

PR #27616 完整报告

sgl-project/sglang

Avoid dual semantics of extend_input_len by computing the candidate on the fly

合并时间: 2026-06-25 08:17

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/27616>

执行摘要

- 一句话: 消除 extend_input_len 的双重语义
- 推荐动作: 目标明确、改动精致、影响可控。建议精读 schedule_policy.py 中 add_dllm_staging_req 和 add_chunked_req 的即时计算模式, 作为消除隐式状态的好范例。关注其堆叠的 #27610 后续 PR。

功能与动机

PR 描述指出: `Req.extend_input_len` 在请求生命周期的不同阶段具有双重语义——准入前表示候选长度, 准入后表示已提交长度。这种重载是 #27610 中长度字段合并的前提障碍。解决双重语义可以使后续将 `extend_input_len` 和 `extend_fill_len` 合并为单个不可变 `Range` 的重命名保持清晰。

实现拆解

1. 在 `schedule_policy.py` 中, 将 `add_dllm_staging_req` 和 `add_chunked_req` 中的候选长度读取从 `req.extend_input_len` 替换为即时计算表达式 `len(req.full_untruncated_fill_ids) - len(req.prefix_indices)`, 仅在截断后通过 `req.set_extend_input_len(new_len)` 提交最终长度。
2. 在 `add_one_req_ignore_eos` 中, 所有对 `req.extend_input_len` 的预提交读取 (预算检查、SWA 检查、`_add_dllm_req` 调用、非分块预填充判断) 均替换为即时计算候选值, 保持提交时通过 `set_extend_input_len` 写入。
3. 在 `schedule_batch.py` 的 `init_next_round_input` 中, 删除末尾的 `self.set_extend_input_len(input_len - len(self.prefix_indices))` 调用, 消除每轮早期存储。
4. 在 `test_prefill_adder.py` 的 mock 中添加 `prefix_indices` 和 `full_untruncated_fill_ids` 属性, 使即时计算表达式在测试中可用。

关键文件:

- `python/sglang/srt/managers/schedule_policy.py` (模块 调度器; 类别 source; 类型 core-logic; 符号 `add_dllm_staging_req`, `add_chunked_req`, `add_one_req_ignore_eos`): 核心变更文件, 修改了 `add_dllm_staging_req`、`add_chunked_req`、`add_one_req_ignore_eos` 等方法, 将候选长度读取从存储字段改为即时计算, 并仅在提交时写入 `extend_input_len`。

- python/sglang/srt/managers/schedule_batch.py (模块 调度器; 类别 source; 类型 core-logic; 符号 init_next_round_input) : 移除了 init_next_round_input 末尾的 set_extend_input_len 调用, 消除了候选长度的早期存储点。
- test/registered/unit/managers/test_prefill_adder.py (模块 测试; 类别 test; 类型 test-coverage; 符号 create_mock_req) : 为 mock 请求添加了 prefix_indices 和 full_untruncated_fill_ids 属性, 使测试能正确支持即时计算表达式。

关键符号: add_dllm_staging_req, add_chunked_req, add_one_req_ignore_eos, init_next_round_input

关键源码片段

python/sglang/srt/managers/schedule_policy.py

核心变更文件, 修改了 add_dllm_staging_req、add_chunked_req、add_one_req_ignore_eos 等方法, 将候选长度读取从存储字段改为即时计算, 并仅在提交时写入 extend_input_len。

```
# python/sglang/srt/managers/schedule_policy.py

# 在 add_dllm_staging_req 和 add_chunked_req 中,
# 将原先直接读取 req.extend_input_len (双重语义) 改为即时计算候选长度:
cand_extend_input_len = len(req.full_untruncated_fill_ids) - len(
    req.prefix_indices
)
truncated = cand_extend_input_len > _rem_tokens
new_len = min(cand_extend_input_len, _rem_tokens)
# 仅在提交时通过 set_extend_input_len 写入最终长度
req.set_extend_input_len(new_len)
req.fill_len = len(req.prefix_indices) + new_len

# 在 add_one_req_ignore_eos 中, 所有预提交读取都替换为即时计算:
cand_extend_input_len = len(req.full_untruncated_fill_ids) - len(
    req.prefix_indices
)
paged_input = self.ceil_paged_tokens(cand_extend_input_len)
# ... 后续所有使用 req.extend_input_len 的地方改为使用 cand_extend_input_len
```

评论区精华

PR 没有 review 评论。根据 PR 描述, 作者指出变更有一个孤立的行为差异: dllm/staging 提交路径现在基于截断长度而非未截断候选值重新计算 extend_logprob_start_len, 这只在 dllm 分块预填充跨块输入 logprobs 时可观测, 且截断值更正确。

- 暂无高价值评论线程

风险与影响

- 风险:

1. 回归风险（低）：候选计算表达式 `len(req.full_untruncated_fill_ids) - len(req.prefix_indices)` 要求两个列表在调用时已正确填充。若 `full_untruncated_fill_ids` 在特定代码路径中未初始化或已修改，可能导致错误。但该字段由请求生命周期早期设置，不易遗漏。
2. dllm 分块预填充行为变化（低）：PR 描述指出 `extend_logprob_start_len` 现在基于截断长度，可能影响跨块输入 `logprobs` 的场景。这是作者认为正确的方向，但仍需确认。
3. 缺少测试覆盖：测试仅有 `mock` 属性添加，没有新增的功能测试验证各方法的候选计算正确性。

- 影响：

1. 对系统：消除了 `extend_input_len` 的双重语义，降低后续重构（如 #27610 中的字段合并）的复杂度和出错概率。
2. 对开发者：`Req.extend_input_len` 现在只有单一含义——已提交的扩展长度；需要候选长度时必须显式计算，避免了隐式状态混乱。
3. 对用户：无直接功能影响。仅 dllm 分块预填充场景下有极细微的 `logprob` 起始长度变化（更正确）。 - 风险标记：核心路径变更，缺少测试覆盖

关联脉络

- PR #27575 Base PR: PR 描述中指明堆叠在 #27575 之上。
- PR #27610 Follow-up: collapse length fields into Range: PR 描述中指明 #27610 堆叠在此 PR 之上，利用本 PR 消除双重语义后进行字段合并。