

PR #27478 完整报告

sgl-project/sglang

[Spec] Guard async-assert probes against `None` tensor

合并时间: 2026-06-07 13:14

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/27478>

执行摘要

- 一句话: 修复 async probe 对 None 张量的崩溃
- 推荐动作: 建议合并。修复直击根因, 代码简洁, 有 CI 验证。可关注后续是否需为其他类似场景添加 None 守卫。

功能与动机

STANDALONE speculative decoding 的 draft 模型使用 `capture_hidden_mode=NULL`, 导致 `logits_output.hidden_states` 合法为 `None`, 但 async probe 未处理该情况, 在 #26335 添加 probe 后触发崩溃。

实现拆解

1. 在 `python/sglang/srt/utils/async_probe.py` 中导入 `Optional` 类型。
2. 将四个 probe 函数的参数类型从 `torch.Tensor` 改为 `Optional[torch.Tensor]`。
3. 在 `maybe_detect_nan` 和 `maybe_detect_inf` 中增加 `if tensor is None: return` 提前返回。
4. 在 `maybe_detect_oob` 和 `maybe_detect_page_aligned` 中将 `indices is None` 条件合并到已有的 `numel() == 0` 短路判断中, 避免在 `None` 上调用 `.numel()`。
5. 无测试文件变更, 但 CI 触发相关测试验证。

关键文件:

- `python/sglang/srt/utils/async_probe.py` (模块 工具层; 类别 source; 类型 core-logic; 符号 `maybe_detect_nan`, `maybe_detect_inf`, `maybe_detect_oob`, `maybe_detect_page_aligned`): 变更唯一文件, 包含所有 probe 函数的 None 守卫和类型更新。

关键符号: `maybe_detect_nan`, `maybe_detect_inf`, `maybe_detect_oob`, `maybe_detect_page_aligned`

关键源码片段

`python/sglang/srt/utils/async_probe.py`

变更唯一文件, 包含所有 probe 函数的 `None` 守卫和类型更新。

```
"""Async invariant probes — fire torch._assert_async without CPU sync.
```

All probes are gated on SGLANG_ENABLE_ASYNC_ASSERT (default off in prod).
When the gate is on, a violation surfaces as an assertion at the next CUDA
sync point instead of as a silent NaN cascade or illegal-address crash.
"""

```
from typing import Optional
```

```
import torch
```

```
from sglang.srt.environ import envs
```

```
def maybe_detect_nan(tensor: Optional[torch.Tensor], msg: str = ""):
    """Async NaN check — no GPU-CPU sync, error surfaces at next sync point."""
    if not envs.SGLANG_ENABLE_ASYNC_ASSERT.get():
        return
    # A None tensor means there is nothing to probe, e.g. hidden_states on
    # capture_hidden_mode=NULL paths (STANDALONE speculative decoding).
    if tensor is None:
        return
    torch._assert_async(~torch.any(torch.isnan(tensor)), f"NaN detected! {msg}")
```

```
def maybe_detect_inf(tensor: Optional[torch.Tensor], msg: str = ""):
    """Async Inf check — fp16 overflow surfaces as Inf before NaN."""
    if not envs.SGLANG_ENABLE_ASYNC_ASSERT.get():
        return
    if tensor is None:
        return
    torch._assert_async(~torch.any(torch.isinf(tensor)), f"Inf detected! {msg}")
```

```
def maybe_detect_oob(indices: Optional[torch.Tensor], low: int, high: int, msg: str):
    """Async OOB check — no GPU-CPU sync, error surfaces at next sync point."""
    if not envs.SGLANG_ENABLE_ASYNC_ASSERT.get():
        return
    if indices is None or indices.numel() == 0:
        return
    torch._assert_async(
        (indices.min() >= low) & (indices.max() < high),
        f"OOB indices not in [{low}, {high}]: {msg}",
    )
```

```
def maybe_detect_page_aligned(
    indices: Optional[torch.Tensor], page_size: int, msg: str
):
    """Async page-alignment check on slot ids."""
    if not envs.SGLANG_ENABLE_ASYNC_ASSERT.get():
```

```
    return
if indices is None or indices.numel() == 0 or page_size <= 1:
    return
torch._assert_async(
    (indices % page_size == 0).all(),
    f"page-misaligned indices (page_size={page_size}): {msg}",
)
```

评论区精华

无 review 评论或讨论。

- 暂无高价值评论线程

风险与影响

- 风险：低风险。变更仅在 probe 函数入口添加 None 检查，不影响原有逻辑；CI 已通过 STANDALONE 推测解码测试。由于 GIL 和异步执行，None 检查后 tensor 仍可能变为 None？但实际场景中 tensor 在框架内是稳定的，风险可忽略。
- 影响：影响范围仅限于启用 SGLANG_ENABLE_ASYNC_ASSERT 且使用 STANDALONE 推测解码的用户。修复后，这些用户不会再因 probe 崩溃而中断引擎。对其他用户无影响。
- 风险标记：低风险

关联脉络

- PR #26335 Add async invariant probes (NaN/Inf/OOB/page-aligned): 此 PR 添加的 probe 引入了对 None tensor 的未处理调用，本 PR 修复了该问题。
- PR #27461 Enable async-assert invariant probes by default in CI: 相关 CI 配置，确保 probe 在 CI 中生效，本 PR 防止 probe 在 STANDALONE spec 中崩溃。