PR #23315 完整报告

sgl-project/sglang

Opt-in strip of thinking tokens from radix cache

合并时间：2026-04-21 13:59

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/23315

执行摘要

- 一句话：为推理模型添加可选的基数树缓存思考令牌剥离功能，以节省 GPU 内存。
- 推荐动作：建议精读此 PR，特别关注 _cache_commit_len() 的设计决策和 opt-in 策略，它展示了如何在最小化变更下处理推理模型特有的缓存问题，代码改动集中且测试全面，是学习缓存优化和向后兼容性权衡的好例子。

功能与动机

根据 PR body 和关联 Issue #22373，推理模型（如 DeepSeek-R1、QwQ 等）在请求完成时将所有输出令牌（包括思考令牌）插入基数树缓存，但由于客户端在多轮提示中丢弃思考令牌（遵循 DeepSeek/OpenAI 约定），这些条目变成死分支，浪费 GPU 内存（例如每个分支约 1.3-1.6 GB）。答案令牌也因 RoPE 位置偏移而无法安全重用，因此剥离两者可以避免内存浪费和缓存污染。

实现拆解

1. 添加配置标志：在 python/sglang/srt/server_args.py 的 ServerArgs 类中添加 strip_thinking_cache: bool 字段（默认 False）和 --strip-thinking-cacheCLI 参数，作为功能的入口开关。
2. 核心逻辑修改：在 python/sglang/srt/managers/schedule_batch.py 的 Req 类中新增 _cache_commit_len() 方法，当 strip_thinking_cache 启用且 reasoning_tokens > 0 时，返回 min(self.kv_committed_len, len(self.origin_input_ids))，仅将提示前缀视为已提交缓存长度；并修改 pop_committed_kv_cache() 和 pop_overallocated_kv_cache() 方法调用此辅助函数，使得思考令牌和答案令牌落入过度分配范围。
3. 断言放宽：在 python/sglang/srt/mem_cache/common.py 的 release_kv_cache() 函数中，将 start_p == end_p 的断言条件从 spec_algo is None 修改为 spec_algo is None and not global_server_args.strip_thinking_cache，以允许 strip 模式下的过度分配，避免崩溃。
4. 测试配套：在 test/registered/unit/mem_cache/test_unified_radix_cache_unittest.py 中添加 test_cache_finished_req_strips_thinking() 测试函数，参数化覆盖不同缓存类型（FULL/SWA/MAMBA）和页面大小，验证 strip 功能下仅提示前缀被缓存且思考令牌不被匹配。

关键文件：

- python/sglang/srt/managers/schedule_batch.py（模块 请求管理；类别 source；类型 core-logic；符号 _cache_commit_len）：这是核心逻辑文件，新增 _cache_commit_len() 方法并修改缓存提交函数，直接控制 strip 功能的行为。
- python/sglang/srt/server_args.py（模块 配置参数；类别 source；类型 configuration）：添加功能配置入口，定义 strip_thinking_cache 字段和 CLI 参数，使功能可选择性启用。
- python/sglang/srt/mem_cache/common.py（模块 缓存通用；类别 source；类型 core-logic）：修改 release_kv_cache 函数中的断言逻辑，允许 strip 模式下的过度分配，避免因 strip 导致的崩溃。
- test/registered/unit/mem_cache/test_unified_radix_cache_unittest.py（模块 单元测试；类别 test；类型 test-coverage；符号 test_cache_finished_req_strips_thinking）：添加单元测试验证 strip 功能，覆盖不同缓存配置和页面大小，确保正确性。

关键符号：_cache_commit_len, pop_committed_kv_cache, pop_overallocated_kv_cache

关键源码片段

python/sglang/srt/managers/schedule_batch.py

这是核心逻辑文件，新增 _cache_commit_len() 方法并修改缓存提交函数，直接控制 strip 功能的行为。

```
def _cache_commit_len(self) -> int:
    # 报告仅提示前缀，使得思考令牌和答案令牌落入过度分配范围，
    # 并通过 release_kv_cache 回收。这是为了修复 Issue #22373。
    if get_global_server_args().strip_thinking_cache and self.reasoning_tokens > 0:
        # 当 strip 启用且存在推理令牌时，仅返回提示前缀长度与已提交长度的最小值，
        # 确保思考令牌和答案令牌被视为过度分配。
        return min(self.kv_committed_len, len(self.origin_input_ids))
    # 默认情况下返回完整的已提交缓存长度，保持向后兼容性。
    return self.kv_committed_len
```

评论区精华

无 review 评论，因此无讨论亮点。

- 暂无高价值评论线程

风险与影响

- 风险：
  - 回归风险：修改了 pop_committed_kv_cache 和 pop_overallocated_kv_cache 的核心路径，若 _cache_commit_len() 逻辑错误，可能导致缓存释放不当或内存泄漏。
  - 兼容性风险：功能为 opt-in 且默认关闭，不影响现有用户；但启用后可能改变缓存行为，需用户明确知晓。
  - 性能影响：剥离思考令牌可以减少死分支，提升缓存命中率和内存利用率，但对推理模型的多轮对话性能有正面影响。

    - 断言放宽：在 common.py 中跳过 start_p == end_p 断言可能掩盖其他非 strip 引起的过度分配问题，需依赖测试覆盖确保正确性。
- 影响：
    - 用户影响：提供可选配置，用户可启用以节省 GPU 内存（尤其是高并发推理场景），但需权衡缓存复用可能的变化。
    - 系统影响：减少基数树缓存中的死分支，降低内存压力，提升缓存效率，可能改善长对话性能。
    - 团队影响：引入新配置参数和缓存逻辑，需文档说明；测试配套完善，便于后续维护。
    - 风险标记：核心路径变更，缓存逻辑调整，测试覆盖

关联脉络

- PR #23107 [Refactor] Replace page_align_keys helper with RadixKey.page_aligned method: 同样涉及基数树缓存重构，本 PR 的 strip 功能可能依赖或影响类似的缓存逻辑。
- PR #23243 [Hybrid-Cache]: Refactor hybrid_pool_assembler.py: 涉及混合缓存模块，与本 PR 的缓存剥离功能在缓存管理上有交叉关注点。