PR #26882 完整报告

sgl-project/sglang

fix(mlx): set canary_manager and materialize overlap-loop inputs on Apple Silicon

合并时间：2026-06-04 00:03

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26882

执行摘要

- 一句话：修复 MLX 后端 canary_manager 缺失与 overlap 循环输入缺失
- 推荐动作：建议精读以理解 MLX 后端与主调度器的交互细节。重点关注 scheduler.py 中 FutureMap 初始化顺序的调整，以及 resolve_forward_inputs 在 overlap 循环中的正确插入点。测试代码展示了如何为硬件后端驱动调度循环的单元测试模式。

功能与动机

修复 Issue #26832 中描述的 MLX 后端在 macOS 上启动崩溃。PR body 指出两个独立问题：

1) MlxModelRunnerStub 缺少 canary_manager 属性； 2) 重叠调度循环未调用 resolve_forward_inputs，导致预填充阶段出现 None input_ids。MLX 后端用户（Apple Silicon）无法正常使用 sglang serve。

实现拆解

实现包含以下步骤：

1. 添加 canary_manager 类属性：在 python/sglang/srt/hardware_backend/mlx/model_runner_stub.py 的 MlxModelRunnerStub 类中，添加 canary_manager = None 作为类属性。因为基类 ModelRunner 在 initialize() 中通过 install_canary() 设置该属性，而 MlxModelRunnerStub 的轻量 initialize() 跳过了该步骤。下游消费者（scheduler、CUDA graph runner、speculative workers）都守卫 canary_manager is not None，因此 None 是安全的默认值。
2. 调整 init_overlap 中 FutureMap 创建顺序：在 python/sglang/srt/managers/scheduler.py 的 init_overlap 方法中，将 FutureMap 的创建（原本在 MLX 特例之后）提前到 MLX 特例之前。这样即使在 MLX 路径下，future_map 也会被初始化，使得非重叠调度路径仍然能使用 FutureMap 进行 decode 的 input_ids relay。MLX 专属结果队列在特例中独立创建。
3. 在 MLX overlap 循环中调用 resolve_forward_inputs：在 python/sglang/srt/hardware_backend/mlx/scheduler_mixin.py 的 event_loop_overlap_mlx 的内部函数 _launch_fresh 中，新增一行 resolve_forward_inputs(batch, self.future_map)。这使得 batch.input_ids 从 CPU staging（prefill）或 FutureMap relay（decode）中 materialize，避免了异步前向调用解引用 None input_ids。

4. 添加回归测试：在 test/registered/unit/hardware_backend/mlx/test_attention_patching.py 中新增两个测试：
   - test_mlx_scheduler_init_overlap_keeps_future_map_relay：验证 init_overlap 在 MLX 模式下仍会创建 future_map 且可正常 stash 数据。
   - test_overlap_loop_materializes_prefill_input_ids：通过驱动一次 overlap 循环迭代，断言 input_ids 已 materialize，否则测试因 _StopLoop 异常提前退出。

关键文件：

- test/registered/unit/hardware_backend/mlx/test_attention_patching.py（模块 MLX 测试；类别 test；类型 test-coverage；符号 test_mlx_scheduler_init_overlap_keeps_future_map_relay, test_overlap_loop_materializes_prefill_input_ids, _StopLoop, fake_forward）：新增两个回归测试，直接验证修复，防止回归。
- python/sglang/srt/managers/scheduler.py（模块 调度器；类别 source；类型 core-logic；符号 init_overlap）：核心调度器，调整 init_overlap 中 FutureMap 创建顺序，影响所有 MLX 调度路径。
- python/sglang/srt/hardware_backend/mlx/model_runner_stub.py（模块 模型运行器；类别 source；类型 data-contract）：添加 canary_manager 类属性，解决启动崩溃。
- python/sglang/srt/hardware_backend/mlx/scheduler_mixin.py（模块 调度混合；类别 source；类型 dependency-wiring；符号 _launch_fresh）：在 MLX overlap 循环中调用 resolve_forward_inputs，修复预填充崩溃。

关键符号：init_overlap, _launch_fresh, test_mlx_scheduler_init_overlap_keeps_future_map_relay, test_overlap_loop_materializes_prefill_input_ids

关键源码片段

python/sglang/srt/managers/scheduler.py

核心调度器，调整 init_overlap 中 FutureMap 创建顺序，影响所有 MLX 调度路径。

```
def init_overlap(self):
    self.device_module = torch.get_device_module(self.device)

    # FutureMap 是始终开启的，用于两种模式下的 input_ids relay
    if self.draft_worker is not None:
        attn_backends = getattr(
            self.draft_worker,
            "spec_v2_attn_backends",
            (self.tp_worker.model_runner.attn_backend,),
        )
    else:
        attn_backends = (self.tp_worker.model_runner.attn_backend,)
    needs_cpu_seq_lens = decide_needs_cpu_seq_lens(self.server_args, attn_backends)
    self.future_map = self.spec_algorithm.create_future_map(
        self.device,
        self.req_to_token_pool,
        needs_cpu_seq_lens=needs_cpu_seq_lens,
```

```
    )

    if use_mlx():
        # MLX 使用自己的 overlap 循环，不创建 CUDA 流，
        # 但正常非重叠调度路径仍然通过 FutureMap 中继 decode input IDs
        self.result_queue: Deque = deque()
        return

    # forward_stream_ctx / copy_stream 也被 PP（非重叠）使用
    self.forward_stream_ctx: CudaStreamContext = self.device_module.stream(
        self.forward_stream
    )
    self.copy_stream: CudaStream = self.device_module.Stream()
    self.copy_stream_ctx: CudaStreamContext = self.device_module.stream(
        self.copy_stream
    )

    if not self.enable_overlap:
        return

    self.batch_record_buf = [None] * 2
    self.batch_record_ct = 0
```

python/sglang/srt/hardware_backend/mlx/model_runner_stub.py

添加 canary_manager 类属性，解决启动崩溃。

```
class MlxModelRunnerStub(ModelRunner):
    # MLX 路径不使用 KV canary，基类 ModelRunner 在完整 initialize() 中
    # 通过 install_canary() 设置该属性但这个轻量覆盖跳过了该步骤。
    # 下游消费者（scheduler、CUDA graph runner、speculative workers）
    # 都守卫 canary_manager is not None，因此 None 是安全默认值。
    canary_manager = None

    def __init__(self, *args, mlx_pool_size: int | None = None, **kwargs):
        self._mlx_pool_size = mlx_pool_size
        super().__init__(*args, **kwargs)
```

评论区精华

Review 讨论集中在验证修复和后续发现：

- yeahdongcn测试修复后服务器正常启动并给出示例输出，确认修复有效。
- SunRuiXin在 M3 上测试后报告了相同问题，并建议使用 --disable-overlap-schedule 作为临时绕过。
- yeahdongcn随后推送了新 commit 解决了 future_map.stash 的 AttributeError 问题（属于本 PR 的后续修复）。
- LijuanTang94在最终验证中确认端到端工作，chat completion 正确返回。

关键讨论点：与 #26952 的方案对比。本 PR 选择保留 FutureMap（MLX 路径也创建），并通过调整顺序让非重叠路径受益；#26952 则计划让 future_map 可为 None。#26952 的文档更新可以独立落地，两者在 model_runner_stub.py 上不冲突。

- 验证修复与下游 future_map 崩溃 (testing): yeahdongcn 最终修复，LijuanTang94 验证端到端工作。
- 与 #26952 方案对比 (design): 本 PR 维持现有方法，#26952 可后续独立落地。
- 临时绕过建议 (question): 该选项可绕过 crash，但本 PR 修复后无需使用。

风险与影响

- 风险：
  1. MLX 后端回归风险：变更仅影响 MLX 后端路径，CUDA 及其他后端不受影响。新增的两个单元测试覆盖了修复场景，降低回归可能。
  2. FutureMap 行为变化：原本 MLX 路径将 future_map 设为 None，现在改为创建实例。但下游代码已通过 is not None 守卫，且非重叠调度路径依赖于 future_map 的存在，因此实际无害。
  3. 性能影响：resolve_forward_inputs 是轻量操作，与 CUDA 路径一致，对端到端性能无影响。 - 影响：用户：MLX 后端（Apple Silicon）用户恢复可用，此前无法启动服务器。系统：只影响 MLX 后端，不改变其他后端行为。团队：维护者需注意本 PR 与 #26952 的关系，但两者兼容，文档更新可后续合并。
- 风险标记：MLX 后端变更，FutureMap 顺序调整，测试覆盖有限

关联脉络

- PR #26832 [Bug] AttributeError: 'MlxModelRunnerStub' object has no attribute 'canary_manager' when running with MLX backend on macOS: 原始 issue，描述崩溃问题，触发本 PR 修复。
- PR #26952 [MLX] Fix AttributeError in overlap scheduler on Apple Silicon: 相似修复方案，与本 PR 讨论对比，在 model_runner_stub.py 上不冲突。