

PR #35177 完整报告

sgl-project/sglang

feat(unified-memory): three sub-pools for mamba + hybrid-SWA models

合并时间: 2026-09-01 06:10

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/35177>

执行摘要

- 一句话: unified 池泛化至 N 子池, 新增浮动池支撑 mamba+SWA 三态 KV
- 推荐动作: 值得精读。三个设计决策尤其值得关注: 一是 `_CapacityField` 用数据描述符 + `epoch` 让容量 memo 失效 "按构造" 发生, 避免了散落的失效钩子; 二是 `_float_open_short_side` 的 demand-vector 策略, 使单带宽短供、耦合多带分配、未来组合准入向量统一为一种表达; 三是 `_relieve_for_alloc` 将 flush/relocate/evict 收敛为单一 ladder。建议结合 `test_unified_capacity_memo.py` 与 `test_multi_ended_allocator.py` 的测试方法论阅读, 这两份测试的防御深度 (属性测试 + 反射扫描 + 伪装写入探测) 本身就值得借鉴。

功能与动机

PR body 明确指出: "Unified memory currently supports exactly two sub-pools, so it cannot serve a model whose KV state comes in three kinds — full-attention KV, sliding-window KV, and recurrent (mamba/conv) state. Inkling is the first such model in tree." 静态池在 boot 时按固定比例砌墙, idle 容量无法互相调剂; request mix 在 boot 时不可知, 拆分是否合理无法预先确定。因此需要把 unified pool 泛化到 N 并让边界随运行负载移动。

实现拆解

变更按五层推进:

1. 池层泛化到 N 子池 (python/sglang/srt/mem_cache/unified_memory_pool.py) : `SubPoolSpec` 新增 `_allowed_grow_directions` ClassVar, `grow_direction` 从 up/down 扩展出 float; `UnifiedKVPool.__init__` 把 `len(sub_pool_specs) == 2` 断言放宽为 `>= 2`, 且强制恰好一个 grow-up 端、恰好一个 grow-down 端, float 数量任意, 并按 `[up, floats(输入序), down]` 规范化链序; `_reserved_floor_bytes` 覆盖 float 的 page-0 预留信封 (mamba 保持 `page_size=1` 不计 page 项)。输入顺序从此无关, 2 池布局 byte-identical (由 `test_unified_npool_sweep.py` 钉住)。
2. 分配器链式化 (python/sglang/srt/mem_cache/multi_ended_allocator.py) : `_peer` 单指针改为 `low_peer/high_peer` 链, `bind_peer` 保留为 2 池 END 对兼容 shim (拒绝 float 成员), 新增 `bind_low_peer/bind_high_peer`; `_current_gap_bytes` 等所有 peer 咨询路径改走透明链 walk (`_chain_low_frontier_above_bytes/_chain_high_frontier_below_bytes/_is_frontier_transparent`), 空 float 被跳过、首个不透明成员截断 gap。

test_multi_ended_allocator.py 的 golden identity 确保 2 池下链式 walk 与旧单 peer 闭式公式一致。

- FloatMultiEndedAllocator 与统一短供 ladder: float 中池 holes-first——free 在内部打洞零拷贝, alloc 先复用洞、再向更大间隙一侧扩展边界, 空池停在双端 frontier 的中点; make_room/compact_holes 是独立的数据搬运器, 仅在邻居短供时按需搬迁。短供处理收敛为模块级 _relieve_for_alloc: flush → 复查 → 让 float 滑动 → 复查 → 交给 evict/retract; _float_open_short_side 由 per-band demand vector 驱动, 单带短供 (如 mamba 态不足) 与复合体耦合多带分配走同一条路径。
- 容量视图 memo 化: 链式 walk 使容量计算比双端对更贵, 而调度器每步读 O(queue) 次; _CapacityField 数据描述符在每次 rebind 时推进 _capacity_epoch, available_size/schedulable_available_size 及复合 joint view 全部 epoch 键控缓存, 失效由构造保证。约束是这些字段只能 rebind、不能原地 mutate——test_unified_capacity_memo.py 专门验证绕过描述符的写入会被 verify_byte_accounting 的 idle 检查捕获。
- 路由、接线与配套: python/sglang/srt/mem_cache/kv_cache_configurator.py 在 mambaish && is_hybrid_swa && !dsv4 时优先走新增的 _init_unified_mamba_swa_pools (PD 组合显式拒绝); mamba 池的物理移动契约统一为 move_kv_cache (修复了 src/dst 参数顺序相反的隐患); per-step SWA free 接收 start_pos 并用算术派生 page 代表, 移除 torch.unique 带来的 host sync, 2 池 SWA 复合同样受益。测试配套为 7+ 文件: CPU 单测 (tri pool 契约、容量 memo 属性测试、N-pool 几何 sweep、no-host-sync 反射扫描) + GPU e2e (test_inkling_unified.py)。

关键文件:

- python/sglang/srt/mem_cache/multi_ended_allocator.py (模块 分配器; 类别 source; 类型 core-logic; 符号 _CapacityField, _float_open_short_side, _relieve_for_alloc, FloatMultiEndedAllocator): 分配器核心: 链式邻居、FloatMultiEndedAllocator、统一短供 ladder、_CapacityField 容量 memo。所有 unified pool 分配路径都经过这里。
- python/sglang/srt/mem_cache/unified_memory_pool.py (模块 内存池; 类别 source; 类型 core-logic; 符号 UnifiedKVPool, SubPoolSpec, init_unified_mamba_swa_pools, move_kv_cache): UnifiedKVPool 从恰好 2 子池泛化为 N 子池并支持 float 中间池; 新增 init_unified_mamba_swa_pools 三池工厂与 move_kv_cache 统一移动契约。
- python/sglang/srt/mem_cache/kv_cache_configurator.py (模块 配置器; 类别 source; 类型 dependency-wiring; 符号 _init_unified_mamba_swa_pools): 路由 gate 新增 tri-pool 分支: mamba + hybrid-SWA 且非 DSV4 的模型优先走 _init_unified_mamba_swa_pools, 修复了 Inkling 类模型误入 2-pool mamba 分支导致 SWA KV 按 full lifetime 存储的问题。
- test/registered/models_e2e/test_inkling_unified.py (模块 端到端测试; 类别 test; 类型 test-coverage; 符号 TestInklingUnifiedTriPool, TestInklingUnifiedVsStaticParity, _unified_args, _static_args): Inkling 三池 e2e 测试: 启动即路由闸门, 覆盖 logprob 一致性、prefix 复用、SWA 滑窗越过窗口、可选 static-vs-unified parity lane。
- test/registered/unit/mem_cache/test_unified_tri_pool.py (模块 三池单测; 类别 test; 类型 test-coverage; 符号 TestUnifiedTriPool, _tri_specs, _FakeKVCache, _FakeUnifiedSWAKVPool): 三池复合分配器的契约测试: 链式接线、联合 available_size

可行性、SWA 洞原地零拷贝复用、边界吸收、mamba slot 分配器独立性。

- test/registered/unit/mem_cache/test_unified_capacity_memo.py (模块 容量测试; 类别 test; 类型 test-coverage; 符号 TestCapacityMemoCoherence, TestTriCapacityMemoCoherence) : 容量 memo 一致性测试: 每种变异后的 memo 新鲜性、随机操作序列属性测试、绕过描述符的写入被 idle 检查捕获、float span 移动使所有邻居 memo 失效。
- python/sglang/srt/managers/scheduler_components/invariant_checker.py (模块 调度校验; 类别 source; 类型 dependency-wiring; 符号 _check_full_pool, _check_swa_pool) : 调度器不变式检查对三池复合改用 conserve 容量视图, 避免浮动边界导致的 full/swa 可用性误判。

关键符号: _CapacityField, _float_open_short_side, _relieve_for_alloc, FloatMultiEndedAllocator.make_room, FloatMultiEndedAllocator.compact_holes, UnifiedMambaSWATokenToKVPoolAllocator, init_unified_mamba_swa_pools, _init_unified_mamba_swa_pools, move_kv_cache, conserve_full_available_size, conserve_swa_available_size, _is_frontier_transparent

关键源码片段

python/sglang/srt/mem_cache/unified_memory_pool.py

UnifiedKVPool 从恰好 2 子池泛化为 N 子池并支持 float 中间池; 新增 init_unified_mamba_swa_pools 三池工厂与 move_kv_cache 统一移动契约。

```
class UnifiedKVPool:
```

```
    """一个 uint8 字节缓冲共享给 N 个子池: 两个 END 池 (grow-up / grow-down)
    各占一端向内生长, 可选若干个 float 中间池活在两端 frontier 之间。
    分配器保证字节区间不相交; 这里不做使用追踪。
    """
```

```
    def __init__(self, *, total_bytes, sub_pool_specs, device,
                  enable_memory_saver, page_size=1):
        assert page_size >= 1, f"page_size must be >= 1; got {page_size}"
        assert len(sub_pool_specs) >= 2, (
            f"UnifiedKVPool needs >= 2 sub-pools; got {len(sub_pool_specs)}"
        )
        names = [s.name for s in sub_pool_specs]
        assert len(set(names)) == len(names), (
            f"sub-pool names must be unique; got {names}"
        )
```

```
    # 方向合法性已在每个 spec 的 __post_init__ 中校验过;
    # 这里只保证链结构: 恰好一个 up 端 + 一个 down 端, float 任意多个。
    up_specs = [s for s in sub_pool_specs if s.grow_direction == "up"]
    down_specs = [s for s in sub_pool_specs if s.grow_direction == "down"]
    float_specs = [s for s in sub_pool_specs if s.grow_direction == "float"]
    assert len(up_specs) == 1 and len(down_specs) == 1, (
        f"UnifiedKVPool needs exactly one grow-up and one grow-down END "
```

```

        f"sub-pool; got directions {[s.grow_direction for s in sub_pool_specs]}"
    )

    # 规范链序（低字节端 -> 高）：grow-up END, float 中间池（保持输入顺序），
    # grow-down END。分配器邻居接线按此顺序；其余访问按名字，输入顺序无关。
    self.sub_pool_specs: List[SubPoolSpec] = [
        up_specs[0],
        *float_specs,
        down_specs[0],
    ]
    self._page_size = page_size
    self._specs_by_name: Dict[str, SubPoolSpec] = {
        s.name: s for s in sub_pool_specs
    }
    # ... 视图构建与 reserved-floor 计算随后按 self.sub_pool_specs 统一进行，
    # float 的 page-0 dummy-write 信封也计入预留，保证 mamba 等 page_size=1
    # 的子池只按单 slot 计费。

```

评论区精华

PR 页面公开 review 评论（review_comments_count = 0），核心讨论全部沉淀在提交历史与 ch-wan 的 CI 操作中。最有价值的交锋是 commit ae178552 的 "Adversarial review"：ch-wan 对本 PR 与该系列早期重构做了对抗性审查，发现 6 个缺陷（最严重的是联合容量按 raw bytes 定价 float 扩展，而 `take_physical_pages` 只能整页消费，导致 boot 后首次 alloc 触发 `alloc_with_virtual` 断言），每个修复都带一个回滚即失败的测试。其次是路由顺序问题：mambaish 分支先检查会吞掉 Inkling 类模型，使其 SWA KV 按 full lifetime 存储、丢失窗口回收。此外还暴露了测试自身缺陷——logprob 断言因参数顺序错误从未真正执行，以及 move-gate 测试 stub 在链式化后腐烂。

- mambaish 路由优先级导致 Inkling 误入 2-pool 分支 (correctness): 新增 tri-pool 分支放在最前：mambaish && is_hybrid_swa && !dsv4 优先路由，并显式拒绝 PD 组合（无 SWA 子池的 whole-envelope 传输方案）。
- Adversarial review 发现的 6 个缺陷 (correctness): 6 个缺陷逐一修复，每个修复附带回滚即失败的测试；包括 region bounds 页取整、chain walk 计数、capacity memo 覆盖等。
- 容量 memo 的失效机制设计 (design): 采用 epoch memo；约束是容量字段只能 rebind、不能原地 mutate，未来新增变异点由测试兜底。
- SWA per-step free 移除 host sync (performance): 采用 start_pos 路径；不可对齐时用 _page_reps_pieces 补第二段，保持无同步；与 dedup fallback 的端状态一致性有测试 pin 住。
- logprob 测试断言从未执行 (testing): 修复调用签名为 `assert_logprobs_match(self.base_url, {model: {"kl_div": 1e-2}}, self.model)`，测试真正生效。
- 注释 ASCII 规范化 (style): 全部新增注释转 ASCII。

风险与影响

- 风险:

1. 核心路径回归风险: multi_ended_allocator.py 改动 +1406/-130, 覆盖所有 unified pool 分配路径。链式 walk 若在 2 池下与旧封闭式公式产生偏差, 会整体改变容量判断; test_multi_ended_allocator.py 用 golden identity 钉住 2 池数学, 但边界组合 (lazy/eager、分组 free、跨池搬迁) 仍多。
2. 容量 memo 静默失效: epoch 机制依赖 " 字段只能 rebind 不能原地 mutate " 的契约, 未来新增变异点若绕过 _CapacityField, 过 / 欠准入不会崩溃、只会静默; 测试用 verify_byte_accounting 兜底, 但这是运行时校验而非编译期保证。
3. float 数据搬迁: make_room/compact_holes 涉及 live 页重定位 + v2p/p2v 重建 + KV 数据拷贝, 正确性依赖 inverse_history 与 pending_reuse 的事件同步; 6 个已修复缺陷表明此类边界条件密集。
4. 现有 2 池 SWA 行为变化: per-step free_swa 改为 start_pos 算术派生 page 代表, 影响所有 SWA 复合 (即使非 tri), 改动虽小但在 decode 热路径。
5. e2e 依赖外部 checkpoint (thinkingmachines/Inkling test revision), CI 稳定性受上游可用性影响; parity lane 需 INKLING_UNIFIED_PARITY=1 才运行, 回归探测有滞后。
6. PD 组合被显式拒绝: hybrid-SWA 三池与 disaggregation 同时使用会抛 ValueError, 功能面收窄是有意为之。 - 影响: 用户与模型面: Inkling 首次获得 unified-memory 支持 (此前直接 boot 崩溃); 2 池模型 (Falcon-H1、Qwen3.5、Kimi-Linear、gpt-oss 等) 布局 byte-identical, GSM8K 精度中位差在 $\pm 0.75pp$ 内、速度中位 ITL 变化在 $-2.85\% \sim +2.98\%$ 内, 基本无回归。系统面: 调度器容量语义从 byte 协调视图切换到 slot-conservation 视图 (conserve_full_available_size/conserve_swa_available_size), invariant checker 与 pool stats observer 同步调整; SWA per-step free 去掉 host sync, decode 路径少一个同步点。团队面: mem_cache 成为 N 池链式架构, 后续新子池 (如 spec-decode scratch band 已预留 float-only 收窄) 有明确扩展点; 测试方法论 (AST 扫描 tombstone、随机操作属性测试、e2e parity lane) 可复用。 - 风险标记: 核心路径变更, 容量一致性风险, 跨池数据搬迁, 外部模型依赖, 调度器语义变更

关联脉络

- PR #35154 (stacked base, see #35177 body): PR body 明确说明本 PR 构建于 #35154 之上, 应在其后 review; 本 PR 的 diff 包含该 PR 的内容。
- PR #35158 (stacked base, see #35177 body): PR body 明确说明本 PR 构建于 #35158 之上, 应在其后 review; 本 PR 的 diff 包含该 PR 的内容。
- PR #37167 [mem_cache] Make release, row-reuse asserts, and presence checks read the KV record: 同域 mem_cache 演进: KV 释放 / 复用 / 存在性检查统一走 KV record, 与本 PR 的链式分配器共同收敛 unified pool 语义。
- PR #37170 [unified-memory] Drop the vacated 'dense' qualifier and the restating comments: unified-memory 术语清理: 去掉废弃的 dense 限定词, 与三池浮动脉络同属 unified-memory 重构线。
- PR #34613 feat(unified-memory): read unified pool from attention backends fa3/flashinfer/trtllm_mha/flashmla: unified pool 读路径迁移到全部注意力后端, 与本 PR 的写 / 容量侧改造互为补充。

- PR #35245 refactor(unified-memory): translate the KV write location once, at ForwardBatch construction: KV write location 在 ForwardBatch 构造时一次性翻译，同一 unified-memory 架构演进线。