PR #26768 完整报告

sgl-project/sglang

Refactor simulated acceptance length generation

合并时间：2026-06-04 09:31

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26768

执行摘要

- 一句话：提取模拟接受长度采样函数以复用
- 推荐动作：值得合入，提高代码复用性。建议关注 _sample_simulated_acc_len 的用户，并考虑添加单元测试覆盖。

功能与动机

方便其他推测解码算法复用模拟接受长度生成逻辑。

实现拆解

1. 在 python/sglang/srt/speculative/spec_utils.py 中新增 _sample_simulated_acc_len(simulate_acc_len, simulate_acc_method, max_len) 函数，返回采样后的整数长度。
2. 将 generate_simulated_accept_index 中原有的采样代码替换为对 _sample_simulated_acc_len 的调用。
3. 将硬编码的 spec_steps + 1 替换为参数 max_len，使其更通用。
4. 修复错误消息中使用 SIMULATE_ACC_METHOD 全局常量而非 simulate_acc_method 参数的 bug。

关键文件：

- python/sglang/srt/speculative/spec_utils.py（模块 推测解码；类别 source；类型 core-logic；符号 _sample_simulated_acc_len, generate_simulated_accept_index）：核心重构文件，提取采样逻辑为独立函数，修复错误消息 bug。

关键符号：_sample_simulated_acc_len, generate_simulated_accept_index

关键源码片段

python/sglang/srt/speculative/spec_utils.py

核心重构文件，提取采样逻辑为独立函数，修复错误消息 bug。

```
# python/sglang/srt/speculative/spec_utils.py

def _sample_simulated_acc_len(
    simulate_acc_len: float,
```

```
    simulate_acc_method: str,
    max_len: int,
) -> int:
    """
    Sample a simulated acceptance length in [1, max_len].

    提取自 generate_simulated_accept_index，供其他推测解码算法复用。
    max_len 替代原来的 spec_steps + 1，更加通用。
    """
    if simulate_acc_method == "multinomial":
        simulated_values = torch.normal(
            mean=simulate_acc_len,
            std=1.0,
            size=(1,),
            device="cpu",
        )
        # clamp simulated values to be between 1 and max_len
        simulated_values = torch.clamp(simulated_values, min=1.0, max=max_len)
        simulate_acc_len = int(simulated_values.round().item())
    elif simulate_acc_method == "match-expected":
        simulate_acc_len = max(1.0, min(max_len, simulate_acc_len))
        lower = int(simulate_acc_len // 1)
        upper = lower + 1 if lower < max_len else lower
        if lower == upper:
            simulate_acc_len = lower
        else:
            weight_upper = simulate_acc_len - lower
            weight_lower = 1.0 - weight_upper
            probs = torch.tensor([weight_lower, weight_upper], device="cpu")
            sampled_index = torch.multinomial(probs, num_samples=1)
            simulate_acc_len = lower if sampled_index == 0 else upper
    else:
        raise ValueError(f"Invalid simulate_acc_method: {simulate_acc_method}")
    return int(simulate_acc_len)


def generate_simulated_accept_index(
    accept_index,
    predict,
    num_correct_drafts,
    bs,
    spec_steps,
    simulate_acc_len: float = SIMULATE_ACC_LEN,
    simulate_acc_method: str = SIMULATE_ACC_METHOD,
):
    assert simulate_acc_len > 0.0
    # 使用提取的函数
    simulate_acc_len = _sample_simulated_acc_len(
        simulate_acc_len, simulate_acc_method, spec_steps + 1
```

```
    )
    # ... 后续逻辑不变
```

评论区精华

无 review 评论。自动机器人确认了代码重构并指出错误消息修复。

- 暂无高价值评论线程

风险与影响

- 风险：风险较低。提取的函数逻辑与原来一致，仅改变组织方式。参数 max_len 替代 spec_steps + 1 需要调用方确保传入正确值。
- 影响：对现有功能无影响，重构后 generate_simulated_accept_index 行为不变。未来其他推测解码算法可直接调用 _sample_simulated_acc_len。
- 风险标记：暂无

关联脉络

- 暂无明显关联 PR