PR #25109 完整报告

sgl-project/sglang

spec: defer verify() idle hidden_size to worker fixup

合并时间：2026-05-13 13:13

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25109

执行摘要

- 一句话：将 idle 时 hidden_size 计算延迟到 worker fixup 统一处理
- 推荐动作：值得精读，展示了如何通过延迟绑定（lazy binding）消除重复逻辑，是良好的架构清理范例。可关注 hidden_size_for 方法的设计。

功能与动机

PR body 指出：'Schema cleanup: EagleVerifyInput.verify() idle stubs no longer compute hidden_size locally from batch.model_config.spec_hidden_size; they hand off to the worker fixup in forward_draft_extend_after_decode, which rebuilds via EagleDraftExtendInput.hidden_size_for(self) (single source, incl. EAGLE-3 aux widening).' 目的是统一 hidden_size 来源，避免 verify() 中直接引用 model_config，提高代码一致性和可维护性。

实现拆解

1. *eagle_info.py*：verify() idle 分支传 None：在两个 idle stub（完全 idle batch 和所有请求完成的 batch）中，*create_idle_input()* 的 hidden_size 和 dtype 参数改为 None，并添加注释说明由 worker fixup 重建。
2. *eagle_worker.py / multi_layer_eagle_worker.py*：forward_draft_extend_after_decode 修复条件：将 if not input_is_idle and draft_extend_input.input_ids.shape[0] == 0 改为 if draft_extend_input.input_ids.shape[0] == 0，使得 fully-idle batch（DP rank 无请求）也能进入修复路径，并通过 prepare_for_idle() 的幂等性安全处理。修复路径内使用 EagleDraftExtendInput.hidden_size_for(self) 作为 single source。
3. 注释更新：所有相关位置添加了注释说明变更意图。

关键文件：

- python/sglang/srt/speculative/eagle_info.py（模块 投机解码；类别 source；类型 core-logic）：核心变更：verify() idle stub 中 hidden_size/dtype 传 None，将计算责任转移给 worker。
- python/sglang/srt/speculative/eagle_worker.py（模块 投机解码；类别 source；类型 core-logic）：worker 侧修复条件：将 not input_is_idle and 改为直接检查 input_ids.shape[0]==0，使 fully-idle 分支也被覆盖。

- python/sglang/srt/speculative/multi_layer_eagle_worker.py（模块 投机解码；类别 source；类型 core-logic）：与 eagle_worker.py 相同的条件修复，用于多层的 Eagle worker。

关键符号：verify, forward_draft_extend_after_decode

关键源码片段

python/sglang/srt/speculative/eagle_info.py

核心变更：verify() idle stub 中 hidden_size/dtype 传 None，将计算责任转移给 worker。

```
# python/sglang/srt/speculative/eagle_info.py
if batch.forward_mode.is_idle():
    # hidden_size=None: worker fixup 在 forward_draft_extend_after_decode
    # 中通过 EagleDraftExtendInput.hidden_size_for(worker) 重建
    # （单一来源，包含 EAGLE-3 aux widening）。
    draft_extend_input = EagleDraftExtendInput.create_idle_input(
        device=batch.device,
        hidden_size=None, # 之前是 batch.model_config.spec_hidden_size
        dtype=None, # 之前是 batch.model_config.dtype
        capture_hidden_mode=CaptureHiddenMode.LAST,
    )
    return EagleVerifyOutput.create_idle(
        draft_extend_input=draft_extend_input,
        logits_output=logits_output,
        device=batch.device,
        spec_steps=self.spec_steps,
    )
```

python/sglang/srt/speculative/eagle_worker.py

worker 侧修复条件：将 not input_is_idle and 改为直接检查 input_ids.shape[0]==0，使 fully-idle 分支也被覆盖。

```
# python/sglang/srt/speculative/eagle_worker.py
if draft_extend_input.input_ids.shape[0] == 0:
    # 通过 hidden_size_for(self) 获得单一隐藏大小来源（包含 EAGLE-3 aux widening）。
    # 两个 stub 来源：完全 idle batch（DP attention rank 无请求）和所有请求完成的 active
    batch。
    # prepare_for_idle() 在已经是 idle 的 batch 上是幂等的。
    batch = batch.copy()
    batch.prepare_for_idle()
    draft_extend_input = EagleDraftExtendInput.create_idle_input(
        device=self.device,
        hidden_size=EagleDraftExtendInput.hidden_size_for(self),
        dtype=EagleDraftExtendInput.dtype_for(self),
        capture_hidden_mode=draft_extend_capture_mode,
    )
    batch.spec_info = draft_extend_input
```

python/sglang/srt/speculative/multi_layer_eagle_worker.py

与 eagle_worker.py 相同的条件修复，用于多层的 Eagle worker。

```
# python/sglang/srt/speculative/multi_layer_eagle_worker.py
if draft_extend_input.input_ids.shape[0] == 0:
    # 注释同 eagle_worker.py，隐藏大小来源统一。
    batch = batch.copy()
    batch.prepare_for_idle()
    draft_extend_input = EagleDraftExtendInput.create_idle_input(
        device=self.device,
        hidden_size=EagleDraftExtendInput.hidden_size_for(self),
        dtype=EagleDraftExtendInput.dtype_for(self),
        capture_hidden_mode=draft_extend_capture_mode,
    )
    batch.spec_info = draft_extend_input
```

评论区精华

无 review 评论，PR 由作者自行合并。

- 暂无高价值评论线程

风险与影响

- 风险：低风险。变更仅影响 idle stub 路径（无实际计算），通过已有测试验证（测试全部通过）。但需注意：hidden_size=None 的 idle input 在进入 worker fixup 之前被误使用可能导致错误，但所有下游调用都已同步修改。
- 影响：影响范围：仅 speculative decoding 的 idle 分支，无用户可见行为变化。对系统：减少了 model_config 的直接依赖，使 hidden_size 获取路径单一化，有利于后续 EAGLE-3 等算法扩展。
- 风险标记：逻辑变更影响 idle 路径，缺少测试覆盖

关联脉络

- PR #25015 Fix Eagle draft decode positions: 同属 speculative decoding 模块，修复 Eagle 投机解码错误。