PR #30626 完整报告

sgl-project/sglang

[UnifiedTree]: Sync mamba int8 checkpoint

合并时间：2026-07-10 23:46

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/30626

执行摘要

- 一句话：为 UnifiedRadixTree 集成 int8 Mamba checkpoint 支持
- 推荐动作：建议所有涉及 Mamba 模型（如 GDN-hybrid、Qwen3-Next）部署的团队关注此 PR。值得精读的部分是 _commit_int8_checkpoint 如何利用 store_from_active 进行压缩，以及 _free_mamba_value 如何双路径释放。这体现了在不破坏现有路径的前提下扩展新功能的设计模式。

功能与动机

为了在 UnifiedRadixTree 模式下支持 int8 Mamba checkpoint，从而降低前缀缓存中 Mamba 状态的显存占用。该功能与已有的 --enable-int8-mamba-checkpoint 对齐，使其能在 UnifiedRadixTree 场景下生效。

实现拆解

1. 添加 int8 checkpoint 池属性和操作方法：在 MambaRadixCache 类中新增 int8_ckpt_pool 属性（通过 getattr 获取 req_to_token_pool.mamba_ckpt_pool），以及 _alloc_int8_ckpt_slot、_commit_int8_checkpoint 和 _free_mamba_value 方法。其中 _commit_int8_checkpoint 分配一个 int8 slot 并用 store_from_active 将活跃池中的数据压缩存储。
2. 修改 evict_component 中的释放逻辑：将直接调用 mamba_allocator.free 统一替换为 _free_mamba_value，该方法会根据 int8_ckpt_pool 是否存在来决定释放到 int8 池还是原始池。
3. 修改 prepare_for_caching_req 的 finished 和 donate 路径：当 int8_ckpt_pool 存在时，将原本直接赋值给 insert_params.mamba_value 的操作改为先调用 _commit_int8_checkpoint 压缩，再赋值。捐赠路径也类似，在分配新 slot 并获取捐赠数据后，通过 _commit_int8_checkpoint 压缩并释放原始活跃数据。
4. 修改 cleanup_after_caching_req 的清理逻辑：确保在 finished 请求的后续清理中，如果 mamba 值已存储在 int8 池，则不会二次释放到活跃池（通过检查 mamba_value_inserted 和 int8_ckpt_pool 状态）。
5. 新增测试覆盖：
   - 单元测试 TestUnifiedRadixCacheInt8MambaCheckpoint：构建带 enable_int8_mamba_checkpoint=True 的 fixture，缓存请求后验证活跃池大小不变、

int8 池大小减一、可驱逐大小增加。

- 端到端测试 TestUnifiedRadixTreeInt8MambaCheckpointE2E：继承现有 int8 checkpoint 测试，在 setUpClass 中设置环境变量 SGLANG_ENABLE_UNIFIED_RADIX_TREE=1 启动服务器，运行 KL 散度和 GSM8K 准确性测试。

关键文件：

- python/sglang/srt/mem_cache/unified_cache_components/mamba_component.py（模块 Mamba 组件；类别 source；类型 core-logic；符号 int8_ckpt_pool, _alloc_int8_ckpt_slot, _commit_int8_checkpoint, _free_mamba_value）：核心源码修改，添加了 int8 checkpoint 池的分配、提交和释放逻辑，重写了 evict、prepare_for_caching_req 和 cleanup_after_caching_req 中的关键路径以支持 int8 压缩存储。
- test/registered/unit/mem_cache/test_unified_radix_cache_unittest.py（模块 单元测试；类别 test；类型 test-coverage；符号 TestUnifiedRadixCacheInt8MambaCheckpoint, _make_req, _cache_finished, test_finished_req_stores_radix_mamba_state_in_int8_pool）：新增单元测试类 TestUnifiedRadixCacheInt8MambaCheckpoint，验证在缓存请求后 mamba 状态被正确存储到 int8 池，并检查池的可用大小变化。
- test/registered/radix_cache/test_int8_mamba_checkpoint_e2e.py（模块 端到端测试；类别 test；类型 test-coverage；符号 TestUnifiedRadixTreeInt8MambaCheckpointE2E, setUpClass, tearDownClass）：新增端到端测试子类 TestUnifiedRadixTreeInt8MambaCheckpointE2E，在启用 UnifiedRadixTree 环境下运行已有的 int8 checkpoint 测试，确保功能在真实模型上正确。

关键符号：_alloc_int8_ckpt_slot, _commit_int8_checkpoint, _free_mamba_value

关键源码片段

python/sglang/srt/mem_cache/unified_cache_components/mamba_component.py

核心源码修改，添加了 int8 checkpoint 池的分配、提交和释放逻辑，重写了 evict、prepare_for_caching_req 和 cleanup_after_caching_req 中的关键路径以支持 int8 压缩存储。

```
# 关键新增：int8 checkpoint 池的提交与释放

@property
def int8_ckpt_pool(self):
    # 从 req_to_token_pool 获取 int8 池，可能为 None（未启用时）
    return getattr(self.cache.req_to_token_pool, "mamba_ckpt_pool", None)

def _alloc_int8_ckpt_slot(self) -> torch.Tensor:
    """分配一个 int8 checkpoint slot，必要时触发驱逐"""
    slot = self.int8_ckpt_pool.alloc(1)
    if slot is None:
        self.cache.evict(EvictParams(num_tokens=0, mamba_num=1))
```

```
        slot = self.int8_ckpt_pool.alloc(1)
        assert slot is not None, "无法分配 int8 mamba checkpoint slot"
    return slot

def _commit_int8_checkpoint(self, active_slots: torch.Tensor) -> torch.Tensor:
    """将活跃池中的 mamba 状态压缩存储到 int8 池，返回 checkpoint slot"""
    ckpt_slot = self._alloc_int8_ckpt_slot()
    self.int8_ckpt_pool.store_from_active(
        self.cache.req_to_token_pool.mamba_pool,
        active_slots.view(-1),
        ckpt_slot,
    )
    return ckpt_slot

def _free_mamba_value(self, mamba_value: torch.Tensor) -> None:
    """根据 int8 池是否存在，选择释放到 int8 池或原始活跃池"""
    if self.int8_ckpt_pool is not None:
        self.int8_ckpt_pool.free(mamba_value)
    else:
        self.cache.req_to_token_pool.mamba_allocator.free(mamba_value)

# 在 prepare_for_caching_req 中，当请求完成且 int8 池存在时，使用 int8 压缩
# （简化自源码）
if is_finished:
    # ... 提取活跃值 active_value ...
    if self.int8_ckpt_pool is not None:
        insert_params.mamba_value = self._commit_int8_checkpoint(active_value)
    else:
        insert_params.mamba_value = active_value
```

评论区精华

仅有一个来自 @yuan-luo 的评论，他验证了该修改不会破坏 ReplaySSM 功能（'I've verified this function doesn't break ReplaySSM.'），未提出其他异议。PR 获得单方批准后合并。

- 不破坏 ReplaySSM 验证 (testing): 验证通过，未提出修改要求。

风险与影响

- 风险：主要风险在于 int8 量化引入的精度损失。虽然已有 KL 散度和 GSM8K 阈值测试，但若模型或数据分布变化，阈值可能失效。此外，int8_ckpt_pool 依赖 req_to_token_pool.mamba_ckpt_pool 存在，若服务器未正确初始化该池会触发 AttributeError；当前实现通过 getattr 返回 None 来优雅降级，但代码中存在 self.int8_ckpt_pool.alloc(1) 调用，如果池为 None 会报错，需确保调用点在池存在时才执行（当前通过在 prepare_for_caching_req 中先判断 self.int8_ckpt_pool is not None 来保护）。
- 影响：影响范围限于启用 --enable-int8-mamba-checkpoint 且使用 UnifiedRadixTree（通过 SGLANG_ENABLE_UNIFIED_RADIX_TREE 环境变量）的用户。对于这类用户，

Mamba 缓存状态的显存占用预计降低约 50%（bf16→int8），同时可能带来轻微的精度开销（已在测试中量化）。未启用的用户不受影响。

- 风险标记：int8 精度风险，依赖新池模块，UnifiedRadixTree 环境变量

关联脉络

- PR #30636 [UnifiedTree]: Sync Replay SSM: 同为 UnifiedTree 的 mamba 组件修改，修复了 ReplaySSM 在请求结束时缓存长度计算问题，与本 PR 有相同代码区域和功能关联。