PR #34808 完整报告

sgl-project/sglang

Fix mamba checkpoint depth under dcp

合并时间：2026-08-15 00:34

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/34808

执行摘要

- 一句话：修复 DCP 下 mamba checkpoint 深度不对齐问题
- 推荐动作：值得精读。这是 DCP 扩大树 page 与 mamba chunk 网格不一致导致隐式双重计算的典型正确性修复，核心设计决策是捐赠深度对齐树网格、内核快照保持 chunk 网格，并借此激活了此前不可达的 _force_track_h 分支。值得关注的三点： 1) math.lcm 归一化两个网格的简洁做法； 2) 通过提交历史展示的对 decode interval 守卫的自我回退与 resolve once for all consumers 的架构判断； 3) 测试用 tree_page_size 参数模拟 DCP 宽页的夹具设计。

功能与动机

PR body 明确指出：DCP sets the radix tree page to page_size * dcp_size, but the mamba prefill tracker still picks its checkpoint on mamba_cache_chunk_size. When those two grids differ, the donated state names a depth the tree cannot represent... a later request that hits that node resumes with a recurrent state which already covers tokens beyond the node, and those tokens go through the linear attention layers a second time. 在 Kimi-Linear 上 --dcp-size 8 --page-size 64 时，状态在 20416 被选取，但树最深节点只能表达到 19968，中间 448 个 token 被重复计算。

实现拆解

1. 在 python/sglang/srt/runtime_context.py 新增 mamba_checkpoint_grid(tree_page) 派生访问器，内部用 math.lcm(mamba_cache_chunk_size(), tree_page) 计算捐赠深度必须落在的网格；在非 DCP 场景下 chunk size 已是 max(model chunk size, page_size) 且二者整除，lcm 等于自身，行为不变。
2. 在 python/sglang/srt/managers/schedule_batch.py 的 _mamba_radix_cache_v2_req_prepare_for_extend 中，读取 self.tree_cache.page_size 计算 checkpoint_grid，将 mask 判断与 mamba_track_seqlen_aligned 的计算从 chunk_size 改为 checkpoint_grid，使捐赠深度落在树可表达的节点边界上；同时保留 mamba_track_fla_chunk_aligned 基于 chunk_size 计算，使 _force_track_h 分支在网格变粗时真正触发（此前两式恒等该分支不可达）。
3. 在 python/sglang/srt/mem_cache/unified_cache/components/mamba_component.py 中新增 mamba_checkpoint_grid 属性并在 finalize_match_result_in_tree_core 中用其计算 branching_seqlen，保证匹配侧的分支深度也落在同一网格上。

4. 新增 test/registered/unit/managers/test_mamba_checkpoint_depth.py 直接验证捐赠深度在 widen 时落在 tree page 上（20224）而不 widen 时保持 chunk 网格（20416）；扩展 test/registered/unit/mem_cache/test_unified_radix_cache_unittest.py 的 build_fixture 支持 tree_page_size 与 mamba_cache_chunk_size 参数，并新增 TestMambaCheckpointGrid 验证组件网格随树 page 变化。
5. 提交历史中曾尝试 reject decode track interval off the checkpoint grid 的守卫，但该守卫在 dcp8 默认 --mamba-track-interval 256 与 widened page 512 组合下会拒绝默认部署启动，随后被撤销（commit df117bd），decode 侧对齐留给后续统一解析。

关键文件：

- python/sglang/srt/managers/schedule_batch.py（模块 调度器；类别 source；类型 core-logic；符号 _mamba_radix_cache_v2_req_prepare_for_extend, mamba_checkpoint_grid）：核心修复点：_mamba_radix_cache_v2_req_prepare_for_extend 改用 checkpoint_grid（lcm(chunk, tree page)）计算捐赠深度，激活此前不可达的 _force_track_h 分支
- python/sglang/srt/runtime_context.py（模块 运行时上下文；类别 source；类型 core-logic；符号 mamba_checkpoint_grid）：新增核心派生访问器 mamba_checkpoint_grid，用 math.lcm 归一化 chunk 网格与树 page 网格，是非 DCP 场景行为不变的保证
- python/sglang/srt/mem_cache/unified_cache/components/mamba_component.py（模块 缓存组件；类别 source；类型 core-logic；符号 MambaComponent.init, finalize_match_result_in_tree_core, mamba_checkpoint_grid）：匹配侧 branching_seqlen 同步改用 mamba_checkpoint_grid，保证分支深度与捐赠深度在同一网格上，避免匹配端再次错位
- test/registered/unit/managers/test_mamba_checkpoint_depth.py（模块 调度器；类别 test；类型 test-coverage；符号 _track_seqlen, TestMambaCheckpointDepth, test_widened_tree_page_moves_the_donated_depth_onto_it, test_unwidened_tree_page_keeps_the_chunk_grid）：新增测试直接验证捐赠深度在 widened tree page 下落在 256 网格上（20224）而非 chunk 网格（20416），是核心行为回归测试
- test/registered/unit/mem_cache/test_unified_radix_cache_unittest.py（模块 缓存层；类别 test；类型 test-coverage；符号 build_fixture, TestMambaCheckpointGrid, _grid, test_grid_follows_the_widened_tree_page）：扩展 build_fixture 支持 tree_page_size/mamba_cache_chunk_size 参数并新增 TestMambaCheckpointGrid，验证缓存组件网格随树 page 变化

关键符号：mamba_checkpoint_grid, _mamba_radix_cache_v2_req_prepare_for_extend, MambaComponent.init, finalize_match_result_in_tree_core, build_fixture, _track_seqlen

关键源码片段

python/sglang/srt/runtime_context.py

新增核心派生访问器 mamba_checkpoint_grid，用 math.lcm 归一化 chunk 网格与树 page 网格，是非 DCP 场景行为不变的保证

```
# python/sglang/srt/runtime_context.py
# 在 mamba_cache_chunk_size 访问器之后新增。

import math

def mamba_checkpoint_grid(tree_page: int) -> int:
    """捐赠 mamba checkpoint 的深度必须落在的粒度，radix 树才能命名它。

    传入树实际分配的 page：DCP 会把它加宽到超过 mamba_cache_chunk_size，
    如果在这里重新推导加宽逻辑，就会复制缓存构建器里已有的谓词。
    """
    # lcm 的意义：chunk_size 与 tree_page 可能互不整除（如 64 与 256），
    # 捐赠深度必须同时是两者的整数倍才能被树节点承载；非 DCP 下
    # chunk_size 已是 max(model chunk, page_size) 且 assert 保证整除，
    # lcm 就等于 chunk_size 本身，行为不变。
    return math.lcm(mamba_cache_chunk_size(), tree_page)
```

python/sglang/srt/mem_cache/unified_cache/components/mamba_component.py

匹配侧 branching_seqlen 同步改用 mamba_checkpoint_grid，保证分支深度与捐赠深度在同一网格上，避免匹配端再次错位

```
# python/sglang/srt/mem_cache/unified_cache/components/mamba_component.py
# MambaComponent 构造与匹配结果最终化片段。

class MambaComponent(TreeComponent):
    def __init__(self, cache, params):
        super().__init__(cache, params)
        self.mamba_cache_chunk_size = mamba_cache_chunk_size()
        # params.page_size 是分配器实际使用的树 page，已被 dcp_size 加宽，
        # 因此它是 checkpoint 深度唯一能落下的网格。
        self.mamba_checkpoint_grid = mamba_checkpoint_grid(params.page_size)
        self.mamba_max_states_per_path = get_exec().mamba.mamba_max_states_per_path

    def finalize_match_result_in_tree_core(self, result, params, value_chunks, best_value_len):
        mamba_boundary_len = len(result.device_indices) + result.host_hit_length
        # Full KV 命中可能越过最新可复用的 mamba 状态：分支点取 Full-KV 命中
        # 范围内最后一个对齐到 checkpoint 网格的位置，保证分支状态同样可被
        # 树节点命名，避免回放时二度计算。
        aligned_seqlen = (
            result.full_kv_hit_length // self.mamba_checkpoint_grid
        ) * self.mamba_checkpoint_grid
        branching_seqlen = (
            aligned_seqlen if aligned_seqlen > mamba_boundary_len else None
        )
        return result._replace(mamba_branching_seqlen=branching_seqlen)
```

test/registered/unit/managers/test_mamba_checkpoint_depth.py

新增测试直接验证捐赠深度在 widened tree page 下落在 256 网格上（20224）而非 chunk 网格（20416），是核心行为回归测试

```
# test/registered/unit/managers/test_mamba_checkpoint_depth.py（新增文件）
# 用最小 fixture 直接驱动调度批次的 mamba 追踪逻辑，验证捐赠深度落在树 page 上。

def _track_seqlen(*, tree_page: int, prefix_len: int, extend_len: int) -> int:
    # 使用 dummy 模型与固定 chunk 64，避免加载真实 HF 配置。
    server_args = ServerArgs(model_path="dummy", page_size=CHUNK)
    server_args._mamba_cache_chunk_size = CHUNK
    set_global_server_args_for_scheduler(server_args)

    req = Req(rid="req", origin_input_text="",
              origin_input_ids=array("q", [1] * (prefix_len + extend_len)),
              sampling_params=SamplingParams(max_new_tokens=1), vocab_size=128)
    req.prefix_indices = torch.arange(prefix_len, dtype=torch.int64)
    req.set_extend_range(prefix_len, prefix_len + extend_len)

    batch = ScheduleBatch(reqs=[req])
    # 用 tree_cache.page_size 模拟 DCP 加宽后的树 page。
    batch.tree_cache = SimpleNamespace(page_size=tree_page)
    batch.req_to_token_pool = MagicMock()
    batch.req_to_token_pool.get_mamba_ping_pong_other_idx.return_value = 1

    batch._mamba_radix_cache_v2_req_prepare_for_extend(req)
    return req.mamba_last_track_seqlen

class TestMambaCheckpointDepth(unittest.TestCase):
    def test_widened_tree_page_moves_the_donated_depth_onto_it(self):
        # 16384 前缀 + 4066 extend：chunk 网格会停在 20416，256 token 的
        # 页面无法命名该深度，修复后应落在 20224（256 的整数倍）。
        depth = _track_seqlen(tree_page=256, prefix_len=16384, extend_len=4066)
        self.assertEqual(depth % 256, 0)
        self.assertEqual(depth, 20224)

    def test_unwidened_tree_page_keeps_the_chunk_grid(self):
        # 树 page 未加宽时保持 chunk 网格 20416，行为与修复前一致。
        depth = _track_seqlen(tree_page=CHUNK, prefix_len=16384, extend_len=4066)
        self.assertEqual(depth, 20416)
```

评论区精华

该 PR 无 review 评论，唯一审核来自 kpham-sgl 直接 APPROVED。设计讨论主要体现在提交历史中：作者自己发现 decode track interval guard 在 dcp8 默认几何下会拒绝所有默认部署启动（The guard rejected the default --mamba-track-interval 256 whenever the widened tree page is 512, which is the dcp8 geometry the fix itself targets），因此主动回退该提交，并指出 Decode still needs the same alignment, but the interval has to be resolved

once for all its consumers rather than checked in one of them。这是一个值得记录的自我纠错。

- decode track interval 守卫导致默认 dcp8 部署无法启动 (design): 作者回退守卫，指出 decode 侧对齐仍需解决，但 interval 应为所有消费者统一解析，而不是在某一处单独检查。
- PR 与 #34760 / #34780 的互补关系 (design): 作者确认两个守卫仍值得保留，作为不变式保护；本修复让它们在 DCP prefill 路径上停止触发，同时保留前缀复用。

风险与影响

- 风险：
  1. 行为变化集中在捐赠深度与 branching depth 的对齐上，测试覆盖了 widened（256）与 unwidened（64）两种网格，但未覆盖 DCP 下 512 页等更大几何；decode 侧 --mamba-track-interval 仍保持原网格，若用户配置的 interval 与 checkpoint_grid 不兼容，decode 段仍可能存在同类不对齐风险（作者在提交历史中承认仍需统一解析）。
  2. MambaComponent.finalize_match_result_in_tree_core 中 branching_seqlen 从 chunk 网格切到 checkpoint 网格，会影响 HiCache 下会话引用 / 分支状态的判定，5 个相关缓存测试已随 PR 跑过（/rerun-test 列出），但 on-device 长序列行为仍需关注。
  3. schedule_batch.py 是核心调度路径，mask 条件从 chunk_size >= 改为 checkpoint_grid >= 会改变 track 触发时机，可能影响 mamba ping-pong 缓冲区的交换节奏，尤其在 checkpoint_grid 大于 chunk_size 时 track 更稀疏。 - 影响：影响面集中在 DCP + mamba 线性注意力（如 Kimi-Linear）场景：修复前 prefix cache 恢复会二度计算最多 page_size * (dcp_size - 1) 个 token 的线性注意力，精度与性能均受损；修复后 cached_tokens 保持在 19968 而非回退到 16384，前缀复用率得以保留。对非 DCP 部署网格不变（lcm 等于 chunk size），行为零变化。团队侧这是 DCP 与 mamba 缓存集成正确性的关键修复，并补充了两个可复用的测试夹具参数（tree_page_size、mamba_cache_chunk_size），后续 DCP 相关测试可直接复用。 - 风险标记：核心调度路径变更，DCP 特定场景，decode 侧对齐未完全解决，测试覆盖有限

关联脉络

- PR #34760 Reject misaligned mamba checkpoint attachment at insert time: 与本 PR 同属一个缺陷的两个半区：本 PR 修复上游捐赠深度计算，34760 在插入侧拒绝不对齐的挂载，作为不变式守卫
- PR #34780 Reject misaligned mamba checkpoint attachment at insert time: 与 34760 相同防御方向，PR body 明确点名二者为本缺陷的下游守卫