PR #30988 完整报告

sgl-project/sglang

[LoRA] Support LoRA under the breakable/full prefill CUDA graph

合并时间：2026-07-27 13:10

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/30988

执行摘要

- 一句话：支持 LoRA 在 breakable prefill CUDA graph 下运行，显著加速短输入
- 推荐动作：此 PR 是高质量变更，设计干净、测试充分。建议阅读核心设计模式：静态预分配 + 原位刷新，这是将 CUDA graph 与动态特性（如 LoRA）兼容的通用方法。同时，其 can_use_prefill_cuda_graph 和 prepare_lora_batch 的一致性设计值得借鉴。对于 LoRA 相关开发者，此 PR 是必读。

功能与动机

自 PR#29458 起 BCG 成为默认 prefill 后端，但启用 LoRA 时 BCG 被完全禁用，导致 LoRA 部署失去 prefill 图加速。解码图已通过原位刷新静态 batch metadata 支持 LoRA，本 PR 为 prefill 实现相同方案。PR body 明确说明：'Enabling LoRA still turns the prefill CUDA graph off entirely (BCG has been the default since #29458), so LoRA deployments lose prefill graph acceleration. The decode graph already supports LoRA via in-place refreshed static batch metadata; this PR does the same for prefill.'

实现拆解

实现按以下步骤拆解：

1. 基础层扩展 (BaseLoRABackend)：在 python/sglang/srt/lora/backend/base_backend.py 添加 supports_prefill_cuda_graph 类属性（默认 False）以及 init_prefill_cuda_graph_batch_info 抽象方法；同时新增 prefill_cuda_graph_batch_info, prefill_cuda_graph_max_bs, prefill_cuda_graph_max_tokens 三个实例属性，用于管理静态预分配元数据。
2. 后端实现：
   - triton(triton_backend.py)：supports_prefill_cuda_graph = True，实现 init_prefill_cuda_graph_batch_info，固定 PREFILL_CUDA_GRAPH_LORA_SEGMENTS = 32 个段槽位，超出此数量的请求 batch fallback 到 eager prefill。
   - csgmv(chunked_backend.py)：同样设置属性为 True，实现时根据 max_num_tokens 动态计算最大段数（上界为 ceil(N/chunk_top) + max_loras_per_batch，至少 16），不限制请求数但限制总 token 数。

- 其他后端（ascend, torch）仅设置 supports_prefill_cuda_graph = False，保持 eager 模式。

3. LoRAManager 协调(lora_manager.py)：添加 init_prefill_cuda_graph_batch_info 桥接调用。添加属性 supports_prefill_cuda_graph（组合后端支持 + 非 MoE LoRA + 非 DP attention）。添加 can_use_prefill_cuda_graph 方法，用于判断当前 batch 是否适合使用预填图（条件包括：prefill 元数据已初始化、非 DP attention、forward_mode 是 extend 且非 cuda_graph 模式、batch_size 和 extend_num_tokens 在范围内）。在 prepare_lora_batch 中新增 use_prefill_cuda_graph 分支，当不处于 decode 图且 can_use_prefill_cuda_graph 返回 True 时，原地刷新 prefill_cuda_graph_batch_info，而非创建新对象。
4. PrefillCudaGraphRunner 集成(prefill_cuda_graph_runner.py)：在初始化时检测 LoRA 支持性，若启用则调用 init_prefill_cuda_graph_batch_info 分配元数据。捕获时使用 lora_ids=[None] * bs，使 LoRA 内核在 rank 0 处 no-op。在 can_run_graph 中添加 LoRA 门控：要求 batch 的元数据已通过 can_use_prefill_cuda_graph 的原地路径准备（即与 prepare_lora_batch 使用的逻辑一致）。Full 后端 _capture_req_slots 在 LoRA 开启时与 LoRA 段槽数对齐。
5. ServerArgs 调整与测试：移除 server_args.py 中 LoRA 自动禁用 BCG 的规则，改为仅禁用 tc_piecewise（Dynamo 不兼容 LoRA）。新增 test/registered/cuda_graph/breakable/test_bcg_with_lora.py 端到端测试，对 triton 和 csgmv 后端分别启动 breakable 和 disabled 服务器，比较 prompt logprobs 确保精度一致（容差 1e-2），并断言适配器确实改变了 logprobs（>5e-2）。新增 test/registered/unit/server_args/test_server_args.py 中的 TestPrefillCudaGraphLoRACompatibility 测试，验证 --enable-lora 不再禁用 breakable，但 tc_piecewise 仍被禁用。

关键文件：

- python/sglang/srt/lora/lora_manager.py（模块 LoRA 管理器；类别 source；类型 core-logic；符号 init_prefill_cuda_graph_batch_info, supports_prefill_cuda_graph, prefill_cuda_graph_max_bs, can_use_prefill_cuda_graph）：核心协调器，新增 prefill cuda graph 的入口、状态和判断逻辑，包括 init_prefill_cuda_graph_batch_info, supports_prefill_cuda_graph, prefill_cuda_graph_max_bs, can_use_prefill_cuda_graph 以及 prepare_lora_batch 的扩展。
- test/registered/cuda_graph/breakable/test_bcg_with_lora.py（模块 BCG LoRA 测试；类别 test；类型 test-coverage；符号 _generate, _prompt_logprobs, _max_abs_diff, BCGLoRAServerMixin）：端到端精度测试，覆盖 triton 和 csgmv 后端，验证 breakable 与 disabled 的 logprobs 一致，并确保适配器确实影响输出。
- python/sglang/srt/lora/backend/chunked_backend.py（模块 CSGMV 后端；类别 source；类型 core-logic；符号 init_prefill_cuda_graph_batch_info）：csgmv 后端实现，支持 prefill CUDA graph，动态计算最大段数。
- python/sglang/srt/lora/backend/triton_backend.py（模块 Triton 后端；类别 source；类型 core-logic；符号 init_prefill_cuda_graph_batch_info）：Triton 后端实现，固定 32 个段槽，超出则 fallback eager。

- python/sglang/srt/lora/backend/base_backend.py（模块 基类；类别 source；类型 core-logic；符号 init_prefill_cuda_graph_batch_info）：基类扩展，添加 supports_prefill_cuda_graph 属性、init_prefill_cuda_graph_batch_info 抽象方法和预填图相关属性。
- python/sglang/srt/model_executor/runner/prefill_cuda_graph_runner.py（模块 Prefill 图运行器；类别 source；类型 data-contract）：PrefillCudaGraphRunner 与 LoRA 的集成，包括捕获时注入 lora_ids、运行门控、Full 后端 slot 对齐。
- test/registered/unit/server_args/test_server_args.py（模块 ServerArgs 测试；类别 test；类型 test-coverage；符号 TestPrefillCudaGraphLoRACompatibility, _handled_args, test_enable_lora_keeps_breakable_prefill_graph, test_lora_paths_keep_breakable_prefill_graph）：单元测试验证 --enable-lora 不再禁用 breakable prefill graph，但 tc_piecewise 仍被禁用。
- python/sglang/srt/server_args.py（模块 参数配置；类别 source；类型 configuration）：移除 LoRA 自动禁用 BCG 的规则，仅禁用 tc_piecewise。

关键符号：init_prefill_cuda_graph_batch_info, supports_prefill_cuda_graph, prefill_cuda_graph_max_bs, can_use_prefill_cuda_graph, prepare_lora_batch, can_run_graph, capture_one_shape, _disable_tc_piecewise_cudagraph_if_incompatible

关键源码片段

python/sglang/srt/lora/lora_manager.py

核心协调器，新增 prefill cuda graph 的入口、状态和判断逻辑，包括 init_prefill_cuda_graph_batch_info, supports_prefill_cuda_graph, prefill_cuda_graph_max_bs, can_use_prefill_cuda_graph 以及 prepare_lora_batch 的扩展。

```
# LoRAManager 中添加 prefill CUDA graph 支持

def init_prefill_cuda_graph_batch_info(self, max_num_tokens: int):
    """分配静态 prefill-CUDA-graph LoRA 元数据，按最大捕获 token 桶大小分配。在捕获前调用。"""
    self.lora_backend.init_prefill_cuda_graph_batch_info(max_num_tokens=max_num_tokens)

@property
def supports_prefill_cuda_graph(self) -> bool:
    """返回是否可以在 prefill CUDA graph 中捕获 LoRA 内核；排除 MoE LoRA 和 DP attention。"""
    return (
        self.lora_backend.supports_prefill_cuda_graph
        and not self.lora_backend.is_moe_lora
        and not self.enable_dp_attention
    )

@property
def prefill_cuda_graph_max_bs(self) -> Optional[int]:
    """prefill-graph LoRA 批次的请求数上限；在 init_prefill_cuda_graph_batch_info 运行前为 None。"""
    return self.lora_backend.prefill_cuda_graph_max_bs
```

```
def can_use_prefill_cuda_graph(self, forward_batch: ForwardBatch) -> bool:
    """判断当前 batch 是否可以使用静态 prefill-graph LoRA 元数据；同时被 prepare_lora_batch 和
    can_run_graph 调用以保持一致性。"""
    max_bs = self.lora_backend.prefill_cuda_graph_max_bs
    max_tokens = self.lora_backend.prefill_cuda_graph_max_tokens
    if max_bs is None or max_tokens is None:
        return False
    # DP attention 下各 rank 的图资格可能分歧，导致集合通信不同步
    if self.enable_dp_attention:
        return False
    # decode-CUDA-graph 的 extend 模式由 decode 静态 batch info 路径处理
    if (not forward_batch.forward_mode.is_extend() or forward_batch.forward_mode.is_cuda_
    graph()):
        return False
    if forward_batch.extend_num_tokens is None:
        return False
    return forward_batch.batch_size <= max_bs and forward_batch.extend_num_tokens <= max_
    tokens
```

评论区精华

Review 讨论主要集中在以下几点：

1. 大量注释简化的自我审查：作者 Yushengsu-thu 在几乎所有变更行的 diff 评论中要求 'remove the comments or reduce to 1 or 2 lines.'，最终提交 [88380f9] 实现了此清理。
2. 非 LoRA batch 的 gate 回归：提交 [7bd924c] 指出 'can_run_graph LoRA gate keyed off enable_lora instead of lora_ids'，导致非 LoRA batch 也错误地禁用了 prefill 图。后来修复为基于 lora_ids 判断。
3. Full backend slot clamp：针对 Full 后端，capture_req_slots 在 LoRA 开启时需要与 LoRA 段槽数对齐，否则可能越界。
4. 模式和一致性：ForwardMode 判断分歧修复，确保长 extend（非 CUDA graph mode）正确路由。
5. DP attention 防护：提交 [e937335] 添加 DP attention 下保持 eager prefill 的规则，因为 per-rank 图资格可能分歧导致集合通信不同步。
6. Oasis-Git 批准：从 CUDA graph 侧认可该 PR，等待 CI 结果。

- 非 LoRA batch 的 gate 回归 (correctness): 修复为基于 lora_ids 判断，非 LoRA batch 恢复 prefill 图加速。
- Full backend slot clamp (design): 在 PrefillCudaGraphRunner.__init__中处理对齐。
- DP attention 下禁用 prefill 图 (design): LoRA + DP attention 路由到 eager prefill。
- 注释过度冗长 (style): 注释被简化为 1-2 行，保留了必要说明。

风险与影响

- 风险：

1. 兼容性风险：此 PR 仅适用于 NVIDIA CUDA 平台。其他硬件（AMD, NPU, CPU 等）以及 tc_piecewise 后端保持 *eager* prefill，未引入新问题，但用户可能期望跨平台一致优化。
2. 内存增长：捕获预填图时，LoRA 静态元数据会额外占用约 0.38 GB（38 个桶）或更多。在捕获 8192 大桶时额外 1.5 GB。对显存紧张的环境可能造成 OOM。
3. triton 固定 32 段限制：PREFILL_CUDA_GRAPH_LORA_SEGMENTS = 32 意味着批量请求数超过 32 时会 fallback 到 *eager* prefill，长 batch 场景性能可能回退。对此有明确注释且是设计取舍。
4. lm_head LoRA 在 *eager* tail：lm_head LoRA 仍然运行在图外，可能成为长序列下的性能瓶颈。
5. 一致性风险：*can_use_prefill_cuda_graph* 和 prepare_lora_batch 使用相同的逻辑判断，但若未来修改不同步，可能导致图准备与运行状态不一致，出现静默错误。
6. FP 精度：使用 *enable-deterministic-inference* 时 batch 组合方式变化可能影响 logprobs，测试容差 1e-2 提供裕度。 - 影响：用户影响：所有使用 LoRA 且 prefill 后端为 breakable 或 full 的生产部署将自动获得短输入加速（128 tokens 时 TTFT 降低约 50%）。长输入性能持平。后端贡献者：需要为自定义 LoRA 后端实现 init_prefill_cuda_graph_batch_info 并设置 supports_prefill_cuda_graph = True 以启用此优化。团队影响：需要维护两个独立的静态 batch_info（*decode* 和 prefill），增加了状态复杂性，但复用相同的 *can_use_prefill_cuda_graph* 门控逻辑降低了维护成本。

- 风险标记：CUDA only，内存增长（捕获阶段），triton 32 段回退，lm_head LoRA 在图外，DP attention 禁用

关联脉络

- PR #25779 [LoRA] Support LoRA under the breakable/full prefill CUDA graph (superseded): 本 PR 取代了 #25779 的早期尝试。
- PR #25783 [LoRA] Support LoRA under the breakable/full prefill CUDA graph (superseded): 本 PR 取代了 #25783 的早期尝试。
- PR #25788 [LoRA] Support LoRA under the breakable/full prefill CUDA graph (superseded): 本 PR 取代了 #25788 的早期尝试。
- PR #29458 Make BCG the default prefill CUDA graph backend: 使 BCG 成为默认 prefill 后端，从而暴露了 LoRA 禁用 BCG 的问题，成为本 PR 的动机之一。