

PR #32373 完整报告

sgl-project/sglang

Fix `--hcache-size` allocating ~2x host memory on hybrid SWA

合并时间: 2026-07-26 08:19

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/32373>

执行摘要

- 一句话: 修复 hybrid SWA 下 `hcache-size` 重复分配宿主内存
- 推荐动作: 建议精读。此 PR 修复了一个重要的内存分配 bug, 逻辑清晰, 测试充分, 设计简洁。值得关注的设计决策是将 `--hcache-size` 定义为总预算并按设备字节比例拆分, 而非分别传递固定值, 这与其他相似场景 (如 `ratio` 模式) 的设计保持一致。

功能与动机

在 hybrid SWA 场景中, 存在两个宿主 KV 池 (full-attention 和 SWA), 固定大小的 `--hcache-size N` 被独立传递给每个池, 导致每个池都分配约 N GB 的宿主内存, 总分配量接近 2N。此 bug 导致宿主内存浪费, 违反了 `--hcache-size` 作为总预算的语义。

实现拆解

1. 新增 `_split_hcache_size` 函数: 位于 `python/sglang/srt/mem_cache/hybrid_cache/hybrid_pool_assembler.py`, 该函数接收 `hcache_size` (总预算) 和 `kv_pools` 元组, 遍历每个 `pool` 调用 `get_kv_size_bytes()` 获取设备池字节数 (支持标量和 (k, v) 元组两种返回格式), 按比例计算每个池应分配的宿主内存大小, 返回浮点数元组。
2. 修改 `build_kv_host_pool` 函数: 增加可选的 `host_size: Optional[float] = None` 参数, 当 `host_size` 不为 `None` 时, 将其作为 `hcache_size` 传递给宿主池构造函数, 取代原始的 `server_args.hcache_size`。
3. 修改 `build_hybrid_swa_stack` 函数: 在创建两个 `build_kv_host_pool` 调用之前, 判断 `server_args.hcache_size > 0`, 若是则调用 `_split_hcache_size` 计算出 `kv_host_size` 和 `swa_host_size`, 分别传递给两个宿主池。
4. 新增单元测试: 新增文件 `test/registered/unit/mem_cache/test_hybrid_pool_assembler.py`, 使用模拟池验证拆分逻辑: 设备字节比例为 75:25 时, 拆分结果为 (75.0, 25.0), 总和为 100, 确保总预算未被放大。

关键文件:

- `python/sglang/srt/mem_cache/hybrid_cache/hybrid_pool_assembler.py` (模块 缓存组装器; 类别 `source`; 类型 `core-logic`; 符号 `_split_hcache_size`, `build_kv_host_pool`, `build_hybrid_swa_stack`): 核心修改文件, 新增 `_split_hcache_size` 函数用于按比例拆分总预算, 修改 `build_kv_host_pool` 支持可选 `host_size` 参数, 修改 `build_hybrid_swa_stack` 调用拆分逻辑。

- test/registered/unit/mem_cache/test_hybrid_pool_assembler.py (模块测试; 类别 test; 类型 test-coverage; 符号 _Pool, init, get_kv_size_bytes, TestSplitHicacheSize): 新增单元测试文件, 验证 _split_hicache_size 按设备字节比例正确拆分总预算, 且总和不变。使用模拟 pool 类支持标量和元组两种返回格式。

关键符号: _split_hicache_size, build_kv_host_pool, build_hybrid_swa_stack

关键源码片段

python/sclang/srt/mem_cache/hybrid_cache/hybrid_pool_assembler.py

核心修改文件, 新增 _split_hicache_size 函数用于按比例拆分总预算, 修改 build_kv_host_pool 支持可选 host_size 参数, 修改 build_hybrid_swa_stack 调用拆分逻辑。

```
def _split_hicache_size(    hicache_size: int, kv_pools: tuple[Any, ...] ) -> tuple[float, .
.]:    # 遍历每个设备池, 获取其 KV 缓存字节数    device_pool_sizes = []    for kv_pool
in kv_pools:        size_bytes = kv_pool.get_kv_size_bytes()        # 支持两种返回格式: 标
量 (int) 或 (k, v) 元组        device_pool_sizes.append(            sum(size_bytes) if
isinstance(size_bytes, tuple) else size_bytes        )    total_device_pool_size =
sum(device_pool_sizes)    # 按比例拆分总预算, 返回与输入 pools 顺序对应的份额元组
return tuple(        hicache_size * size_bytes / total_device_pool_size        for
size_bytes in device_pool_sizes    )

def build_kv_host_pool(    *,    kv_pool: Any,
page_size: int,    server_args: ServerArgs,    use_mla: bool,
override_kv_cache_dim: Optional[int] = None,    host_size: Optional[float] = None, # 新
增: 若提供则替代 server_args.hicache_size ):    kv_host_pool_cls = (
MLATokenToKVPoolHost if use_mla else get_mha_host_pool_cls(kv_pool)    )    kwargs
= {}    if override_kv_cache_dim is not None:        kwargs["override_kv_cache_dim"] =
override_kv_cache_dim    return kv_host_pool_cls(        kv_pool,
server_args.hicache_ratio,        server_args.hicache_size if host_size is None else
host_size, # 关键: 使用传入的 host_size        page_size,
server_args.hicache_mem_layout,        allocator_type=_get_allocator_type(server_args),
**kwargs,    )
```

test/registered/unit/mem_cache/test_hybrid_pool_assembler.py

新增单元测试文件, 验证 _split_hicache_size 按设备字节比例正确拆分总预算, 且总和不变。使用模拟 pool 类支持标量和元组两种返回格式。

```
class _Pool:
    def __init__(self, kv_bytes):
        self._kv_bytes = kv_bytes # 支持标量或 (k, v) 元组

    def get_kv_size_bytes(self):
        return self._kv_bytes

class TestSplitHicacheSize(CustomTestCase):
    def test_splits_total_budget_by_device_bytes(self):
        # 模拟两个设备池: 第一个返回标量 75GB, 第二个返回 (k, v) 元组共 25GB
        shares = _split_hicache_size(
```

```
100, (_Pool(75 * 10**9), _Pool((15 * 10**9, 10 * 10**9)))
)
# 验证比例: 75% 和 25%
self.assertEqual(shares, (75.0, 25.0))
# 验证总预算不变, 未翻倍
self.assertEqual(sum(shares), 100)
```

评论区精华

代码审查由 [xiezhq-hermann](#) 批准, 无未解决的问题。PR body 已清晰说明问题根因 (固定大小被独立传递给两个池) 和修复方案 (按比例拆分), 审查者无额外评论。

- 暂无高价值评论线程

风险与影响

- 风险: 风险较低。变更集中在 `hybrid_pool_assembler.py` 中的两个函数和一个新增辅助函数, 逻辑简单且通过单元测试覆盖。非 hybrid SWA 场景 (如单池或 ratio 模式) 不受影响, 因为 `build_hybrid_swa_stack` 仅在 SWA 混合场景中调用, 且仅当 `hicache_size > 0` 时才触发拆分。回归风险限于 hybrid SWA 初始化路径。
- 影响: 影响范围仅限于 hybrid SWA 场景中使用固定 `--hicache-size` 的用户。对于受影响用户, 此修复将宿主内存占用从约 2N 降为 N, 节省内存资源。ratio 模式 (`--hicache-ratio`) 不受影响, 单池场景也不受影响。测试覆盖了核心拆分逻辑, 但缺少集成测试验证真实环境中的内存分配。
- 风险标记: 核心路径变更, 缺少集成测试

关联脉络

- PR #29901 Radix Cache Split: Spin off TreeCore: 涉及缓存架构重构, 与本 PR 的宿主池拆分逻辑相关
- PR #31181 [Hicache][1/2]Support Mamba branching in Unified Radix Cache with HiCache: 扩展 HiCache 功能, 本 PR 修复其中 hybrid SWA 场景的内存分配 bug