PR #33758 完整报告

sgl-project/sglang

[bugfix] Stop/EOS inside a spec accept run beats the max_new_tokens finish

合并时间：2026-08-08 06:18

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/33758

执行摘要

- 一句话：修复推测解码中停止标记被长度限制抢先结束的 bug
- 推荐动作：值得精读。PR 展示了修复核心解码路径优先级问题的典型方式：重排检查顺序 + 新增兜底限制，并用生产事故复现测试锁定行为。对于维护推理引擎的工程师来说，update_finish_state 的实现细节值得深入理解，避免后续改动破坏停止语义。

功能与动机

PR body 指出：Speculative decoding 可以提交一个多 token 的 accept run，其中既包含停止 token / stop 字符串，又跨过 max_new_tokens。旧的长度优先排序将这类步骤标记为 FINISH_LENGTH 且 finished_len == max_new_tokens，导致停止之后过接受的 token（例如接受 EOS 草稿后采样的 bonus token）泄漏到输出中，形成 [..., <eos>, <junk>]。需要让停止标记优先于长度限制，且停止匹配超出预算时既不泄漏也不超限。

实现拆解

1. 重排 Req.update_finish_state 中的检查顺序（python/sglang/srt/managers/schedule_batch.py）：将原本最前面的长度检查（len(output_ids) >= max_new_tokens 时直接 FINISH_LENGTH）移动到停止检查之后。这样 stop string、stop token / EOS 的匹配会先于长度上限被处理，修复接受运行中停止 token 被长度结束掩盖的问题。
2. 新增 _cap_finished_len_at_max_new_tokens 兜底方法：当停止匹配的 finished_len 超过 max_new_tokens 时，将 finished_reason 降级为 FINISH_LENGTH，并把 finished_len 截断到上限。该方法在词表边界、stop string、stop token 三条匹配路径命中后都会被调用，保证输出不会超过长度预算。
3. 调整 _check_vocab_boundary_finish 的越界判断：将 token_id >= self.vocab_size 改为仅当 vocab_size is not None 时才执行上界检查，下界检查 token_id < 0 始终保留。这使仅做 prefill 的 embedding / scoring 请求（vocab_size 为 None）也能安全走到长度检查，不会因与 None 比较而崩溃。
4. 新增纯 CPU 回归测试（test/registered/unit/managers/test_finish_length_speculative.py）：通过 register_cpu_ci 注册到 CPU 套件，驱动真实的 Req.update_finish_state，覆盖 EOS 中途命中、stop string 中途命中、超出预算降级、无停止按长度结束、vocab_size=None 兼容以及生产事故复现（<|eot|> + bonus junk token）等 9 个场景。

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 _cap_finished_len_at_max_new_tokens, update_finish_state, _check_vocab_boundary_finish）：核心逻辑修改文件：重排 update_finish_state 检查顺序，新增 _cap_finished_len_at_max_new_tokens 兜底，并调整词表边界检查以兼容 vocab_size=None。
- test/registered/unit/managers/test_finish_length_speculative.py（模块 测试；类别 test；类型 test-coverage；符号 _make_req, TestFinishLengthSpeculative, TestPostEosBonusTokenIncident）：新增的纯 CPU 回归测试，覆盖停止标记优先、超预算降级、vocab_size 为 None 及生产事故复现，是本次修复质量的关键保障。

关键符号：_cap_finished_len_at_max_new_tokens, update_finish_state, _check_vocab_boundary_finish

关键源码片段

python/sglang/srt/managers/schedule_batch.py

核心逻辑修改文件：重排 update_finish_state 检查顺序，新增 _cap_finished_len_at_max_new_tokens 兜底，并调整词表边界检查以兼容 vocab_size=None。

```
def _cap_finished_len_at_max_new_tokens(self) -> None:
    """将超出长度预算的停止匹配降级为长度结束。

    推测解码可能提交一个同时跨过 max_new_tokens 且包含停止标记的
    accept run；若停止位置在预算之外，继续保留停止会让输出超过
    上限，因此这里把 finished_reason 改为 FINISH_LENGTH，并将
    finished_len 截断到 max_new_tokens。
    """
    max_new_tokens = self.sampling_params.max_new_tokens
    if self.finished_len is not None and self.finished_len > max_new_tokens:
        self.finished_reason = FINISH_LENGTH(length=max_new_tokens)
        self.finished_len = max_new_tokens


def update_finish_state(self, new_accepted_len: int = 1):
    if self.finished():
        return

    if self.to_finish:
        self.finished_reason = self.to_finish
        self.to_finish = None
        return

    new_accepted_tokens = self.output_ids[-new_accepted_len:]

    # 先做词表边界清理，避免越界 token 进入后续 decode。
    if self._check_vocab_boundary_finish(new_accepted_tokens):
        self._cap_finished_len_at_max_new_tokens()
        return
```

```
    # 停止字符串优先于 stop token / EOS：推测解码一步可接受多个 token，
    # 基于 token 的检查只会截掉最后一个 token，导致停止字符串泄漏。
    if self._check_str_based_finish(new_accepted_len):
        self._cap_finished_len_at_max_new_tokens()
        return

    # 停止 token / EOS 优先于长度上限：accept run 可能在 EOS 落地的
    # 同一步跨过 max_new_tokens，若先按长度结束会保留 EOS 之后的
    # 多余 token（如 bonus token）。
    if self._check_token_based_finish(new_accepted_tokens):
        self._cap_finished_len_at_max_new_tokens()
        return

    # 没有命中任何停止条件时才按长度结束。
    if len(self.output_ids) >= self.sampling_params.max_new_tokens:
        self.finished_reason = FINISH_LENGTH(
            length=self.sampling_params.max_new_tokens
        )
        self.finished_len = self.sampling_params.max_new_tokens
        return

    if self.grammar is not None and self.grammar.is_terminated():
        self.finished_reason = FINISH_MATCHED_TOKEN(matched=self.output_ids[-1])
        return
```

test/registered/unit/managers/test_finish_length_speculative.py

新增的纯 CPU 回归测试，覆盖停止标记优先、超预算降级、vocab_size 为 None 及生产事故复现，是本次修复质量的关键保障。

```
class TestFinishLengthSpeculative(CustomTestCase):
    def test_eos_mid_run_beats_length_cap(self):
        # 一次推测解码提交了 [12, EOS, 20]，跨过 max_new_tokens=5 的同时
        # EOS 也落地。旧的长度优先逻辑会以 FINISH_LENGTH(finished_len=5)
        # 结束，并把 EOS 之后的过接受 token 泄漏进输出。
        req = _make_req([10, 11, 12, EOS_ID, 20], max_new_tokens=5)
        req.update_finish_state(new_accepted_len=3)
        self.assertTrue(req.finished())
        self.assertIsInstance(req.finished_reason, FINISH_MATCHED_TOKEN)
        self.assertEqual(req.finished_reason.matched, EOS_ID)
        self.assertEqual(req.finished_len, 4)
        self.assertEqual(
            list(req.output_ids_through_stop),
            [10, 11, 12, EOS_ID],
        )


class TestPostEosBonusTokenIncident(CustomTestCase):
    # 生产事故复现：spec verification 没有 EOS 意识，接受 <|eot|>
    # 后仍会采样 bonus token，target 的 post-EOS argmax 退化为原始字节
```

```
    # token 2，accept run 结束为 [..., 200008, 2]。finished_len 截断
    # 必须隐藏这个 junk，使每个输出都结束在 200008。
    def test_eot_then_bonus_junk_crossing_cap_is_trimmed(self):
        req = self._eot_req([10, 11, 12, EOT_ID, POST_EOS_JUNK_ID], max_new_tokens=5)
        req.update_finish_state(new_accepted_len=3)
        self.assertTrue(req.finished())
        self.assertIsInstance(req.finished_reason, FINISH_MATCHED_TOKEN)
        self.assertEqual(req.finished_reason.matched, EOT_ID)
        self.assertEqual(list(req.output_ids_through_stop), [10, 11, 12, EOT_ID])
        self.assertNotIn(POST_EOS_JUNK_ID, req.output_ids_through_stop)
```

评论区精华

PR 无实质性的代码审核讨论，评论中仅出现 3 次 /tag-and-rerun-ci 命令（作者和维护者触发 CI 重跑）。合并者 hanming-lu 在最终 commit 中修复了 vocab_size 为 None 时的比较问题，属于维护者参与完善。

- CI 重跑 (other): 仅触发 CI 重跑，无代码变更。

风险与影响

- 风险：
  1. 核心路径变更：update_finish_state 是每个生成请求结束路径的必经逻辑，重排检查顺序可能影响非推测解码场景下的停止行为（例如原本先按长度结束但现在会匹配到停止）。作者已运行现有的 stop/grammar/streamer 相关测试（14 passed），但更大规模的端到端回归仍需 CI 覆盖。
  2. 语义变化：停止匹配优先会让 finished_reason 从 FINISH_LENGTH 变为 FINISH_MATCHED_TOKEN/STR，下游如果依赖 finish reason 做日志、统计或客户端提示，需要确认兼容。
  3. vocab_size 为 None 时的兼容：跳过上界检查后，若 token_id 为超大正值但 vocab_size 未知，将不会被错误截断，但负 token 仍会被拒绝，行为符合预期。 - 影响：影响用户：修复推测解码下输出包含 EOS 后垃圾 token（如 [..., <eos>, <junk>]）的问题，提升输出质量和合规性；stop string 场景同理。影响系统：完成状态逻辑在所有生成请求路径上共用，重排是纯顺序调整，无新增计算，对性能无影响。影响团队：新增的单元测试为推测解码停止 / 长度交互提供了可复用的回归保护，后续改动可通过该测试快速验证。 - 风险标记：核心路径变更，行为语义变化，vocab_size 兼容性

关联脉络

- 暂无明显关联 PR