PR #26335 完整报告

sgl-project/sglang

[Spec] Async-assert probes across EAGLE/MTP; zero `tgt_cache_loc`

合并时间：2026-05-27 02:34

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26335

执行摘要

- 一句话：为 EAGLE/MTP 添加异步断言探测并清理 NaN 检测路径
- 推荐动作：建议 SRT speculative 路径维护者和使用 DeepSeek 系列模型用户关注本 PR 的防御性增强。环境变量合并和废弃标志移除的设计值得参考，异步断言的使用模式可在类似场景复用。若频繁遇到 flaky 测试或非法地址问题，建议在 CI 中启用 SGLANG_ENABLE_ASYNC_ASSERT 环境变量。

功能与动机

引用 PR body：test_eagle_infer_b::TestEAGLEServerAdditional (page_size=256, topk=5, fa3, fp16) 测试已不稳定数天，有三种失败模式：NaN 目标 logits、copy_all_layer_kv_cache_tiled 中的非法地址、embed_tokens(input_ids) 中 token id 超出 vocab size。本 PR 是深度防御 + 仪器化，不声称修复 bug，只缩小非法地址暴露面并确保下一次触发时能给出精确的异步断言。

实现拆解

1. 创建独立探测模块：新增 python/sglang/srt/utils/async_probe.py，包含四个异步断言函数 maybe_detect_nan、maybe_detect_inf、maybe_detect_oob、maybe_detect_page_aligned。所有函数通过 SGLANG_ENABLE_ASYNC_ASSERT 环境变量门控，默认关闭，避免生产环境开销。
2. 修改 get_src_tgt_cache_loc 默认值：在 spec_utils.py 中将 tgt_cache_loc 从 torch.empty_like 改为 torch.zeros_like，使得任何未被覆盖的尾部索引落在预留的 padding slot 0 上，而非缓存分配器返回的随机垃圾值，从而减少非法地址触发概率。
3. 插入探测点：在 eagle_info.py、eagle_info_v2.py、eagle_worker.py、eagle_worker_v2.py、multi_layer_eagle_worker_v2.py、frozen_kv_mtp_worker.py 等文件的合适位置（如 softmax 后、top_k/top_p renorm 后、accept_index 索引前、input_ids 使用前）添加 maybe_detect_nan/inf/oob 调用，覆盖每种推测解码变体的关键数据约束。
4. 清理废弃功能：删除 server_args.py 中的 enable_nan_detection 字段和 --enable-nan-detection CLI 参数；移除 sampler.py 的 _preprocess_logits 中基于 CPU 同步的 NaN 替换逻辑（torch.where + crash_on_warnings 分支），因为异步断言已取代其角色；在 environ.py 中将两个独立 env 合并为一个 SGLANG_ENABLE_ASYNC_ASSERT。

5. 内存池防御：在 mem_cache/memory_pool.py 的 move_kv_cache 入口添加 tgt_loc / src_loc 越界探测，避免因 OOB 索引导致 copy_all_layer_kv_cache_tiled 内核中的非法地址。

关键文件：

- python/sglang/srt/utils/async_probe.py（模块 异步探测；类别 source；类型 core-logic；符号 maybe_detect_nan, maybe_detect_inf, maybe_detect_oob, maybe_detect_page_aligned）：新增的核心模块，包含全部四个异步断言函数(maybe_detect_nan/inf/oob/page_aligned)，所有探测通过 SGLANG_ENABLE_ASYNC_ASSERT 环境变量门控。
- python/sglang/srt/speculative/spec_utils.py（模块 推测解码；类别 source；类型 core-logic；符号 maybe_detect_nan, maybe_detect_oob）：移除原生的 maybe_detect_nan/oob 函数定义，并修改 get_src_tgt_cache_loc 将 empty_like 改为 zeros_like，显著降低因未初始化缓存位置导致的非法地址风险。
- python/sglang/srt/layers/sampler.py（模块 采样器；类别 source；类型 core-logic）：移除基于 CPU 同步的 NaN 检测与替换逻辑（torch.where 分支）以及对应的 use_nan_detection 属性和 crash_on_warnings 导入，简化采样器代码。
- python/sglang/srt/server_args.py（模块 参数配置；类别 source；类型 configuration）：移除废弃的 enable_nan_detection 字段及其对应的 CLI 参数 --enable-nan-detection，同时移除 _handle_deprecated_args 中的迁移逻辑。
- python/sglang/srt/environ.py（模块 环境配置；类别 source；类型 configuration）：合并 SGLANG_SPEC_NAN_DETECTION 和 SGLANG_SPEC_OOB_DETECTION 为单一的 SGLANG_ENABLE_ASYNC_ASSERT 环境变量，简化配置入口。
- python/sglang/srt/speculative/eagle_info.py（模块 推测解码；类别 source；类型 dependency-wiring）：添加多个 maybe_detect_oob 和 maybe_detect_nan 探测点，覆盖 input_ids、accept_index、num_correct_drafts、accept_tokens 以及 top_k/top_p renorm 后的 target_probs。
- python/sglang/srt/mem_cache/memory_pool.py（模块 缓存管理；类别 source；类型 dependency-wiring）：在 move_kv_cache 入口添加 tgt_loc 和 src_loc 的越界探测，防止 copy_all_layer_kv_cache_tiled 内核因 OOB 索引而崩溃。

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

关键源码片段

python/sglang/srt/speculative/spec_utils.py

移除原生的 maybe_detect_nan/oob 函数定义，并修改 get_src_tgt_cache_loc 将 empty_like 改为 zeros_like，显著降低因未初始化缓存位置导致的非法地址风险。

```
@torch.compile(dynamic=True, disable=_is_npu)
def get_src_tgt_cache_loc(
    seq_lens: torch.Tensor,
    out_cache_loc: torch.Tensor,
```

```
    accept_index: torch.Tensor,
    num_correct_drafts: torch.Tensor,
    draft_token_num: int,
    page_size: int,
):
    src_cache_loc = out_cache_loc[accept_index]
    # Use zeros_like instead of empty_like: any uncovered tail stays at slot 0
    # (reserved padding) instead of caching-allocator garbage, reducing risk
    # of illegal-address crashes in subsequent gather kernels.
    tgt_cache_loc = torch.zeros_like(src_cache_loc)
    extended_len = seq_lens + draft_token_num
    keep_len = torch.minimum(
        (seq_lens + num_correct_drafts + 1 + page_size - 1) // page_size * page_size,
        extended_len,
    )
    to_free_num_slots = extended_len - keep_len
    return src_cache_loc, tgt_cache_loc, to_free_num_slots
```

评论区精华

本 PR 无人工 review 讨论，仅有一个自动机器人的配额警告评论，与技术内容无关。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. 废弃的 --enable-nan-detection CLI 参数移除可能破坏依赖该参数的旧启动脚本；
  2. 合并环境变量后，旧变量 SGLANG_SPEC_NAN_DETECTION 和 SGLANG_SPEC_OOB_DETECTION 不再生效，已无迁移代码，用户需手动设置 SGLANG_ENABLE_ASYNC_ASSERT；
  3. 异步断言依赖 torch._assert_async，在某些 PyTorch 版本或非 CUDA 后端上可能不可用或行为不一致；
  4. 探测调用默认关闭，但若错误启用可能带来微小 GPU 开销；
  5. 移除 sampler 中 NaN 恢复路径后，若其他路径仍有 NaN 且探测未打开，可能导致更快崩溃（但这是期望行为）。 - 影响：影响范围：直接涉及 sglang/srt 中所有 EAGLE v1/v2、multi-layer v2、frozen-KV MTP 以及内存池模块的推测解码路径。用户无感知（探测默认关闭）。对于使用 --enable-nan-detection 的用户，需迁移至 SGLANG_ENABLE_ASYNC_ASSERT=1。开发者在调试 flaky 测试时将受益于更精确的异步断言定位。 - 风险标记：废弃 CLI 移除可能破坏旧脚本，环境变量迁移无后向兼容，异步断言依赖 PyTorch 实现，探测默认关闭但若误开有微小开销

关联脉络

- PR #26358 Revert "[perf][spec decoding] Skip full-vocab softmax in EAGLE draft when topk == 1 (#26235)": 同为 speculative decoding 路径的稳定性相关 PR，涉及相同

文件（eagle_worker_v2.py、eagle_draft_extend_cuda_graph_runner.py），本 PR 的探测增强有助于诊断类似问题时提供更早的断言。

- PR #26397 Reland "[perf][spec decoding] Skip full-vocab softmax in EAGLE draft when topk == 1 (#26235)": 与上一 PR 关联的重新提交，同样涉及 speculative 路径优化，本 PR 的探测机制可帮助验证优化后的数值稳定性。