

PR #33598 完整报告

sgl-project/sglang

Clarify post-capture KV reservation logs

合并时间: 2026-08-05 06:06

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/33598>

执行摘要

- 一句话: 澄清 post-capture KV 预留日志, 区分 VA 上界与物理分配
- 推荐动作: 不值得精读, 属于低风险日志澄清; 关注点在于 post-capture KV 定容与虚拟地址预留的语义区分。可以作为观察 mem_cache 模块日志演进的参考。

功能与动机

PR body 明确指出: “The pre-capture VMM size is a virtual upper bound, but its startup log looks like a physical allocation.” 也就是 pre-capture 阶段 VMM 只预留虚拟地址空间, 物理内存尚未全部提交, 原有日志会让用户误以为容量已物理分配, 影响显存占用判断。

实现拆解

1. 通用 KV 池日志 (python/sglang/srt/mem_cache/memory_pool.py) : 在 `_finalize_allocation_log` 中根据 `self.post_capture_active` 条件切换措辞, 为 True 时输出 “VA upper bound”, 否则保持 “is allocated”; 该方法同时覆盖 K/V 分离与单 KV size 两条路径。
2. VMM arena 启动日志 (python/sglang/srt/mem_cache/kv_vmm_backing.py) : 将 KvVmmArena 就绪日志的 `reserved=` 字段改名为 `reserved_va=`, 提示该数值来自 `cuMemAddressReserve`, 是虚拟地址预留而非物理提交。
3. SWA 池日志 (python/sglang/srt/mem_cache/swa_memory_pool.py) : SWAKVPool 初始化日志由固定 “mem usage” 改为条件显示 “VA upper bound” 或 “mem usage”, 复用 `post_capture_active` property 判断。

测试与配置配套: PR body 声明为 log-only 变更, 未新增测试; 仅通过 `py_compile` 与 pre-commit hooks 校验语法与格式, 无部署或 schema 变更。

关键文件:

- python/sglang/srt/mem_cache/memory_pool.py (模块 缓存池; 类别 source; 类型 logging; 符号 `_finalize_allocation_log`) : 在统一日志入口 `_finalize_allocation_log` 中按 `post_capture_active` 区分 VA upper bound 与 is allocated, 是本次日志澄清的核心。
- python/sglang/srt/mem_cache/kv_vmm_backing.py (模块 缓存池; 类别 source; 类型 logging; 符号 `KvVmmArena.init`) : KvVmmArena 就绪日志将 `reserved` 改名为 `reserved_va`, 明确虚拟地址预留语义, 是本 PR 的另一关键日志点。

- python/sglang/srt/mem_cache/swa_memory_pool.py (模块 缓存池; 类别 source; 类型 logging; 符号 SWAKVPool.init) : SWAKVPool 初始化日志区分 VA upper bound 与 mem usage, 与 memory_pool.py 的修正保持一致。

关键符号: _finalize_allocation_log, KvVmmArena.init, SWAKVPool.init

关键源码片段

python/sglang/srt/mem_cache/memory_pool.py

在统一日志入口 _finalize_allocation_log 中按 post_capture_active 区分 VA upper bound 与 is allocated, 是本次日志澄清的核心。

```
# memory_pool.py —— KV cache 分配结束后的统一日志入口
# 在 post-capture 模式下, KV 池只保留了虚拟地址 (VA) 上界,
# 物理内存尚未按最终大小提交, 因此日志需要区分“VA upper bound”与“is allocated”。
def _finalize_allocation_log(self, num_tokens: int):
    cache_name = (
        f"{self.allocation_label} KV Cache"
        if self.allocation_label is not None
        else "KV Cache"
    )
    kv_size_bytes = self.get_kv_size_bytes()
    if isinstance(kv_size_bytes, tuple):
        k_size, v_size = kv_size_bytes
        k_size_GB = k_size / GB
        v_size_GB = v_size / GB
        logger.info(
            f"{cache_name} {'VA upper bound' if self.post_capture_active else 'is allocated'}. dtype: {self.dtype}, "
            f"#tokens: {num_tokens}, K size: {k_size_GB:.2f} GB, "
            f"V size: {v_size_GB:.2f} GB"
        )
        self.mem_usage = k_size_GB + v_size_GB
    else:
        kv_size_GB = kv_size_bytes / GB
        logger.info(
            f"{cache_name} {'VA upper bound' if self.post_capture_active else 'is allocated'}. dtype: {self.dtype}, "
            f"#tokens: {num_tokens}, KV size: {kv_size_GB:.2f} GB"
        )
        self.mem_usage = kv_size_GB
```

python/sglang/srt/mem_cache/kv_vmm_backing.py

KvVmmArena 就绪日志将 reserved 改名为 reserved_va, 明确虚拟地址预留语义, 是本 PR 的另一关键日志点。

```
# kv_vmm_backing.py —— KvVmmArena 虚拟地址预留与就绪日志
# cuMemAddressReserve 只保留地址空间, 物理内存按需提交,
# 因此日志用 reserved_va 显式标注这是 VA 上界而非物理分配。
```

```

self.reserved = self._align(reserve_bytes)
self.base = int(
    _check(
        drv.cuMemAddressReserve(self.reserved, self.granularity, 0, 0),
        "cuMemAddressReserve",
    )
)
# commit_range 记账结构: 记录映射 VA -> (大小, 句柄) 与各偏移已提交字节。
self._ranges = {}
self._committed_by_offset = {}
self._range_backed = 0
self._closed = False

logger.info(
    "KvVmmArena[%s] ready: device=%d reserved_va=%.1f GiB "
    "granularity=%d KiB",
    self._sfx,
    self.device_id,
    self.reserved / (1024**3),
    self.granularity // 1024,
)

```

评论区精华

本 PR 没有 review 评论或技术讨论线程，唯一一条 discussion 是作者触发的 [/tag-and-rerun-ci](#) CI 重跑指令，不涉及实现层面的交锋。

- 暂无高价值评论线程

风险与影响

- 风险：风险极低：仅日志文本变化，可能影响依赖日志解析的监控或自动化脚本；
post_capture_active 在 _finalize_allocation_log 调用时需与分配路径状态一致，否则可能出现措辞与实际不符（该状态由内存池统一维护，回归风险低）。本次未新增测试，属于 log-only 变更。
- 影响：影响范围仅限启动与分配阶段的 INFO 日志；不改变 KV 缓存分配、内存占用、API 或性能，对普通用户基本无感，对排查显存占用问题有正向帮助。
- 风险标记：日志格式变更可能影响监控解析，缺少测试覆盖

关联脉络

- PR #33445 [mem_cache] Label HiCache host pools and clarify post-capture KV sizing logs: 同属 mem_cache 模块的 post-capture KV 日志澄清与标签改进，是本次改动的直接延续。
- PR #33427 Enable post-capture KV sizing with DP attention: 引入了 post-capture KV 定容能力，本次 PR 正是为该模式下的日志措辞修正。

- PR #33375 [Observability] Add startup, memory, and hybrid SWA diagnostics: 新增内存与 SWA 诊断日志，本次 SWA 池日志措辞与其观测目标一致。