

PR #35412 完整报告

sgl-project/sglang

[Fix] Land the decode mamba checkpoint depth on the tree page under DCP

合并时间: 2026-08-21 03:15

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/35412>

执行摘要

- 一句话: 修复 DCP 下 decode mamba checkpoint 深度落空树页网格的静默错误
- 推荐动作: 值得精读。核心看点: 用 lcm 派生网格同时满足 chunk、树页、用户间隔三个约束且不污染 ServerArgs 原始值; 对检查点深度 (checkpoint depth) 与 flush 节奏 (flush/write cadence) 两类读者做刻意区分; 以及“缓存复用结构完全不变、只改变挂载 state”的静态论证方法。建议与 #34808 一起阅读, 理解同一不变量在 extend 与 decode 两侧的实现差异; 同时关注 PR 遗留的解码半程精度验证缺失, 后续应有跟进 PR 补齐。

功能与动机

PR body 明确指出: 捐赠的 mamba checkpoint 只有在 radix 树能命名的深度才可复用, 而 `cache_finished_req / cache_unfinished_req` 会用 `page_aligned(tree_page)` 对插入键取整, 因此落在树页网格外的 checkpoint 会被挂到前一个节点——它的 state 却已覆盖越过该节点的 token, 后续请求匹配该节点时从中途恢复线性注意力递归并重新消费这些 token, 表现为 “Silent, no assert, wrong output”。#34808 修复了 extend 路径的 `mamba_checkpoint_grid(tree_page)`, 但明确声明 decode 仍使用自己的 `--mamba-track-interval` 网格, 而该网格在 DCP 下并不落在树页上: 守卫 `assert mamba_track_interval % view.page_size == 0` 只针对配置页, DCP 下树实际分配在 `page_size × dcp_size`, 故 break condition 为 `mamba_track_interval % (page_size × dcp_size) != 0`。两个已发布的 Kimi-K3 B300 recipe (`TestKimiK3B300Balanced` 与 `TestKimiK3B300MegaMoE`) 恰好命中坏行: 树页 512 而 decode 每 256 打点, 每个 256 的奇数倍命名的深度都没有节点承载。

实现拆解

1. 新增派生网格函数: 在 `python/sglang/srt/runtime_context.py` 的 `mamba_checkpoint_grid` 旁新增 `mamba_track_grid(tree_page)`, 对 `mamba_checkpoint_grid(tree_page)` (即 `mamba_cache_chunk_size()` 与 `tree_page` 的 lcm) 与 `get_exec().mamba.mamba_track_interval` 再取一次 lcm。这样 decode 捐赠深度同时满足三个约束: chunk 对齐 (extend 前缀依赖 `_force_track_h` 的断言)、树页网格 (DCP 下为 `page_size × dcp_size`)、用户配置的 track interval。函数为纯推导, 不修改 `server_args` 中操作员填写的原始值。
2. 调度与 spec 读取点替换: `schedule_batch.prepare_for_decode`、`spec_utils` 的 `spec_prepare_for_decode / commit_mamba_states_after_verify /`

_verify_commit_step_indices、dspark_worker_v2._commit_target_mamba_states_after_verify、dflash_worker_v2.update_target_mamba_state_after_verify 全部从 get_exec().mamba.mamba_track_interval 改为 mamba_track_grid(batch.tree_cache.page_size)，这些位置都持有 batch，可直接拿到树页句柄，与 extend 路径的取数方式一致。

3. batch_result_processor 边界点改造：_mamba_prefix_cache_update（known_boundary 断言）、_mamba_check_track_boundary、_mamba_lazy_spec_update 三处改用 mamba_track_grid(self.tree_cache.page_size)。原因是递给 processor 的 batch 是 result-queue 的 copy()，字段列表只覆盖 process_batch_result 所需，不携带 tree_cache，直接读 batch.tree_cache.page_size 会得到 None 并抛异常（commit fa6bc760 专门修复此坑）。hybrid_linear_attn_backend 的 ReplaySSM force-flush 掩码与 Inkling sconv 保留原始 mamba_track_interval，因为它们是 flush/write 节奏而非 checkpoint 深度；由于 lcm 保证派生网格是原始间隔的倍数，每个快照步骤仍是 flush 步骤，但触发步骤从“同一组”变为“嵌套集合”。
4. 测试配套：test_mamba_checkpoint_depth.py 新增 TestMambaTrackGrid 三个 CPU 用例（树页加宽向上取整、间隔已对齐保持不变、结果始终落在 chunk 上）；test_batch_result_processor_mamba_boundary.py 为手工构造的 ScheduleBatch 与 processor double 补 tree_cache；test_ngram_mamba_verify_update.py 因 spec_utils 不再导入 get_exec，把 patch 目标从 mamba_track_interval 改为 mamba_track_grid。CI 上 unit 测试 rerun 通过，但 e2e test_kimi_k3_b300.py 最后一次 rerun 失败（8-gpu-b300 ②），合并前未给出解释，作者预告的 cold-vs-warm 解码半程精度对比也未执行。

关键文件：

- python/sglang/srt/runtime_context.py（模块 运行上下文；类别 source；类型 core-logic；符号 mamba_track_grid）：新增 mamba_track_grid(tree_page)，是整个修复的网格推导入口，对 checkpoint depth、tree page、track interval 三者取 lcm
- python/sglang/srt/speculative/dspark_components/dspark_worker_v2.py（模块 投机解码；类别 source；类型 dependency-wiring；符号 _commit_target_mamba_states_after_verify）：DSPARK 验证路径在 verify 后提交 mamba 状态，是受影响 recipe TestKimiK3B300MegaMoE 的直接修复点
- python/sglang/srt/managers/scheduler_components/batch_result_processor.py（模块 结果处理；类别 source；类型 core-logic；符号 _mamba_prefix_cache_update，_mamba_lazy_spec_update，_mamba_check_track_boundary）：decode 边界打点与 known_boundary 断言所在，三处读取点统一改为从自身 tree_cache 派生网格；同时记录 result-queue copy() 不带 tree_cache 的坑
- python/sglang/srt/speculative/spec_utils.py（模块 投机解码；类别 source；类型 core-logic；符号 _verify_commit_step_indices，commit_mamba_states_after_verify，spec_prepare_for_decode）：共享 spec 提交与 prepare 路径的网格统一，影响所有投机解码后 mamba 状态提交

- python/sglang/srt/managers/schedule_batch.py (模块 调度器; 类别 source; 类型 core-logic; 符号 prepare_for_decode) : prepare_for_decode 是 decode 主路径打点入口, 与 extend 路径取数方式对齐
- python/sglang/srt/speculative/dflash_worker_v2.py (模块 投机解码; 类别 source; 类型 dependency-wiring; 符号 _update_target_mamba_state_after_verify) : dflash 验证路径同样是 spec 提交来源, 与 dspark 同步更换取数来源
- test/registered/unit/managers/test_mamba_checkpoint_depth.py (模块 测试配套; 类别 test; 类型 test-coverage; 符号 TestMambaTrackGrid, _grid, test_widened_tree_page_rounds_the_interval_up, test_interval_already_on_the_tree_page_is_untouched) : 新增 TestMambaTrackGrid 三个 CPU 用例, 覆盖树页加宽、间隔不变、chunk 对齐三种关键行为, 与 #34808 的 extend 用例并列
- test/registered/unit/managers/test_batch_result_processor_mamba_boundary.py (模块 测试配套; 类别 test; 类型 test-coverage) : 手工构造的 ScheduleBatch 与 processor double 原先不带 tree_cache, 现需补齐 (与 #34808 对其 double 的做法一致)
- test/registered/unit/spec/test_ngram_mamba_verify_update.py (模块 测试配套; 类别 test; 类型 test-coverage) : spec_utils 不再导入 get_exec 后, 原 patch 目标失效, 需改到 mamba_track_grid

关键符号: mamba_track_grid, _commit_target_mamba_states_after_verify, _update_target_mamba_state_after_verify, _verify_commit_step_indices, commit_mamba_states_after_verify, spec_prepare_for_decode, _mamba_prefix_cache_update, _mamba_check_track_boundary, _mamba_lazy_spec_update, prepare_for_decode

关键源码片段

python/sglang/srt/runtime_context.py

新增 mamba_track_grid(tree_page), 是整个修复的网格推导入口, 对 checkpoint depth、tree page、track interval 三者取 lcm

```
# mamba_cache_chunk_size(): 系统实际使用的 mamba 状态缓存粒度,
# 取模型 mamba chunk size 与 page_size 的最大值, 并缓存到 ServerArgs.
def mamba_cache_chunk_size() -> int:
    return get_server_args().mamba_cache_chunk_size
```

```
# mamba_checkpoint_grid(): extend 路径捐赠 checkpoint 的深度网格。
# DCP 会把树页加宽到 page_size * dcp_size, 所以这里必须传入树实际
# 分配的页面, 而不是配置里的 page_size, 否则推导会重复一份已经
# 住在 cache builder 里的谓词。
def mamba_checkpoint_grid(tree_page: int) -> int:
    return math.lcm(mamba_cache_chunk_size(), tree_page)
```

```
# mamba_track_grid(): decode 路径 (--mamba-track-interval) 的等价网格。
```

```
# decode 捐赠的深度除了要落在 chunk 与树页网格上，还必须是用户配置
# 间隔的倍数——lcm 保证三者同时成立，且不修改 ServerArgs 里操作员
# 填写的原始值，后续 flush/write 节奏类读者仍可读原始间隔。
```

```
def mamba_track_grid(tree_page: int) -> int:
    return math.lcm(
        mamba_checkpoint_grid(tree_page),
        get_exec().mamba.mamba_track_interval,
    )
```

python/sglang/srt/speculative/dspark_components/dspark_worker_v2.py

DSPARK 验证路径在 verify 后提交 mamba 状态，是受影响 recipe
TestKimiK3B300MegaMoE 的直接修复点

```
def _commit_target_mamba_states_after_verify(
    self,
    *,
    batch: ScheduleBatch,
    seq_lens_pre_verify: torch.Tensor,
    seq_lens_post_verify: torch.Tensor,
    commit_lens: torch.Tensor,
) -> None:
    # 只处理链式布局 (topk == 1)；树形布局需要共享 spec_utils 的索引映射
    assert get_spec().speculative_eagle_topk in (None, 1)
    attn_backend = self.target_worker.model_runner.attn_backend

    last_correct_step_indices = commit_lens.to(torch.int64) - 1
    mamba_steps_to_track = None

    if batch.mamba_track_indices is not None:
        # 关键修复点：DCP 下树页被加宽，直接使用 mamba_track_interval
        # 会让 checkpoint 深度落不到树节点上；改用派生的
        # mamba_track_grid，并从 batch 的 tree_cache 拿真实树页
        mamba_track_interval = mamba_track_grid(batch.tree_cache.page_size)
        to_track_mask = (
            seq_lens_pre_verify // mamba_track_interval
            != seq_lens_post_verify // mamba_track_interval
        )
        tracking_point = (
            seq_lens_post_verify // mamba_track_interval * mamba_track_interval
        )
        # 定位 verify 步序列中第一次跨过网格的那一步
        to_track_ith = torch.clamp(tracking_point - seq_lens_pre_verify - 1, min=0)
        can_track_mask = to_track_mask & (
            to_track_ith < commit_lens.to(to_track_ith.dtype)
        )
        mamba_steps_to_track = torch.where(
            can_track_mask,
            to_track_ith.to(torch.int64),
            torch.full_like(to_track_ith, -1, dtype=torch.int64),
```

```
)

# 把最后接受的 verify 步骤对应的 KDA/mamba 状态写入持久缓存
attn_backend.update_mamba_state_after_mtp_verify(
    last_correct_step_indices=last_correct_step_indices,
    mamba_track_indices=batch.mamba_track_indices,
    mamba_steps_to_track=mamba_steps_to_track,
    model=self.target_worker.model_runner.model,
    req_pool_indices=batch.req_pool_indices,
)
```

test/registered/unit/managers/test_mamba_checkpoint_depth.py

新增 TestMambaTrackGrid 三个 CPU 用例，覆盖树页加宽、间隔不变、chunk 对齐三种关键行为，与 #34808 的 extend 用例并列

```
class TestMambaTrackGrid(unittest.TestCase):
    # 通过 override_server_args 临时注入配置，验证 mamba_track_grid
    # 的三种关键行为：树页加宽时向上取整、间隔已对齐时保持不变、
    # 推导结果始终落在 mamba cache chunk 网格上。
    def _grid(self, *, interval: int, tree_page: int, chunk: int = CHUNK) -> int:
        with get_context().override_server_args(
            mamba_track_interval=interval, _mamba_cache_chunk_size=chunk
        ):
            return mamba_track_grid(tree_page)

    def test_widened_tree_page_rounds_the_interval_up(self):
        # 树页 512 而间隔 256: lcm 会把它抬到 512, checkpoint 深度才能落树
        self.assertEqual(self._grid(interval=256, tree_page=512), 512)

    def test_interval_already_on_the_tree_page_is_untouched(self):
        # 非 DCP 或树页等于间隔时，两个网格原本重合，结果保持不变
        self.assertEqual(self._grid(interval=256, tree_page=256), 256)
        self.assertEqual(self._grid(interval=256, tree_page=128), 256)
        self.assertEqual(self._grid(interval=256, tree_page=64), 256)

    def test_grid_stays_on_the_chunk_size(self):
        # chunk 成为主导约束：192 与 64 对齐但 128 未对齐时，取 128 的倍数
        self.assertEqual(self._grid(interval=192, tree_page=64, chunk=128) % 128, 0)
```

评论区精华

PR 无实质 review 评论 (ispobock 直接 APPROVED)，核心论述集中在 PR body 与提交信息中：

- “A donated mamba checkpoint is only reusable at a depth the radix tree can name... Silent, no assert, wrong output.”——点明问题的严重性在于完全静默。
- “The insert-time guard that would have caught this was deferred to #34760 and #34780 — both closed, never merged”——解释了为何 main 上没有任何检查能拦住这个错位。

- “What changes is which state is attached: previously a state that had already consumed 256 more tokens than the node it hung on.”——静态论证缓存复用结构不变，只改变挂载的 state。
- commit fa6bc760 记录了一个真实踩坑：“The batch handed to the result processor is the result-queue batch.copy(), whose field list is scoped to what process_batch_result needs and does not carry tree_cache — so batch.tree_cache.page_size was None there and raised on the...”，随后改为从 processor 自身的 self.tree_cache 读取。
- 作者在 body 中明确留白：“Accuracy Tests Not yet measured — this is why the PR is a draft.”，解码半程的 teacher-forced logprob 对比未完成，合并时仍以此状态合入。
- 解码半程准确度测试尚未测量 (testing): 合并时仍未提供解码半程的精度对比证据，仅以页面对齐的静态推导作为依据。
- result-queue batch.copy() 不携带 tree_cache 导致 AttributeError (correctness): 改为从 processor 自身的 self.tree_cache 读取 page_size，测试 double 同步补 tree_cache。
- CI rerun: unit 测试最终通过，e2e test_kimi_k3_b300.py 失败 (testing): e2e 失败在合并时未解决，与本 PR 解码半程精度验证缺失相互叠加，构成主要遗留风险。

风险与影响

- 风险：
 1. 解码半程正确性证据缺失：PR 自述未完成 #34808 同级别的 cold-vs-warm 精度对比，论证仅基于页面对齐的静态推导；合并前 CI 上 test_kimi_k3_b300.py 最后一次 rerun 失败 (workflow 32325747992)，两者叠加使修复效果缺乏端到端实证。
 2. flush 语义变化：hybrid_linear_attn_backend.py:403 注释原本保证 force-flush 掩码与 track 打点“fire on the same steps”，改动后变为嵌套步骤集合，ReplaySSM 快照写入频率减半；作者断言每个快照步骤仍是 flush 步骤，但这依赖 lcm 网格是原始间隔倍数这一性质，未来若有人改动 ReplaySSM 掩码的取数来源，该不变性会被破坏。
 3. checkpoint 稀疏化：若 mamba_track_interval 与实际树页、chunk 不存在约数关系，lcm 会显著放大网格，使 decode 打点变稀疏；本 PR 场景 (64/512/256) 仅翻倍到 512，但在更极端配置下可能有空间利用率下降。
 4. 测试补丁脆弱：test_ngram_mamba_verify_update.py 直接 patch mamba_track_grid 绑定，并依赖 spec_utils 不再导入 get_exec；后续重构若恢复该导入或改名，测试会以 AttributeError 形式失败。
 5. 上游守卫缺失仍在：#34760 / #34780 两个 insert-time mamba 深度检查均未合入，未来新的 page/dcp/interval 组合仍可能静默错位，本 PR 只是从源头让网格对齐，而非在插入点兜底。- 影响：用户侧：修复 Kimi-K3 B300 DCP8 (HiCache 与 DSPARK 两条 recipe) 下 decode 阶段 radix 缓存恢复导致的重放 token、错误输出的问题，受影响配置的 ping-pong 快照写入减半是附带收益；非 DCP 以及 page 64 × dcp 4 等原本对齐的配置完全无行为变化。系统侧：所有 mamba 混合架构的 decode checkpoint 深度统一由 mamba_track_grid 派生，后续新增读取点只需传入 tree_cache 即可获得一致性；batch_result_processor 对 self.tree_cache 的依赖成为新的隐式约束，测试 double 必须同步补齐。团队侧：暴露了 DCP 加宽树页后既有配置守卫失效的隐患，并提示

“flush 节奏”与“checkpoint 深度”两类读者需区分对待。 - 风险标记：核心调度与验证路径变更，解码半程准确度证据缺失，e2e 测试合并前失败，flush 触发步骤集合变化，上游 insert-time guard 仍缺失

关联脉络

- PR #34808 （材料未提供标题）：同一不变量的 extend 侧修复，引入 `mamba_checkpoint_grid(tree_page)`；本 PR 是其 decode 对应，并特意在旁边新增 `mamba_track_grid`
- PR #35380 （材料未提供标题）：DCP HiCache KL 场景测试；因 `page 64 × dcp 4` 与默认间隔 256 恰好对齐，无法暴露 decode 网格错位
- PR #34760 （材料未提供标题）：被 defer 的 insert-time mamba 深度检查之一，未合并，导致错位继续静默
- PR #34780 （材料未提供标题）：同 34760，另一个被 defer 的 insert-time guard，未合并
- PR #33477 （材料未提供标题）：引入 `test_batch_result_processor_mamba_boundary.py`，本 PR 为该测试的 double 补 `tree_cache`
- PR #35689 Skip empty linear-attention state buffers in PD transfer: 同为 mamba 状态与缓存传输正确性维护线，均在 DCP/PD 场景触及 mamba state 处理