PR #22493 完整报告

sgl-project/sglang

Add MambaPool kvcache offloading during retraction

合并时间：2026-04-22 08:51

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/22493

执行摘要

- 一句话：在请求回退期间添加 MambaPool 的 KV 缓存卸载，确保 Mamba 混合模型的完整状态保存和恢复。
- 推荐动作：该 PR 值得精读，重点关注 HybridLinearKVPool 的设计，学习如何通过可选参数扩展缓存池以支持多种状态类型的联合卸载，以及 CPU-GPU 数据传输的同步机制。

功能与动机

根据 PR body 描述，Mamba 混合模型（如 Qwen3.5-397B-A17B）将 SSM 状态（conv 和 temporal 缓冲区）存储在 MambaPool 中，与注意力 KV 缓存分离。在请求回退期间，只有 KV 缓存被保存到 CPU，而 Mamba 状态被丢弃，导致回退请求的生成损坏。

实现拆解

1. MambaPool CPU 快照方法：在 python/sglang/srt/mem_cache/memory_pool.py 的 MambaPool 类中添加 get_cpu_copy 和 load_cpu_copy 方法，使用 torch.cuda.synchronize() 确保数据一致性，将 conv 和 temporal 状态复制到 CPU 并恢复。
2. HybridLinearKVPool 扩展：在同一文件中扩展 HybridLinearKVPool 的 get_cpu_copy 和 load_cpu_copy 方法，通过可选 mamba_indices 参数联合处理 KV 缓存和 Mamba 状态的卸载，对非混合模型保持无操作。
3. 请求状态卸载集成：修改 python/sglang/srt/managers/schedule_batch.py 中 Req.offload_kv_cache 和 load_kv_cache 方法，传递 mamba_pool_idx 参数以触发完整状态卸载，并添加注释澄清 kv_cache_cpu 存储 KV 和 Mamba 状态元组。
4. 调度器日志增强：更新 python/sglang/srt/managers/scheduler.py 的 update_running_batch 方法，添加 #mamba_num_gained 日志以追踪回退期间 Mamba 池状态变化。
5. 测试配套：在 test/registered/unit/mem_cache/test_mamba_unittest.py 中添加两个单元测试 test_mamba_pool_cpu_offload 和 test_hybrid_kv_pool_cpu_offload，验证 CPU 卸载往返的正确性。

关键文件：

- python/sglang/srt/mem_cache/memory_pool.py（模块 内存缓存池；类别 source；类型 core-logic；符号 get_cpu_copy, load_cpu_copy）：实现 MambaPool 和

HybridLinearKVPool 的 CPU 卸载核心逻辑，是功能变更的主要文件。

- test/registered/unit/mem_cache/test_mamba_unittest.py（模块 单元测试；类别 test；类型 test-coverage；符号 test_mamba_pool_cpu_offload, test_hybrid_kv_pool_cpu_offload）：添加单元测试验证 MambaPool 和 HybridLinearKVPool 的 CPU 卸载功能正确性，确保回归覆盖。
- python/sglang/srt/mem_cache/allocator.py（模块 分配器；类别 source；类型 core-logic；符号 get_cpu_copy, load_cpu_copy）：修改 allocator 的 get_cpu_copy 和 load_cpu_copy 方法以支持 **kwargs 参数传递，确保与下层缓存池接口兼容。
- python/sglang/srt/managers/scheduler.py（模块 调度器；类别 source；类型 core-logic）：更新调度器以记录回退期间 Mamba 池的状态变化，增强日志和监控能力。
- python/sglang/srt/managers/schedule_batch.py（模块 调度批次；类别 source；类型 core-logic）：修改 Req 的 offload_kv_cache 和 load_kv_cache 方法，集成 Mamba 状态卸载，并添加注释澄清数据存储。

关键符号：get_cpu_copy, load_cpu_copy

关键源码片段

python/sglang/srt/mem_cache/memory_pool.py

实现 MambaPool 和 HybridLinearKVPool 的 CPU 卸载核心逻辑，是功能变更的主要文件。

```
class MambaPool:
    # ... 其他方法省略 ...

    def get_cpu_copy(self, indices, **kwargs):
        # 同步 CUDA 以确保数据一致性，防止异步操作导致状态不一致
        torch.cuda.synchronize()
        # 将 Mamba 缓存中的 conv 状态（列表形式）复制到 CPU，使用非阻塞传输提升性能
        conv_cpu = [
            conv[:, indices].to("cpu", non_blocking=True)
            for conv in self.mamba_cache.conv
        ]
        # 将 temporal 状态复制到 CPU
        temporal_cpu = self.mamba_cache.temporal[:, indices].to(
            "cpu", non_blocking=True
        )
        torch.cuda.synchronize()
        return conv_cpu, temporal_cpu # 返回 CPU 上的 Mamba 状态元组

    def load_cpu_copy(self, mamba_cache_cpu, indices, **kwargs):
        conv_cpu, temporal_cpu = mamba_cache_cpu
        torch.cuda.synchronize()
        # 将 CPU 上的 conv 状态恢复至 GPU，使用非阻塞传输
        for i, conv in enumerate(self.mamba_cache.conv):
            conv[:, indices] = conv_cpu[i].to(conv.device, non_blocking=True)
        # 恢复 temporal 状态
        self.mamba_cache.temporal[:, indices] = temporal_cpu.to(
```

```
            self.mamba_cache.temporal.device, non_blocking=True
        )
        torch.cuda.synchronize()
```

评论区精华

Review 中主要讨论了 kv_cache_cpu 的内容澄清：

- yizhang2077 询问 kv_cache_cpu 是否同时包含 KV 缓存和 Mamba 状态。
- hzh0425 进一步质疑 Mamba 状态存储位置。
- hlu1 澄清实现细节，指出 HybridLinearKVPool.get_cpu_copy 会复制两者到 CPU，并在后续提交中添加注释以明确。
- kv_cache_cpu 内容澄清 (correctness): hlu1 澄清 HybridLinearKVPool.get_cpu_copy 会复制两者到 CPU，并在后续提交中添加注释以明确实现细节。

风险与影响

- 风险：技术风险包括：
  - 状态不一致风险：如果 mamba_indices 传递错误或处理不当，可能导致 Mamba 状态未正确保存或恢复，影响生成准确性。
  - 性能影响：CPU 卸载涉及 GPU-CPU 数据传输和同步，可能增加延迟，尤其是在高频回退场景下。
  - 兼容性问题：修改了 get_cpu_copy 和 load_cpu_copy 接口为 **kwargs，需确保所有调用方兼容，但已通过默认参数保持向后兼容。
  - 影响：对使用 Mamba 混合模型的用户，修复了请求回退时生成损坏的问题，提升了系统健壮性和推理可靠性；对非混合模型无影响，因为变更通过可选参数实现。影响范围限于涉及缓存卸载和回退调度的模块，如内存池、调度器和请求管理。
  - 风险标记：状态一致性风险，核心路径变更

关联脉络

- PR #23300 [bug] Fix cache salt and extra keys for prefix cache isolation: 同样涉及 KV 缓存和调度器的修复，关注缓存状态一致性问题，与本 PR 在缓存管理方面有技术关联。
- PR #22911 [perf] support return_routed_experts with overlap scheduling: 涉及调度和性能优化，虽然专注于 MoE，但与本 PR 在调度器日志和状态管理方面有相似之处。