

PR #27293 完整报告

sgl-project/sglang

[HiCache][Dsv4] Don't cache C128 State pool in L3

合并时间: 2026-06-08 21:23

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/27293>

执行摘要

- 一句话: HiCache 不再缓存 C128 State 池
- 推荐动作: 该 PR 值得精读, 展示了如何分析缓存系统中无效的池注册问题, 并进行针对性优化。设计决策 (基于页大小整除性判断 State 是否可被读取) 值得类似场景借鉴。

功能与动机

C128 State 池被添加到 HiCache 的池列表中, 但由于页大小 256 能被 128 整除, 缓存命中边界上尾部实际为空, 缓存命中时永远不会被消耗。同时槽位仍然被分配, 导致 L3 缓存中每百万 token 约 0.54GB 的开销。

实现拆解

1. 识别无效池条目: 在 `python/sglang/srt/mem_cache/hybrid_cache/hybrid_pool_assembler.py` 的 `build_deepseek_v4_hicache_stack` 函数中, 发现 `DEEPSEEK_V4_C128_STATE` 池的注册是冗余的, 因为其 State 大小 (128) 整除页大小 (256), 导致缓存命中时 State 数据永远不会被读取。
2. 移除 State 池创建和注册: 删除了创建 `c128_state_host_pool` 和 `build_pool_entry` 中对应的 `DEEPSEEK_V4_C128_STATE` 条目, 仅保留 `DEEPSEEK_V4_C128 (KV)` 池的注册。
3. 添加注释说明: 在移除位置添加注释, 解释不再注册 C128 State 池的原因, 便于后续维护。
4. 验证: 通过 AIME25、GPQA、MMLU 等精度测试确认移除后模型精度无退化, 同时 CI 测试全部通过。

关键文件:

- `python/sglang/srt/mem_cache/hybrid_cache/hybrid_pool_assembler.py` (模块 HiCache ; 类别 source; 类型 core-logic) : 核心变更文件, 移除了 C128 State 池在 HiCache 中的注册, 减少 L3 缓存无效占用。

关键符号: 未识别

关键源码片段

`python/sglang/srt/mem_cache/hybrid_cache/hybrid_pool_assembler.py`

核心变更文件, 移除了 C128 State 池在 HiCache 中的注册, 减少 L3 缓存无效占用。

```
# python/sglang/srt/mem_cache/hybrid_cache/hybrid_pool_assembler.py#L432-L454
```

```

# 修改前: 创建 c128_state_host_pool 并注册到 HiCache
# 修改后: 仅保留 C128 (KV) 池, State 池不再注册
if c128_layer_mapping:
    c128_host_pool = DeepSeekV4PagedHostPool(
        pool_name=str(PoolName.DEEPSEEK_V4_C128),
        device_buffers=kvcache.c128_kv_pool.kv_buffer,
        item_bytes=kvcache.c128_kv_pool.bytes_per_page_padded,
        num_host_pages=num_host_pages,
        slot_page_size=page_size,
        layout=server_args.hicache_mem_layout,
        allocator_type=server_args.hicache_storage_backend,
    )
# 注释说明: C128 state pool 有意不注册到 HiCache。
# 因为 page_size=256 % 128 == 0, state 在加载时不会被消耗。
entries.extend([
    build_pool_entry(
        name=PoolName.DEEPSEEK_V4_C128,
        host_pool=c128_host_pool,
        device_pool=kvcache.c128_kv_pool,
        layer_mapping=c128_layer_mapping,
        transfer_layer_num=transfer_layer_num,
    ),
    # 移除了之前注册 DEEPSEEK_V4_C128_STATE 的 build_pool_entry 调用
])

```

评论区精华

审核者 hzh0425 和 ispobock 均认可该变更, ispobock 特别感谢了作者的发现和测试结果。讨论中作者详细解释了 C128 State 池不会被读取的技术原因: State 缓冲区在 `c128_forward` 中通过 `kv_buf` 读取, 但仅在 prefill 期间且 plan 有效时才调用 `c128_forward`, 而 plan 的填充逻辑决定了 State 数据不会被有效使用。此外, 作者提供了多组 AIME25、GPQA、MMLU 的冷 / 热 L3 缓存精度对比, 结果均在正常波动范围内, 证明移除后精度无损。

- 确认修改合理性 (design): ispobock 认可修改, 并感谢作者的发现和测试结果。
- 精度测试验证 (testing): 精度无损, 验证了修改的安全性。
- 技术原理说明 (design): 确认移除是安全的。

风险与影响

- 风险: 风险较低。主要风险是移除 State 池后, 如果未来某处代码路径开始读取 C128 State 且依赖 HiCache 预加载, 会导致功能缺失。但当前代码中 State 池的读取路径已被证明在缓存命中时永远不会被调用, 且移除后 CI 和精度测试均通过。
- 影响: 对用户: HiCache 启用且使用 DeepSeek-V4 模型时, L3 缓存占用减少约 0.54GB/1M tokens, 可间接提升其他数据的缓存效率。对系统: 移除无效的 L3 分配逻辑, 减少不必要的内存带宽和存储空间消耗。对团队: 代码简化, 移除约 18 行冗余代码。
- 风险标记: 核心路径变更, 缺少测试覆盖

关联脉络

- PR #27489 Fix TP deadlock in unified radix cache writing_check / loading_check: 同为 HiCache 相关修复，涉及 unified radix cache 的正确性。
- PR #27554 [UnifiedTree]: Support hicache metrics: 同为 HiCache 相关特性，添加指标支持。