'。

实现拆解

实现按以下步骤进行：

1. 引入 CaptureHiddenMode：在 ForwardBatchInfo 中新增 CaptureHiddenMode 枚举（NULL/FULL/LAST），用于控制 BCG 捕获时是否保留隐藏状态以及保留哪些。
2. BreakableCudaGraphRunner 扩展：在 __init__中根据模型类型（草稿工作器或目标工作器）设置 capture_hidden_mode；为草稿工作器分配 static_draft_hidden_states 缓冲区；重新实现 replay_prepare 方法（不再从 PiecewiseCudaGraphRunner 绑定），在回放时将隐藏状态复制到缓冲区并填充 spec_info。
3. 多模态支持：检测 layer_model.forward 是否接受 input_embeds 参数，并在 _run_forward 中传递 forward_batch.input_embeds。
4. ModelRunner 改动：init_piecewise_cuda_graphs 新增 force_for_draft_worker 参数，草稿工作器在 __init__期间跳过 PCG 初始化，而是在 EagleWorkerV2 中调用 init_lm_head 后显式调用 init_piecewise_cuda_graphs(force_for_draft_worker=True)。
5. ServerArgs 简化：在预填充分离模式下，直接设置 disable_cuda_graph=True（无论 disable_piecewise_cuda_graph 设置如何），因为预填充服务器从不运行解码。
6. 工具函数：在 eagle_utils.py 中新增 get_draft_hidden_dim，用于计算草稿隐藏状态维度，支持 Eagle3 的 aux hidden state。

关键文件：

- python/sglang/srt/model_executor/breakable_cuda_graph_runner.py（模块 BCG 运行器；类别 source；类型 core-logic；符号 replay_prepare）：核心变更文件，增加了草稿工作器支持（capture_hidden_mode、replay_prepare）和多模态 input_embeds 支持。
- python/sglang/srt/speculative/eagle_utils.py（模块 投机解码工具；类别 source；类型 core-logic；符号 get_draft_hidden_dim）：新增 get_draft_hidden_dim 函数，用于计算草稿隐藏状态维度，支持 Eagle3 的 aux hidden state。
- python/sglang/srt/model_executor/model_runner.py（模块 模型执行器；类别 source；类型 data-contract；符号 init_piecewise_cuda_graphs）：修改 init_piecewise_cuda_graphs 方法签名，增加 force_for_draft_worker 参数，使得草稿工作器可以延迟初始化 PCG/BCG。
- python/sglang/srt/server_args.py（模块 配置管理；类别 source；类型 core-logic）：简化预填充分离模式下 CUDA 图禁用逻辑，直接设置 disable_cuda_graph=True。
- python/sglang/srt/speculative/eagle_worker_v2.py（模块 投机解码工作器；类别 source；类型 core-logic）：在 EagleWorkerV2 中，当启用 BCG 时，调用 init_piecewise_cuda_graphs(force_for_draft_worker=True) 以初始化草稿工作器的 BCG。

关键符号：replay_prepare, get_draft_hidden_dim, init_piecewise_cuda_graphs

关键源码片段

python/sglang/srt/model_executor/breakable_cuda_graph_runner.py

核心变更文件，增加了草稿工作器支持（capture_hidden_mode、replay_prepare）和多模态 input_embeds 支持。

```
# 在 __init__ 中根据模型类型设置隐藏状态捕获模式
self.capture_hidden_mode = CaptureHiddenMode.NULL
if model_runner.server_args.enable_return_hidden_states:
    self.capture_hidden_mode = CaptureHiddenMode.FULL
if (
    model_runner.spec_algorithm is not None
    and model_runner.spec_algorithm.is_eagle()
):
    if model_runner.is_draft_worker:
        # 草稿模型只需最后一步隐藏状态，用于下一轮预测
        self.capture_hidden_mode = CaptureHiddenMode.LAST
    else:
        self.capture_hidden_mode = CaptureHiddenMode.FULL

# ... 在 _init_buffers 中为草稿工作器分配临时缓冲区
if model_runner.is_draft_worker:
    from sglang.srt.speculative.eagle_utils import get_draft_hidden_dim

    hidden_dim = get_draft_hidden_dim(model_runner)
    self.static_draft_hidden_states = torch.zeros(
        (self.max_num_tokens, hidden_dim),
        dtype=model_runner.dtype,
        device=self.device,
```

```
    )
```

python/sglang/srt/speculative/eagle_utils.py

新增 get_draft_hidden_dim 函数，用于计算草稿隐藏状态维度，支持 Eagle3 的 aux hidden state。

```
def get_draft_hidden_dim(model_runner: ModelRunner) -> int:
    """Derive the hidden dimension of target hidden states fed to the draft model."""
    hf_config = model_runner.model_config.hf_config
    eagle_config = getattr(hf_config, "eagle_config", {})
    use_aux = eagle_config.get("use_aux_hidden_state", False)
    spec_algorithm = model_runner.spec_algorithm

    if spec_algorithm is not None and spec_algorithm.is_eagle3() and use_aux:
        # Eagle3 使用辅助隐藏状态，维度 = 各层隐藏状态之和
        base = getattr(hf_config, "target_hidden_size", None)
        if base is None:
            base = model_runner.model_config.hidden_size
        layer_ids = eagle_config.get("eagle_aux_hidden_state_layer_ids", [])
        num_aux = max(len(layer_ids), 1)
        return base * num_aux
    # 默认使用 spec_hidden_size（标准 Eagle/Eagle2）
    return model_runner.model_config.spec_hidden_size
```

python/sglang/srt/model_executor/model_runner.py

修改 init_piecewise_cuda_graphs 方法签名，增加 force_for_draft_worker 参数，使得草稿工作器可以延迟初始化 PCG/BCG。

```
def init_piecewise_cuda_graphs(self, force_for_draft_worker: bool = False):
    """Initialize piecewise CUDA graph runner."""
    self.piecewise_cuda_graph_runner = None

    if self.server_args.disable_piecewise_cuda_graph:
        logger.info(
            "Disable piecewise CUDA graph because --disable-piecewise-cuda-graph is set"
        )
        return

    # Draft models skip here during __init__; the eagle worker calls
    # this method explicitly (force_for_draft_worker=True) after
    # init_lm_head so graphs capture the final embedding weights.
    if self.is_draft_worker and not force_for_draft_worker:
        return
    # ... 后续初始化逻辑
```

评论区精华

Review 中主要讨论了以下问题：

- Assert vs ValueError：gemini-code-assist[bot] 建议将 input_embeds 参数检查从 assert 改为 ValueError，因为 assert 在优化模式下可能被跳过。作者在后续提交中修改为 raise ValueError。
- Import 位置：merrymercy 要求将 import inspect 移到文件顶部，以避免在函数内部导入。已修复。
- 参数命名：Oasis-Git 建议将 force 参数重命名为更明确的 force_for_draft_worker，作者采纳并在第三笔提交中完成。
- Prefill disaggregation CUDA 图禁用：Oasis-Git 询问为什么直接禁用 cuda graph，merrymercy 解释 disable_cuda_graph 实际控制解码 CUDA 图，预填充服务器从不运行解码，禁用可节省内存。
  - Assert 应替换为 ValueError (correctness): 作者在第二笔提交中将 assert 替换为 raise ValueError。
  - 函数内 import 应移到文件顶部 (style): 已修复，import inspect 被移到文件顶部。
  - 参数命名：force 应改为 force_for_draft_worker (design): 作者在第三笔提交中重命名为 force_for_draft_worker。
  - 为何在 prefill disaggregation 时直接禁用 cuda graph (design): Oasis-Git 表示理解，该设计合理。

风险与影响

- 风险：技术风险：
  1. 草稿工作器隐藏状态维度不匹配：get_draft_hidden_dim 推导的维度与模型实际输出不一致可能导致张量形状错误。需要确保各模型配置正确。
  2. 多模态 input_embeds 路径：新增加的 input_embeds 传递路径在模型不支持时可能触发错误（虽已添加参数检查，但非多模态模型也可能传递 None，需确保正确处理）。
  3. 缺少测试覆盖：本次修改未新增对应测试用例，草稿工作器的 BCG 捕获和回放逻辑缺乏自动化验证，回归风险较高。
  4. ServerArgs 变更影响：强制禁用预填充服务器的 cuda graph 可能影响某些依赖解码 cuda graph 的边缘场景（但理论上预填充不运行解码，风险较低）。
- 影响：影响范围：
  - 用户：使用 Eagle 投机解码的用户将自动受益于 BCG 带来的性能提升（尤其是草稿预填充阶段）。未使用 Eagle 或 BCG 的用户不受影响。
  - 系统：增加了 BCG 运行器的通用性，为后续扩展（如支持更多投机解码算法）奠定基础。预填充分离模式下节省了不必要的 cuda graph 内存分配。
  - 团队：需要维护 BCG 中的草稿相关分支，增加了测试和调试的复杂度。
  - 风险标记：缺少测试覆盖，新路径未充分验证，核心路径变更

关联脉络

- 暂无明显关联 PR