PR #33136 完整报告

sgl-project/sglang

[CP] Support breakable CUDA graphs for zigzag strategy

合并时间：2026-08-03 14:11

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/33136

执行摘要

- 一句话：zigzag CP 预填支持 breakable CUDA 图
- 推荐动作：建议精读。该 PR 展示了“在已验证路径上安全放开 CUDA graph 互斥”的完整方法论：资格判断集中化（supports_prefill_cp_bcg）、固定地址缓冲 + 捕获容量校验（PrefillCPBCGInput）、以及 dispatch 边界的统一门控（_prefill_cuda_graph_allows_context_parallel）。值得关注的设计决策包括：不引入架构白名单而是依赖配置组合判断、replay 时对 CP 元数据做几何二次校验、以及将 eager logits tail 保留在图外。对于需要扩展其他 CP 策略（如 flashinfer / DSA）BCG 支持的团队，本 PR 的模块边界与错误处理模式可以直接复用。

功能与动机

PR body 明确指出本 PR 的目标是“Enable breakable prefill CUDA graphs for the validated zigzag context-parallel path as an independent PR, then incorporate the requested review refactor”。此前 CP 预填路径与 breakable CUDA graph 互斥（server_args 中 CP 直接禁用 BCG），导致 zigzag CP 下 prefill 只能走 eager。该 PR 希望在不引入架构限制的前提下，让已经验证过的 zigzag CP + TRT-LLM MHA 组合也能享受 BCG 的预填加速，同时保持 CP 之外的路径不受影响。

实现拆解

实现按以下 5 步拆解：

1. 新建 CP-BCG 专属模块python/sglang/srt/layers/cp/bcg.py（+240 行）。将 CP 与 breakable CUDA graph 结合所需的全部逻辑收敛于此：supports_prefill_cp_bcg() 做资格判断（enable_prefill_cp、attn_cp_size == tp_size、cp_strategy == "zigzag"、prefill attention backend 为 TRT-LLM MHA，无架构白名单）；enable_cp_v2_bcg_capture() 叠加全局 enable_cp_v2() 开关；filter_prefill_cp_bcg_capture_num_tokens() 过滤掉小于 attn_cp_size * 2 的 token bucket；PrefillCPBCGInput dataclass 保存固定地址的 CP-local input_embeds / positions 缓冲与 bucket_local_tokens 字典，并提供 create() / prepare()；execute_prefill_cp_bcg() 负责 replay 后的输出切片、PP 中间层返回、CP gather 与 eager logits tail。
2. 接入 prefill CUDA graph runnerpython/sglang/srt/model_executor/runner/prefill_cuda_graph_runner.py（+52/-2）。在 __init__ 中初始化 self.enable_cp_v2_bcg_capture 与 self.prefill_cp_bcg_input，仅当 backend 为 BreakableCudaGraphBackend 且资格判断

通过时启用；启用后调用 filter_prefill_cp_bcg_capture_num_tokens 收缩捕获 token bucket 列表，并创建固定缓冲。capture_one_shape() 在 capture 阶段调用 PrefillCPBCGInput.prepare(capture=True)，load_batch() 在 replay 前调用 prepare(capture=False) 重建 CP 元数据并校验容量；execute() 中 CP-v2 路径分派到 execute_prefill_cp_bcg，而非通用 _execute_body_capture。

3. 调度边界门控python/sglang/srt/model_executor/model_runner.py（+14/-1）。新增 _prefill_cuda_graph_allows_context_parallel(prefill_runner, forward_batch)：get_cp_strategy() is None（无 CP）或（runner 启用 enable_cp_v2_bcg_capture 且 is_cp_v2_active(forward_batch)）时允许走 prefill CUDA graph；否则回退 eager，避免 CP 批次误入不支持 BCG 的图路径。原 _forward_raw 中硬编码 get_cp_strategy() is None 被替换。
4. 放开配置互斥python/sglang/srt/server_args.py（+3/-1）。_disable_breakable_cudagraph_if_incompatible() 中"context parallel"规则由无条件禁用改为 attn_cp_size > 1 and not supports_prefill_cp_bcg(self)，即满足资格的组合不再被配置阶段禁用。
5. 测试与验证配套test/registered/cp/test_gpt_oss_4gpu_mxfp4_cp.py（+2/-0）。在 GPT-OSS 120B mxfp4 4×B200 CP 测试中追加 --cuda-graph-backend-prefill breakable，使集成测试覆盖 BCG 路径；按 review 要求删除两个 standalone 单测文件（test_flashinfer_a2a_cp.py、test_prefill_cuda_graph_cp.py），改为通过现有注册测试覆盖。

实现时未触碰 decode CUDA graph、DSA prefill CP 等其他路径；所有 CP 专用逻辑均通过 sglang.srt.layers.cp.bcg 对外提供，降低了 runner 与 server_args 的耦合。

关键文件：

- python/sglang/srt/layers/cp/bcg.py（模块 CP 执行；类别 source；类型 core-logic；符号 supports_prefill_cp_bcg, enable_cp_v2_bcg_capture, filter_prefill_cp_bcg_capture_num_tokens, _slice_output_rows）：新增的核心模块，集中承载 CP 与 breakable CUDA graph 结合的全部逻辑：资格判断、bucket 过滤、固定地址 CP-local 输入缓冲、replay 容量校验与 eager tail 执行。
- python/sglang/srt/model_executor/runner/prefill_cuda_graph_runner.py（模块 图运行器；类别 source；类型 data-contract；符号 enable_cp_v2_bcg_capture, prefill_cp_bcg_input, capture_one_shape, load_batch）：prefill CUDA graph runner 的接入点：初始化 CP-BCG 状态、capture 阶段准备 CP-local 输入、replay 阶段重建元数据并分派到 execute_prefill_cp_bcg。
- python/sglang/srt/model_executor/model_runner.py（模块 执行调度；类别 source；类型 data-contract；符号 _prefill_cuda_graph_allows_context_parallel, _forward_raw）：在 prefill CUDA graph 调度边界增加 CP-v2 门控，防止不满足条件的 CP 批次进入图执行，保证回退 eager 的安全性。
- python/sglang/srt/server_args.py（模块 参数配置；类别 source；类型 configuration；符号 _disable_breakable_cudagraph_if_incompatible）：调整 BCG 兼容性规则：满足 supports_prefill_cp_bcg 的 CP 配置不再被配置阶段禁用，是功能生效的开关。

- test/registered/cp/test_gpt_oss_4gpu_mxfp4_cp.py（模块 CP 测试；类别 test；类型 test-coverage；符号 test_mxfp4_120b）：集成测试开启 breakable prefill CUDA graph，是 PR body 中 4×B200 验证的核心覆盖点。

关键符号：supports_prefill_cp_bcg, *enable_cp_v2_bcg_capture*, filter_prefill_cp_bcg_capture_num_tokens, PrefillCPBCGInput.*create*, PrefillCPBCGInput.*prepare*, *execute_prefill_cp_bcg*, _prefill_cuda_graph_allows_context_parallel

关键源码片段

python/sglang/srt/layers/cp/bcg.py

新增的核心模块，集中承载 CP 与 breakable CUDA graph 结合的全部逻辑：资格判断、bucket 过滤、固定地址 CP-local 输入缓冲、replay 容量校验与 *eager* tail 执行。

```
# python/sglang/srt/layers/cp/bcg.py
# CP 与 breakable CUDA graph 结合的核心模块。
# 所有函数以 server_args 或 runner 为显式参数，避免隐藏全局状态。


def supports_prefill_cp_bcg(server_args: ServerArgs) -> bool:
    """判断当前 prefill-CP 配置是否允许保留 BCG。"""
    # 只放开 zigzag 策略 + TRT-LLM MHA + attn_cp_size == tp_size 的组合；
    # 不绑定具体模型架构，避免像早期版本那样写死 GptOssForCausalLM。
    resolved = server_args._resolved()
    prefill_attention_backend, _ = server_args._resolved_attention_backends()
    return (
        server_args.enable_prefill_cp
        and resolved.attn_cp_size == server_args.tp_size
        and server_args.cp_strategy == "zigzag"
        and prefill_attention_backend == "trtllm_mha"
    )


def execute_prefill_cp_bcg(
    runner: PrefillCudaGraphRunner,
    forward_batch: ForwardBatch,
    static_forward_batch: ForwardBatch,
    static_num_tokens: int,
    raw_num_tokens: int,
    **kwargs,
):
    """Replay CP-local 图体，再以 eager 方式执行全局 gather 与 logits tail。"""
    cp_input = runner.prefill_cp_bcg_input
    assert cp_input is not None
    model = runner.model_runner.model
    with runner._prefill_forward_context(
        static_forward_batch,
        num_tokens=static_num_tokens,
```

```
        raw_num_tokens=raw_num_tokens,
    ):
        # 先按捕获时的 static 形状 replay，得到固定几何的本地输出。
        local_output = runner.backend.replay(
            ShapeKey(size=static_num_tokens),
            static_forward_batch,
            **kwargs,
        )
        # 图输出按 live 本地 token 数切片，避免把 padding 行当真实结果。
        local_output = _slice_output_rows(local_output, cp_input.live_local_tokens)

        capture_aux_hidden_states = getattr(model, "capture_aux_hidden_states", False)
        aux_hidden_states = None
        if capture_aux_hidden_states:
            hidden_states, aux_hidden_states = local_output
        else:
            hidden_states = local_output

        # 非 last PP rank 只需把 hidden states 传下去，不在这里做 gather。
        if not model.pp_group.is_last_rank:
            return (
                (hidden_states, aux_hidden_states)
                if capture_aux_hidden_states
                else hidden_states
            )

        # last PP rank 才做 CP gather，随后接 eager logits 处理。
        hidden_states = cp_gather_after_forward(
            hidden_states,
            static_forward_batch,
            torch.cuda.current_stream(),
        )
        return model.logits_processor(
            forward_batch.input_ids,
            hidden_states,
            model.lm_head,
            forward_batch,
            aux_hidden_states,
        )
```

评论区精华

所有 review 评论均来自作者 Fridge003（自审或代表团队），核心诉求是 收敛职责、去除冗余、缩小测试面：

- “Open a new file python/sglang/srt/layers/cp/bcg.py dedicated for Prefill CP + BCG”——将 CP 与 BCG 结合的逻辑从 prefill runner 和 server_args 中拆出，避免 runner 膨胀。结论：已新建 bcg.py 并全部迁入。

- “Move this function to python/sglang/srt/layers/cp/bcg.py You might pass server_args as input”——_supports_breakable_prefill_cp 应移到 bcg.py 并以 server_args 为参数。结论：改为 supports_prefill_cp_bcg(server_args)，并在 server_args 中调用。
- “Remove this line … GptOssForCausalLM”——不能把能力绑定到单一架构。结论：已删除架构白名单，只要配置组合满足即可。
- “Can we group these variables into a structure, maybe PrefillCPBCGInput”——把 runner 上散落的多个 CP 缓冲 / 状态字段封装为 dataclass。结论：已实现 PrefillCPBCGInput。
- “Remove this test” / “For testing this feature, we might enable breakable cuda graph in test/registered/cp/test_gpt_oss_4gpu_mxfp4_cp.py”——删除两个 standalone 单元测试，改用 4×B200 集成测试直接开启 BCG。结论：已删除并给集成测试增加参数。
- “Remove this two lines, since not is_cp_v2_active(forward_batch) has been included in self.enable_cp_v2_bcg_capture”——can_run_graph 中重复的 is_cp_v2_active 检查应移除，改由 _prefill_cuda_graph_allows_context_parallel 统一门控。结论：已按此处理。
- 另有关于移除 _validate_cp_flashinfer_dispatch_capacity 调用、恢复 a2a 断言两行、移除 streaming API 测试辅助的评论，均已在最终实现中落实。

截至合并前，17 条 review 评论全部有对应处理提交（“Address prefill CP BCG review feedback”与“Address follow-up prefill CP BCG review”），未发现未解决的反对意见。

- 新建 bcg.py 模块收敛 CP-BCG 逻辑 (design): 已新建 bcg.py，所有 CP-BCG 专用函数与 PrefillCPBCGInput 结构均迁入，runner 只保留调用。
- 移除架构白名单限制 (design): 已删除架构检查，supports_prefill_cp_bcg 只依赖 zigzag + trtllm_mha + cp size 配置。
- 将散落变量封装为 PrefillCPBCGInput (design): 已实现 PrefillCPBCGInput dataclass，包含 input_embeds、positions、bucket_local_tokens、live_local_tokens。
- 测试策略：删除 standalone 单测，改由集成测试覆盖 (testing): 两个单测文件已删除；test_gpt_oss_4gpu_mxfp4_cp.py 增加 --cuda-graph-backend-prefill breakable 参数。
- 移除 can_run_graph 中冗余 CP-v2 检查 (correctness): 已移除 runner 内重复检查，统一由 model_runner 的 _prefill_cuda_graph_allows_context_parallel 门控。
- 移除 flashinfer dispatch 容量校验与无关 a2a 改动 (refactor): 相关代码已移除，server_args 的 a2a 断言恢复原样。

风险与影响

- 风险：
  1. 组合限制风险：BCG 仅对 enable_prefill_cp + attn_cp_size == tp_size + cp_strategy == zigzag + trtllm_mha 开放，其他 CP 配置（如 flashinfer、dsa）仍被 server_args 禁用或走 eager。若用户误以为所有 CP 都支持而配置了不满足条件的组合，行为与之前一致（禁用 BCG），风险可控但需要文档说明。
  2. replay 几何 padding 风险：PrefillCPBCGInput.prepare() 在 replay 时会把 per_rank_actual_token 与 max_rank_len 强制改为捕获容量（captured_local_tokens）

，意味着小于捕获容量的 live batch 会被 padding 回固定几何执行。这保证了图安全，但 padding 会引入额外计算，极端情况下可能抵消 BCG 收益；且 live_physical_tokens > captured_local_tokens 时直接抛 RuntimeError 而非自动回退 eager，存在可用性隐患。

3. capture bucket 过滤风险：filter_prefill_cp_bcg_capture_num_tokens() 若过滤后为空会直接 ValueError 使启动失败。当用户自定义了很小的 --cuda-graph-max-prefill-tokens 且开启 CP 时可能触发，需要提示信息兜底。
4. 核心 dispatch 变更风险：model_runner.py 的 _forward_raw 门控替换了原先"CP 一律不走 prefill CUDA graph"的保守逻辑，若 is_cp_v2_active() 或 enable_cp_v2_bcg_capture 状态在不同 rank 间不一致，可能导致部分 rank 走图、部分走 eager 的集体不一致。测试仅覆盖 4×B200 一种拓扑，单卡 / 多卡混合场景缺验证。
5. 测试覆盖局限：新增覆盖集中在 GPT-OSS mxfp4 单模型、4 GPU 场景，test_cp_strategy_unit.py / test_gqa_prefill_cp.py 等周边测试虽在 CI 通过，但未覆盖新增 CP-BCG 逻辑的失败路径（如 bucket 过小、live 超容量）。 - 影响：用户影响：启用 zigzag CP + TRT-LLM MHA 的用户（典型为 GPT-OSS 等长上下文模型）在 prefill 阶段可直接获得 breakable CUDA graph 加速，无需额外配置（自动启用时自动过滤 token bucket）；其他 CP 用户行为不变。PR body 报告 4×B200 上输出吞吐约 11993 token/s，GPQA 得分 0.606 满足门槛。系统影响：server_args 的 BCG 兼容性规则发生变化，CP 不再是一刀切禁用；model_runner 的图调度增加一层条件判断，对非 CP 路径为恒真短路，开销可忽略。新增 bcg.py 模块为后续扩展其他 CP 策略的 BCG 支持提供了挂载点。团队影响：该 PR 是 #32714 大特性的拆分之一，与其他 split PR 无文件重叠，降低了合并冲突与回归面；review 推动的模块收敛使 runner 保持简洁，后续维护者理解 CP-BCG 只需看 bcg.py。

- 风险标记：核心路径变更，仅限 zigzag + TRT-LLM 组合，replay 几何强校验无回退，capture bucket 过滤后启动失败，测试覆盖单一拓扑

关联脉络

- PR #32714 （主 PR）Zigzag prefill CP + breakable CUDA graph 整体特性：本 PR body 明确声明为 Split PR 2 of #32714，与主特性共享设计意图，文件集刻意保持不重叠。
- PR #33137 [CP] Fuse zigzag attention into a single call: 同属 zigzag CP 优化线，改动 zigzag.py 与 trtllm_mha_backend.py，与本 PR 的 CP-BCG 路径在注意力后端上相互依赖，共同提升 zigzag CP 预填性能。