PR #27407 完整报告

sgl-project/sglang

Route the eager forward path through the CUDA graph input-buffer registry

合并时间：2026-06-07 05:35

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/27407

执行摘要

- 一句话：*eager* 前向路径通过 CUDA graph 缓冲注册表路由
- 推荐动作：值得精读，尤其是 *extract_buffer* 的 None 携带逻辑和 *computed slot* 暴露策略，以及如何通过参数化构建函数支持不同前向路径。提供了清晰的注释和单元测试。建议部署者评估 *decode* 延迟并在需要时启用 SGLANG_EAGER_INPUT_NO_COPY。

功能与动机

CUDA graph *capture* 和 *replay* 已经使用 CudaGraphBufferRegistry 组装前向输入张量，但 *eager* 前向路径（forward_decode/forward_extend/forward_idle）仍然直接从 ForwardBatch 读取，导致两套并行的输入构建方式，且 *registry* 的 *extract_buffer* 接口没有实际调用者。此 PR 将 *eager* 路径纳入同一注册表，统一输入构建路径。

实现拆解

1. 修改 CudaGraphBufferRegistry.extract_buffer：当 slot 为普通拷贝且 ForwardBatch 对应字段为 None 时（如非多模态批次的 mrope_positions），不再暴露 *stale/zero* buffer 而是直接从模板携带 None；*computed* slots 始终暴露。同时向 build_decode_registry 和 build_prefill_registry 添加 register_global_num_tokens 和 register_input_embeds 标志，使 *eager* 路径可以跳过某些 *computed* slots 以避免覆盖 Batch 上已有的正确值（如 DP attention 下的 global_num_tokens）。
2. 在 ModelRunner 中添加 _EagerBufferRegistry 容器和 _ensure_eager_registry 辅助方法，按需惰性构建并增长（next_power_of_2）适合 *eager* batch 大小的注册表。在 forward_decode/forward_extend/forward_idle 中，使用注册表构建 ForwardBatch 视图，注册表填充后通过 *extract_buffer* 获取视图，未缓冲字段（input_embeds, seq_lens, spec_info 等）直接从 Batch 携带。
3. 提供 SGLANG_EAGER_INPUT_NO_COPY 环境变量作为逃生口：当设置时，*eager* 路径跳过注册表拷贝，直接使用 *dataclasses.replace* 包装 Batch 自身张量，避免 *per-iter* 的 *device-to-device* 拷贝对延迟敏感解码路径的性能影响。
4. 更新配套测试：新增 *extract_buffer* 对缺席字段的处理、*computed* slots 暴露行为、register_global_num_tokens=False 和 register_input_embeds=False 的单元测试；以及验证注册表支持的视图在 TBO filter_batch 操作中与原始 Batch 行为一致的集成测试。

关键文件：

- python/sglang/srt/model_executor/model_runner.py（模块 模型执行器；类别 source；类型 core-logic；符号 _EagerBufferRegistry, _ensure_eager_registry, _ensure_eager_decode_registry, _ensure_eager_prefill_registry）：核心变更文件，添加 _EagerBufferRegistry 和 _ensure_eager_registry 等辅助，修改 forward_decode/forward_extend/forward_idle 通过注册表构建 ForwardBatch 视图
- python/sglang/srt/model_executor/cuda_graph_buffer_registry.py（模块 缓冲注册表；类别 source；类型 data-contract；符号 extract_buffer, build_decode_registry, build_prefill_registry, _global_num_tokens_post_fill）：修改 extract_buffer 以携带缺席字段，为 build_decode_registry 和 build_prefill_registry 添加新参数
- test/registered/unit/model_executor/test_cuda_graph_buffer_registry.py（模块 注册表测试；类别 test；类型 test-coverage；符号 test_extract_carries_none_for_absent_plain_slot, test_extract_exposes_computed_slot_even_when_fb_field_none, test_register_global_num_tokens_false_carries_fb_values, test_register_input_embeds_false_keeps_mrope_carries_embeds）：新增 extract_buffer 行为单元测试，覆盖 None 携带、computed slot 暴露、参数标志等
- test/registered/unit/batch_overlap/test_tbo_filter_batch_marker.py（模块 批次过滤测试；类别 test；类型 test-coverage；符号 _make_valued_batch, TestTboFilterBatchOnRegistryView, _registry_view, test_filter_registry_view_matches_raw_batch）：新增 TBO filter_batch 对注册表视图的兼容性测试
- python/sglang/srt/environ.py（模块 环境配置；类别 source；类型 configuration；符号 SGLANG_EAGER_INPUT_NO_COPY）：添加 SGLANG_EAGER_INPUT_NO_COPY 环境变量

关键符号：_EagerBufferRegistry, _ensure_eager_registry, _ensure_eager_decode_registry, _ensure_eager_prefill_registry, _eager_fb_view, extract_buffer, build_decode_registry, build_prefill_registry, _global_num_tokens_post_fill

关键源码片段

python/sglang/srt/model_executor/model_runner.py

核心变更文件，添加 _EagerBufferRegistry 和 _ensure_eager_registry 等辅助，修改 forward_decode/forward_extend/forward_idle 通过注册表构建 ForwardBatch 视图

```
# python/sglang/srt/model_executor/model_runner.py

@dataclass
class _EagerBufferRegistry:
    # Holds a lazily-built CudaGraphBufferRegistry and its current capacity.
    # TODO: Replace with a more formal capacity management strategy if needed.
    registry: Optional["CudaGraphBufferRegistry"] = None
    max_bs: int = 0
    max_num_tokens: int = 0

class ModelRunner(ModelRunnerKVCacheMixin):
```

```
def __init__(self, ...):
    # ...
    # Each runner keeps two registries: one for decode/idle (axis=bs) and
    # one for prefill/extend (axis=tokens), both grown on demand.
    self._eager_decode_registry = _EagerBufferRegistry()
    self._eager_prefill_registry = _EagerBufferRegistry()

def _ensure_eager_registry(
    self,
    cache: _EagerBufferRegistry,
    raw_bs: int,
    raw_num_tokens: int,
    build: Callable[[int, int], "CudaGraphBufferRegistry"],
) -> "CudaGraphBufferRegistry":
    # Built on first use and grown (next power of two) when a batch exceeds
    # the current capacity.
    if (
        cache.registry is not None
        and raw_bs <= cache.max_bs
        and raw_num_tokens <= cache.max_num_tokens
    ):
        return cache.registry
    cache.max_bs = next_power_of_2(max(raw_bs, cache.max_bs))
    cache.max_num_tokens = next_power_of_2(
        max(raw_num_tokens, cache.max_num_tokens)
    )
    cache.registry = build(cache.max_bs, cache.max_num_tokens)
    return cache.registry

def _ensure_eager_decode_registry(
    self, raw_bs: int, raw_num_tokens: int
) -> "CudaGraphBufferRegistry":
    # Builds a decode registry with `register_global_num_tokens=False` to
    # prevent overwriting DP attention values already on the batch.
    seq_len_fill = self.dummy_seq_len_fill_value
    return self._ensure_eager_registry(
        self._eager_decode_registry,
        raw_bs,
        raw_num_tokens,
        lambda bs, num_tokens: build_decode_registry(
            device=self.device,
            max_bs=bs,
            max_num_tokens=num_tokens,
            seq_len_fill_value=seq_len_fill,
            cache_loc_dtype=self.cache_loc_dtype,
            require_gathered_buffer=False,
            require_mlp_tp_gather=self.server_args.enable_mlp_tp_communication,
            dp_size=1,
            register_global_num_tokens=False,
```

```
            ),
        )
```

python/sglang/srt/model_executor/cuda_graph_buffer_registry.py

修改 extract_buffer 以携带缺席字段，为 build_decode_registry 和 build_prefill_registry 添加新参数

```
# python/sglang/srt/model_executor/cuda_graph_buffer_registry.py

def extract_buffer(
    self,
    *,
    padded_bs: int,
    padded_num_tokens: int,
    forward_batch_template: "ForwardBatch",
) -> "ForwardBatch":
    # Return a FB view (dataclasses.replace of forward_batch_template)
    # whose slot fields are buffer views and whose non-slot fields are carried
    # from the template. A plain copy slot whose FB field is None this iter
    # is carried (not exposed as a stale buffer); computed slots are always
    # exposed.
    import dataclasses

    replace_kwargs: Dict[str, Any] = {"batch_size": padded_bs}
    for slot in self._slots.values():
        if not slot.enabled or slot.buffer is None:
            continue
        # Structured slots use dotted names and are not top-level FB attributes.
        if "." in slot.name:
            continue
        is_computed = slot.post_fill is not None or not slot.copy_from_fb
        if (
            not is_computed
            and slot.source_fn is None
            and getattr(forward_batch_template, slot.name, None) is None
        ):
            # Absent this iter (fill_from skipped it): carry the template's value.
            continue
        replace_kwargs[slot.name] = slot.slice_for(padded_bs, padded_num_tokens)
    return dataclasses.replace(forward_batch_template, **replace_kwargs)
```

评论区精华

在 Review 中，chatgpt-codex-connector[bot] 指出一个 P1 问题：当 eager decode 运行 DP/MLP sync 时，prepare_mlp_sync_batch 已在 ForwardBatch 中写入了正确的 global_num_tokens_gpu，但 build_decode_registry 默认注册了 computed slot（copy_from_fb=False），extract_buffer 始终暴露 computed slot，而 require_gathered_buffer=False 时 post_fill 是 no-op 导致 buffer 为零，从而覆盖了 batch

的正确值。作者 ch-wan 确认并修复：为 build_decode_registry 添加 register_global_num_tokens 参数（默认 True），*eager decode* 传递 False 跳过该 slot 注册，batch 携带着原有值。

- DP attention 下 global_num_tokens_gpu 被零 buffer 覆盖 (correctness): ch-wan 确认并修复：向 build_decode_registry 添加 register_global_num_tokens 参数，*eager decode* 传递 False，跳过该 slot 注册，batch 值被携带。

风险与影响

- 风险：
  1. 性能退化：默认情况下 *eager decode* 路径每次 forward 新增一次 *device-to-device* 拷贝（fill_from），可能增加解码延迟。作者未提供基准数据，但提供了 SGLANG_EAGER_INPUT_NO_COPY 逃生口。
  2. DP attention 路径风险：虽然 register_global_num_tokens=False 已修复，但可能还有其他依赖 *computed* slot 的路径尚未排查。
  3. pdmux 和 forward_split_prefill 被显式排除，若将来需要统一则需额外适配。
  4. 修改了 ForwardBatch 视图构建方式，可能影响其他直接操作 ForwardBatch 字段的代码段。 - 影响：影响所有使用 *eager* 前向的模型（默认模式下 *decode/extend/idle* 均受影响）。提供了逃生环境变量，可缓解性能影响。CI 测试覆盖了 *greedy* 生成字节一致性和 TBO filter_batch 行为。需要 GPU CI 验证 DP attention 端到端场景。 - 风险标记：核心路径变更，性能退化可能性，DP attention 协同风险

关联脉络

- 暂无明显关联 PR