

PR #27625 完整报告

sgl-project/sglang

Remove Req.extend_logprob_start_len field and make it pure

合并时间: 2026-06-25 09:06

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/27625>

执行摘要

- 一句话: 移除 Req.extend_logprob_start_len 字段, 改为纯自由函数
- 推荐动作: PR 的设计权衡分析极具学习价值, 建议涉及并发或重叠执行的工程师精读 PR 正文。合并后建议密切监控重叠调度场景下的日志概率正确性, 并可考虑为 compute_extend_logprob_start_len 添加单测。

功能与动机

原设计将 `extend_logprob_start_len` 作为 Req 的可变字段, 在重叠调度中该字段可能被下一轮迭代的 `set_extend_range` 覆盖, 而后续输出处理时仍会读取该字段, 导致使用了错误的相对位置。将字段改为自由函数并配合调用者快照, 可以从结构上消除这种时序错误, 使得显式读取过期值变得不可能。

实现拆解

1. 在 `schedule_batch.py` 中定义纯自由函数 `compute_extend_logprob_start_len`, 接收 `logprob_start_len`、`prefix_len`、`extend_len`、`full_untruncated_fill_len`, 返回相对起始位置, 替换原来的 `Req._recompute_extend_logprob_start_len` 方法。
2. 移除 `Req.__init__` 中对 `self.extend_logprob_start_len` 的初始化, 以及 `set_extend_range` 中对 `_recompute_extend_logprob_start_len` 的调用, 删除该私有方法。
3. 在 `scheduler.py` 的 `run_batch` 中, 将 `batch_result.extend_logprob_start_len_per_req` 赋值为 `batch.extend_logprob_start_lens` (快照), 而不是逐个从 `req` 读取字段。
4. 在 `decode_schedule_batch_mixin.py` 的 `prepare_for_prebuilt` 中, 移除对 `req.extend_logprob_start_len = 0` 的赋值以及从字段构建列表的代码, 直接设置 `self.extend_logprob_start_lens = None`, 因为 `decode` 批次不需要该值。
5. 调整测试 `test_prefill_adder.py`, 从 `mock req` 中删除已不存在的属性 `extend_logprob_start_len`。

关键文件:

- `python/sglang/srt/managers/schedule_batch.py` (模块 调度器; 类别 `source`; 类型 `core-logic`; 符号 `_recompute_extend_logprob_start_len`, `compute_extend_logprob_start_len`): 核心变更文件: 定义新的自由函数 `compute_extend_logprob_start_len`, 删除 `Req._recompute_extend_logprob_start_len` 方法和字段 `extend_logprob_start_len`, 更新 `set_extend_range` 和 `reset_for_retract` 等

使用点。

- python/sglang/srt/managers/scheduler.py (模块 调度器; 类别 source; 类型 core-logic) : 关键消费点修改: run_batch 中使用 batch.extend_logprob_start_lens 快照而非逐个读取 req.extend_logprob_start_len。
- python/sglang/srt/disaggregation/decode_schedule_batch_mixin.py (模块 解码批次; 类别 source; 类型 core-logic) : 移除对 req.extend_logprob_start_len 的赋值和基于字段的列表构建, 改为直接设置 extend_logprob_start_lens = None。
- test/registered/unit/managers/test_prefill_adder.py (模块 测试; 类别 test; 类型 test-coverage) : 从 mock req 中移除已不存在的属性 extend_logprob_start_len, 确保测试通过。

关键符号: compute_extend_logprob_start_len, _recompute_extend_logprob_start_len

评论区精华

作者在 PR 描述中详细对比了字段、属性、自由函数三种设计方案的优劣, 并解释为何属性是最糟选择 (静默错误)。特别讨论了 chunked prefill + overlap 场景下的时序问题, 以及 encoder 调整必须分层处理 (不能合并到自由函数中) 的原因。未收到外部 reviewer 实质性质疑, PR 被直接合并。

- 字段 vs 属性 vs 自由函数设计分析 (design): 选择自由函数 + 调用者快照方案, 并实现之。

风险与影响

- 风险: 新的自由函数必须与原 _recompute_extend_logprob_start_len 逻辑一致, 特别是 logprob_start_len == -1 的哨兵处理。所有消费方都已迁移到批次快照, 但仍需排查是否有遗漏的直接字段读取。prepare_for_prebuilt 将 extend_logprob_start_lens 设为 None 是正确的, 但可能存在其他路径错误地依赖该列表。测试仅删除了旧属性, 未新增对自由函数的单元测试, 回归风险中等。
- 影响: 对用户透明, 修复了重叠调度下日志概率计算可能使用错误位置的潜在 bug。系统内部数据流更清晰, 降低了因时序导致难排查 bug 的风险。该设计模式 (快照而非可变字段) 可在调度器中其他类似场景推广。团队需注意未来任何新的字段属性都应遵循类似模式。
- 风险标记: 核心路径变更, 缺少新增测试, 潜在字段读取遗漏

关联脉络

- PR #27616 Avoid dual semantics of extend_input_len by computing the candidate on the fly: 与 #27625 同属扩展字段生命周期清理系列, 都涉及 extend 相关语义的精确化。