

PR #29122 完整报告

sgl-project/sglang

[Spec] Make the overlap bonus-token relay unconditional

合并时间: 2026-06-25 06:34

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/29122>

执行摘要

- 一句话: 无条件 relay bonus_tokens, 删除废弃标志
- 推荐动作: 该 PR 是紧随 #29118 的清理步骤, 值得快速合并以保持代码整洁。但建议在合并 #29118 之后验证确认没有回归。

功能与动机

PR body 指出: "Once dflash joins the shared **bonus_tokens** relay channel, **need_bonus_tokens** is true for every spec algorithm that reaches the FutureMap relay". 因此该标志变得多余, 删除它可以简化代码。

实现拆解

1. 在 overlap_utils.py 的 _lazy_init_forward_buf 方法中, 删除 self.need_bonus_tokens = getattr(draft_input, "bonus_tokens", None) is not None 行。
2. 在 _resolve_spec_extras 方法中, 将原来受 self.need_bonus_tokens 条件保护的 draft_input.bonus_tokens = bonus_tokens 改为无条件执行 (对应 if self.need_topk: 分支内和 elif 分支)。
3. 在 stash 方法中, 移除 if self.need_bonus_tokens: 条件, 直接无条件执行 self.output_tokens_buf[indices] = draft_input.bonus_tokens.to(...).
4. 在 _DEBUG_ASSERT 块中, 移除 if self.need_bonus_tokens: 条件, 直接对 bonus_tokens 进行断言检查。

关键文件:

- python/sglang/srt/managers/overlap_utils.py (模块 调度器; 类别 source; 类型 core-logic; 符号 FutureMap, _lazy_init_forward_buf, _resolve_spec_extras, stash) : 唯一修改的文件, 核心中继逻辑所在。删除 need_bonus_tokens 标志及相关条件, 使 bonus_tokens relay 无条件执行, 简化了 FutureMap 类的内部逻辑。

关键符号: _lazy_init_forward_buf, _resolve_spec_extras, stash

关键源码片段

[python/sglang/srt/managers/overlap_utils.py](#)

唯一修改的文件，核心中继逻辑所在。删除 `need_bonus_tokens` 标志及相关条件，使 `bonus_tokens relay` 无条件执行，简化了 `FutureMap` 类的内部逻辑。

```
# python/sglang/srt/managers/overlap_utils.py
class FutureMap:
    def _lazy_init_forward_buf(self, draft_input: EagleDraftInput):
        self._forward_buf_initialized = True
        # need_bonus_tokens 已删除，因为所有 spec 算法均通过 bonus_tokens 通道 relay
        self.need_topk = self.spec_algo.need_topk()
        self.need_hidden_states = (
            spec_need_hidden_states()
            and getattr(draft_input, "hidden_states", None) is not None
        )
        # ... 根据 need_topk / need_hidden_states 分配 buffer ...

    def _resolve_spec_extras(self, batch: ScheduleBatch) -> None:
        # ...
        if self.need_topk:
            # ... gather_spec_extras 返回 bonus_tokens ...
            draft_input.bonus_tokens = bonus_tokens # 无条件赋值
            # ...
        else:
            draft_input.bonus_tokens = self.output_tokens_buf[indices] # 无条件赋值
        # debug assert 也始终执行
        if _DEBUG_ASSERT:
            _assert_nonneg_and_invalidate(
                draft_input.bonus_tokens, self.output_tokens_buf, indices
            )

    def stash(self, payload: EagleDraftInput, indices: torch.Tensor) -> None:
        # ...
        self.output_tokens_buf[indices] = draft_input.bonus_tokens.to(
            self.output_tokens_buf.dtype
        ) # 无条件 stash
        # ...
```

评论区精华

该 PR 没有 review 评论，属于作者自合并的清理性变更。

- 暂无高价值评论线程

风险与影响

- 风险：风险较低。变更本质是删除一个在所有 speculative decoding 算法中均为 `True` 的标志，理论上不影响行为。但需确保未来引入新的 speculative decoding 算法时，如果其 `bonus_tokens` 字段可能为 `None`，会导致无条件赋值失败。不过从 PR body 描述看，所有现有算法均已填充该字段。

- 影响：直接影响 FutureMap 类的 `_lazy_init_forward_buf`、`_resolve_spec_extras` 和 `stash` 方法，消除条件分支，使 `bonus_tokens` 始终被 relay。不影响外部接口或用户可见行为。
- 风险标记：缺少测试覆盖

关联脉络

- PR #29118 [Spec] Dflash joins the shared `bonus_tokens` relay channel: 本 PR 基于 #29118，在 `dflash` 加入 `bonus_tokens` 中继后，清理废弃标志。