PR #25774 完整报告

sgl-project/sglang

drop output ids

合并时间：2026-05-20 08:50

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25774

执行摘要

- 一句话：移除 ScheduleBatch.output_ids，统一使用 input_ids 作为跨迭代桥梁
- 推荐动作：本 PR 是一次关键的内部重构，值得 SRT 调度层开发者精读。尤其注意重叠模式下占位符的使用方式，以及后续 PR #25819 将继续清理 prepare_for_extend 中的冗余移位。

功能与动机

Eliminate the SB.output_ids field — a forward-stream-produced tensor that was leaking onto the schedule stream. input_ids becomes the unified cross-iter bridge: end-of-forward sets it (placeholder for overlap, real tokens for non-overlap); next iter's resolve_future consumes it. (来自 PR body)

实现拆解

1. 移除属性定义：在 ScheduleBatch 类（schedule_batch.py）中删除 output_ids: torch.Tensor 属性定义，将所有内部引用（prepare_for_decode、filter_batch、merge_batch）中对 output_ids 的访问和赋值替换为对 input_ids 的访问。
2. 重写 run_batch 出口：在 scheduler.py 的 run_batch 方法中，不再使用临时变量 future_indices_or_next_token_ids，而是直接在各个分支中将 batch.input_ids 设置为占位符（重叠模式下为 -future_indices.indices，非重叠模式下为 next_token_ids.to(torch.int64)）。注意 PP 和 DLLM 返回类型可能不是 Tensor，需用 isinstance 保护。
3. 调整 prepare_for_decode 入口：在 ScheduleBatch.prepare_for_decode 中删除原来的 self.input_ids = self.output_ids.to(torch.int64); self.output_ids = None 赋值，因为 input_ids 已在上一轮 run_batch 结束时设置完成。penalizer 累积改为直接使用 self.input_ids。
4. 适配解聚解码路径：在 decode_schedule_batch_mixin.py 的 process_prebuilt 中，不再将 self.output_ids 作为列表积累并转为张量，而是使用本地列表 last_tokens 构造 last_tokens_tensor，然后仅在非 spec 场景下赋值给 self.input_ids；spec 场景下 last_tokens_tensor 作为 bonus_tokens 传入 spec_info。
5. 同步 MLX 和 PP 适配：在 MLX 后端的 _finalize 中，将 pending.batch_copy.output_ids = result.next_token_ids 改为 pending.batch_copy.input_ids = result.next_token_ids;

在 PP mixin 的 _pp_prep_batch_result 中将 batch.output_ids =pp_outputs["next_token_ids"] 改为 batch.input_ids =pp_outputs["next_token_ids"].to(torch.int64)。

6. 更新 benchmark 脚本：在 bench_one_batch.py 的 *decode* 函数中，将 batch.output_ids = input_token_ids 改为 batch.input_ids = input_token_ids.to(torch.int64)。

关键文件：

- python/sglang/srt/managers/scheduler.py（模块 调度器；类别 source；类型 core-logic；符号 run_batch, _build_hisparse_decode_batch）：核心调度循环，run_batch 出口处修改 input_ids 设置逻辑，是本次变更的主战场。
- python/sglang/srt/managers/schedule_batch.py（模块 批处理；类别 source；类型 core-logic；符号 output_ids, prepare_for_decode, filter_batch, merge_batch）：定义 ScheduleBatch 类，移除 output_ids 属性并调整 prepare_for_decode、filter_batch、merge_batch 中的引用。
- python/sglang/srt/disaggregation/decode_schedule_batch_mixin.py（模块 解聚解码；类别 source；类型 dependency-wiring；符号 process_prebuilt）：process_prebuilt 方法中不再设置 self.output_ids，改用本地列表并赋值给 self.input_ids，同时避免在 spec 路径下覆盖 prefill 的 input_ids。
- python/sglang/srt/hardware_backend/mlx/scheduler_mixin.py（模块 MLX 后端；类别 source；类型 core-logic；符号 _finalize）：MLX 后端的 _finalize 中 batch_copy.output_ids 改为 batch_copy.input_ids，以匹配新的属性名。
- python/sglang/srt/managers/scheduler_pp_mixin.py（模块 流水线并行；类别 source；类型 core-logic；符号 _pp_prep_batch_result）：PP 路径中 batch.output_ids = pp_outputs['next_token_ids'] 改为 batch.input_ids = ...，并添加 to(torch.int64) 转换。
- python/sglang/bench_one_batch.py（模块 基准工具；类别 source；类型 core-logic；符号 *decode*）：benchmark 脚本中的 *decode* 函数更新赋值。

关键符号：prepare_for_decode, process_prebuilt, run_batch, _build_hisparse_decode_batch, filter_batch, merge_batch, _finalize, _pp_prep_batch_result, *decode*

关键源码片段

python/sglang/srt/managers/scheduler.py

核心调度循环，run_batch 出口处修改 input_ids 设置逻辑，是本次变更的主战场。

```
# === scheduler.py (extract from run_batch) ===
if self.enable_overlap:
    with self._overlap_forward_isolation(batch):
        bs = len(batch.seq_lens)
        future_indices = self.future_map.alloc_future_indices(bs)
        with self.forward_stream_ctx:
            self.forward_stream.wait_stream(self.schedule_stream)
            self.future_map.resolve_future(batch)
```

```
            batch_result = self.model_worker.forward_batch_generation(batch)
            if batch_result.extra_keep_alive_refs:
                self.batch_record_buf[self.batch_record_ct].extend(
                    batch_result.extra_keep_alive_refs
                )
            batch_result.copy_done = self.device_module.Event()
            if batch_result.delay_sample_func is None:
                self.future_map.store_to_map(future_indices, batch_result)
                batch_result.copy_to_cpu(...)
            else:
                batch_result.future_indices = future_indices

        # 占位符：下一轮 resolve_future 会从 token_ids_buf 中
        # 通过负索引取真实 token
        batch.input_ids = -future_indices.indices
    else:
        batch_result = self.model_worker.forward_batch_generation(batch, **kwargs)
        if isinstance(batch_result.next_token_ids, torch.Tensor):
            # PP 中间 rank 返回 None，DLLM 返回 per-req list
            batch.input_ids = batch_result.next_token_ids.to(torch.int64)
```

python/sglang/srt/managers/schedule_batch.py

定义 ScheduleBatch 类，移除 output_ids 属性并调整 prepare_for_decode、filter_batch、merge_batch 中的引用。

```
# === schedule_batch.py (class ScheduleBatch) ===
class ScheduleBatch:
    # ... 其他属性 ...
    input_ids: torch.Tensor = None # shape: [b], int64, 跨迭代 token 桥
    # output_ids 属性被完全移除

    def prepare_for_decode(self):
        # ...
        if self.sampling_info.penalizer_orchestrator.is_required:
            # 之前使用 self.output_ids.to(torch.int64)，现直接使用
            # 已经在 run_batch 末尾设置好的 input_ids
            self.sampling_info.penalizer_orchestrator.cumulate_output_tokens(
                self.input_ids
            )
        # 原 self.input_ids = self.output_ids.to(torch.int64); self.output_ids = None 被移除
        # ...

    def filter_batch(self, ...):
        # ...
        if self.input_ids is not None:
            self.input_ids = self.input_ids[keep_indices_device]
        # 原 self.output_ids 相关的处理已全部替换

    def merge_batch(self, other):
```

```
        # ...
        if self.input_ids is not None and other.input_ids is not None:
            self.input_ids = torch.cat([self.input_ids, other.input_ids])
        # 注意：边界条件可能导致 data loss，详见风险分析
```

python/sglang/srt/disaggregation/decode_schedule_batch_mixin.py

process_prebuilt 方法中不再设置 self.output_ids，改用本地列表并赋值给 self.input_ids，同时避免在 spec 路径下覆盖 prefill 的 input_ids。

```
# === decode_schedule_batch_mixin.py ===
def process_prebuilt(self, server_args, future_map):
    """Assign the buffered last input id to schedule batch"""
    last_tokens: List[int] = []
    for req in self.reqs:
        last_tokens.append(req.output_ids[-1])
        # grammar 等处理 ...
    last_tokens_tensor = torch.tensor(
        last_tokens, dtype=torch.int64, device=self.device
    )
    if self.spec_algorithm.is_eagle():
        # spec 路径：last_tokens_tensor 作为 bonus_tokens 传入
        spec_info.prepare_for_extend(self) # 保留 prefill prompt 的 input_ids
        spec_info.bonus_tokens = last_tokens_tensor
        # ...
    else:
        # 非 spec：直接作为下一轮 decode 的输入
        self.input_ids = last_tokens_tensor
```

评论区精华

Review 中 gemini-code-assist[bot] 指出在 merge_batch 方法中，对 input_ids 的合并逻辑可能存在问题：如果一个批次的 input_ids 是 None 而另一个不是，会导致数据丢失或形状不匹配。该评论未得到作者公开回应，PR 即已合并。建议后续跟踪此边界情况的修复。

- merge_batch 中 input_ids 合并边界条件 (correctness): PR 已合并，但该问题未公开回复，也未见后续修复。

风险与影响

- 风险：主要风险：1）output_ids 属性被移除，任何外部代码或未合并分支中直接访问 batch.output_ids 的地方将报错。2）merge_batch 中 input_ids 的合并逻辑不完善，可能导致 data loss（当 self.input_ids 为 None 时）。3）PP 中间 rank 返回 None，DLLM 返回 list，这些场景下 batch.input_ids 不会被设置，依赖其非 None 的后续操作可能出错。4）重叠模式下使用负索引作为占位符，若 resolve_future 未能正确解析，可能导致错误的 token 输入。
- 影响：影响范围限于 SRT 调度层内部，对最终用户透明。但所有涉及 ScheduleBatch 构造、拷贝、合并的路径（包括 speculative decoding、disaggregation、PP、MLX）均需确保使用新的 input_ids 语义。影响程度中等，变更机械但需全面审查调用点。性能方面，减少了

跨流张量传输和一次 to(torch.int64) 转换，预期在重叠模式下有微小提升。

- 风险标记：merge_batch 边界条件待修复，output_ids 属性移除需审计所有调用方，重叠模式占位符依赖 resolve_future 正确实现

关联脉络

- PR #25819 drop dead prepare_for_extend shift in disagg prebuilt: PR body 明确指出此 PR 的 follow-up，将清理解聚预构建路径中死掉的 prepare_for_extend 移位逻辑。