PR #29429 完整报告

sgl-project/sglang

Let the presence of req.kv indicate the existence of owned kv resources

合并时间：2026-07-15 14:47

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/29429

执行摘要

- 一句话：用 req.kv 存在性表示 KV 资源是否已分配
- 推荐动作：值得精读。该 PR 是 SGLang KV 生命周期解耦系列的重要基石，展示了如何用类型系统和对象存在性代替代理字段管理资源。阅读时可以重点关注 schedule_batch.py 中 Req 的初始化、reset_for_retract 中的断言以及 mem_cache/common.py 中 release_kv_cache 的结尾变更。

功能与动机

原代码中 req.kv 总是存在一个占位对象，导致无法区分“未分配 KV”和“KV 已分配但 kv_allocated_len=0”。该 PR 通过让 req.kv 的存在性直接表示资源所有权，使 allocate/free 路径只在实际分配 / 释放 KV 时操作 req.kv，为后续进一步解耦做准备。

实现拆解

1. 变更字段默认值：在 Req.__init__（schedule_batch.py）中，将 self.kv: Optional[ReqKvInfo] = None 替代原来的 ReqKvInfo(...)，消除零占位对象。
2. 删除占位工厂函数：在 streaming_session.py 中删除 _new_kv() 函数，SessionSlot.kv 字段也改为 Optional[ReqKvInfo] = None；在 save_from_req 和 try_cache_finished_req 中增加 req.kv = None 以在所有权转移后清除引用。
3. 在分配点创建 ReqKvInfo：DecodeWorker._pre_alloc（disaggregation/decode.py）中，当 req.kv is None 时才创建 ReqKvInfo，否则只更新 kv_allocated_len；prepare_for_extend（schedule_batch.py）中类似处理。
4. 在释放点置 None：release_kv_cache（mem_cache/common.py）末尾增加 req.kv = None；free_swa_out_of_window_slots 开头增加 if req.kv is None: return 保护；scheduler_pp_mixin.pyprofile_and_init_predictor 中释放时也置 None。
5. 适配不变检查：invariant_checker.py 中在遍历 reqs 时跳过 req.kv is None 的请求。
6. 测试文件：test_unified_radix_cache_unittest.py 调整测试断言以适配新语义。

关键文件：

- python/sglang/srt/managers/schedule_batch.py（模块 调度批次；类别 source；类型 core-logic；符号 Req.init, Req.reset_for_retract, Req.prepare_for_extend）：Req 类的 kv 字段定义和生命周期管理的核心变更所在

- python/sglang/srt/session/streaming_session.py（模块 流式会话；类别 source；类型 core-logic；符号 _new_kv (removed), SessionSlot.init, SessionSlot.save_from_req, StreamingSession.try_cache_finished_req）：SessionSlot 的 kv 字段及所有权传递逻辑
- python/sglang/srt/disaggregation/decode.py（模块 解码分配；类别 source；类型 dependency-wiring；符号 DecodeWorker._pre_alloc）：解码时 KV 预分配路径创建 ReqKvInfo
- python/sglang/srt/mem_cache/common.py（模块 缓存层；类别 source；类型 core-logic；符号 free_swa_out_of_window_slots, release_kv_cache）：KV 缓存释放时置 None，SWA 滑动窗口释放保护
- python/sglang/srt/managers/scheduler_components/invariant_checker.py（模块 不变检查；类别 source；类型 core-logic；符号 _add_owner (loop integration)）：不变检查适配 req.kv 可能为 None
- python/sglang/srt/managers/scheduler_pp_mixin.py（模块 流水线调度；类别 source；类型 core-logic；符号 profile_and_init_predictor）：流水线调度 profiling 释放时置 None

关键符号：Req.init, Req.reset_for_retract, Req.prepare_for_extend, SessionSlot.init, SessionSlot.save_from_req, StreamingSession.try_cache_finished_req, DecodeWorker._pre_alloc, free_swa_out_of_window_slots, release_kv_cache, profile_and_init_predictor

关键源码片段

python/sglang/srt/managers/schedule_batch.py

Req 类的 kv 字段定义和生命周期管理的核心变更所在

```
# file: python/sglang/srt/managers/schedule_batch.py

class Req:
    def __init__(self, ...):
        # ...
        # 之前：self.kv: ReqKvInfo = ReqKvInfo(kv_allocated_len=0, swa_evicted_seqlen=0)
        # 现在：初始化为 None，真正分配时才会创建 ReqKvInfo 对象
        self.kv_committed_len = 0
        self.kv: Optional[ReqKvInfo] = None
        # ...

    def reset_for_retract(self):
        # ...
        # 调用 reset_for_retract 之前，kv 应已在 release 路径中被置为 None
        assert self.kv is None, "expect it is already released"
        self.kv_committed_len = 0
        # 原先还重置 self.kv.kv_allocated_len = 0 和 self.kv.swa_evicted_seqlen = 0，现在不再需要
        # ...
```

python/sglang/srt/session/streaming_session.py

SessionSlot 的 kv 字段及所有权传递逻辑

```
# file: python/sglang/srt/session/streaming_session.py

# 之前存在 _new_kv 工厂函数，现已删除
# def _new_kv() -> ReqKvInfo:
# from sglang.srt.managers.schedule_batch import ReqKvInfo
# return ReqKvInfo(kv_allocated_len=0, swa_evicted_seqlen=0)

@dataclass
class SessionSlot:
    # ...
    # KV pool state (None means no KV is currently held by this slot)
    req_pool_idx: Optional[int] = None
    kv_committed_len: int = 0
    kv: Optional[ReqKvInfo] = None # 原先为 field(default_factory=_new_kv)
    # ...

    def save_from_req(self, req: Req, is_first: bool):
        # ...
        # 从 req 拷贝 kv 状态后，将 req.kv 置为 None 以标记所有权转移
        req.req_pool_idx = None
        req.kv = None # 新增行
        req.mamba_pool_idx = None
        # ...
```

评论区精华

Review 由 Gemini Code Assist 自动生成，提出两点建议：

- 在 reset_for_retract 中的 assert self.kv is None 可能在意外状态时造成服务器崩溃，建议改为安全日志并清理。
- 在 _pre_alloc 中的 import ReqKvInfo 在条件内部会导致热路径开销，建议移至函数顶部。

两条建议均未被采纳，最终合并保留了原有代码。

- assert 在 reset_for_retract 中可能引起崩溃 (correctness): 未采纳，作者保留了 assert 以在开发阶段捕获违反契约的情况。
- import ReqKvInfo 在热路径条件内部 (performance): 未采纳，作者保留在条件内部以避免循环依赖（可能）。

风险与影响

- 风险：主要风险在于遗漏某些代码路径仍假设 req.kv 永不为 None，可能引发 AttributeError。但作者已逐一检查并修复了所有相关引用点（8 个文件），并在 free_swa_out_of_window_slots 和 invariant_checker 中添加 None 保护。assert 在生产环境中可能被禁用（-O），但默认启用，若意外触发将导致进程退出，但概率极低。整体风险可控。
- 影响：影响范围包括所有涉及 KV 资源管理的模块：调度批次、流式会话、前缀缓存、分解解码、不变检查以及流水线调度。由于语义变更为等价变换（行为不变），对用户透明；对后续开发者则需要理解 req.kv is None 的含义，不能依赖其始终存在。团队需要确保未来

新增的代码也遵循此契约。

- 风险标记：核心路径变更，assert 潜在崩溃风险，热路径 import 开销未优化

关联脉络

- PR #29430 Fix abusing presence of req.req_pool_idx to indicate the presence of req.kv resources: 同一解耦链条中的后续修复，直接依赖本 PR 的语义变更
- PR #29431 Lightweight extract allocation logic from mem_cache/common.py to more clearly show nearly parallel variants: 基于本 PR 的 req.kv 语义，进一步提取分配逻辑
- PR #29432 Fix bookkeeping fields not encapsulated with real allocations in normal alloc, PD pre-alloc, DFlash and EAGLE: 进一步封装 KV 记账到分配函数，依赖本 PR 引入的 req.kv 存在性语义