PR #22597 完整报告

sgl-project/sglang

Fix swa input length limitation

合并时间：2026-04-12 16:03

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/22597

执行摘要

- 一句话：修复 SWA 模型输入长度限制过严问题，允许大于 SWA 池但小于全池的输入长度。
- 推荐动作：该 PR 值得精读，特别是对于处理 SWA 模型或调度系统的工程师。值得关注的设计决策包括：1. 将 SWA 预算计算从简单的 min() 限制重构为显式偏移跟踪；2. _swa_budget_for_req 方法中考虑分块预填充和滑动窗口保留的逻辑；3. 保持向后兼容性的同时修复功能限制。

功能与动机

根据 PR body 描述，对于 SWA 模型，SWA 池通常远小于全池。之前系统会拒绝输入长度介于 SWA 池大小和全池大小之间的请求。然而，SWA 缓存在分块预填充后可以被驱逐，因此更大的输入长度应该被接受。该 PR 修复了预算计算以允许这种情况。PR body 中提供了具体示例：在修复前，输入长度 90000 令牌的请求会被拒绝（最大允许长度 19994 令牌）；修复后可以成功运行。

实现拆解

实现方案主要修改了调度策略模块的预算管理逻辑：1. 在 schedule_policy.py 中新增 rem_swa_token_offset 属性跟踪 SWA 令牌偏移；2. 新增 rem_swa_tokens 属性计算剩余 SWA 令牌数；3. 新增 _swa_budget_for_req 方法计算每个请求的 SWA 预算，考虑分块预填充和滑动窗口大小；4. 修改 rem_total_tokens 和 cur_rem_tokens 属性，移除对 SWA 池大小的 min() 限制；5. 在 budget_state 和 _update_prefill_budget 方法中添加 SWA 预算检查；6. 在 model_runner.py 中修复 max_token_pool_size 方法，使其返回全池大小而非 SWA 池和全池的最小值。

关键文件：

- python/sglang/srt/managers/schedule_policy.py（模块 scheduling）：核心调度策略文件，包含了 SWA 预算管理的主要逻辑变更，新增了 SWA 令牌偏移跟踪和预算计算方法。
- python/sglang/srt/model_executor/model_runner.py（模块 model_execution）：修复了 max_token_pool_size 方法的返回值，确保 SWA 模型返回全池大小而非受限的 SWA 池大小。

关键符号：rem_swa_tokens, _swa_budget_for_req, rem_total_tokens, cur_rem_tokens, budget_state, _update_prefill_budget, max_token_pool_size

评论区精华

review 讨论主要集中在代码可读性和维护性改进：1. gemini-code-assist[bot] 建议在 add_chunked_req 方法中使用更直接的 min() 逻辑来包含 SWA 令牌限制，提高可读性；2. 同一 reviewer 指出 add_one_req 方法中的 SWA 预算检查存在代码重复（第 805-808 行和第 821-824 行），建议提取为私有辅助方法以提高可维护性。作者在第二个 commit 中采纳了第一个建议的部分内容（更新了代码注释）。

- SWA 预算检查代码重复 (design): 建议被提出但未在提供的材料中看到采纳，可能因为 PR 已合并或作者选择保持现状。
- 剩余令牌计算逻辑优化 (design): 作者在第二个 commit 中更新了代码注释，但未完全采纳建议的逻辑重构。

风险与影响

- 风险：主要风险包括：1. 核心调度路径变更：修改了 schedule_policy.py 中的关键预算计算逻辑，可能影响所有 SWA 模型的请求调度；2. 边界条件处理：_swa_budget_for_req 方法中的 max(alloc, self.tree_cache.sliding_window_size) 逻辑需要确保滑动窗口大小预留足够；3. 兼容性风险：修改了 max_token_pool_size 方法的返回值，可能影响依赖此值的其他组件；4. 测试覆盖不足：从提供的材料看，没有明确提到新增的单元测试或验证。
- 影响：影响范围：1. 用户影响：SWA 模型用户现在可以提交输入长度大于 SWA 池但小于全池的请求，提高了系统可用性；2. 系统影响：修改了核心调度器的预算管理逻辑，影响所有使用 SWA 模型的请求调度；3. 团队影响：引入了更细粒度的 SWA 预算跟踪，提高了代码复杂度但增强了功能正确性。影响程度中等，主要针对特定 SWA 模型场景。
- 风险标记：核心路径变更，边界条件处理，缺少测试覆盖

关联脉络

- PR #21499 Add SWA support for runtime busy memory check: 同样涉及 SWA 模型支持，修改了 scheduler_runtime_checker_mixin.py，与本 PR 的 SWA 预算管理相关。
- PR #22562 [mem] Flatten memory checkers into composable per-pool invariant checks: 涉及内存检查器重构，与本 PR 的调度策略修改同属 observability 和 scheduling 模块。