PR #26821 完整报告

sgl-project/sglang

Add periodic KV-canary stats logging and kernel-run-counter health check

合并时间：2026-05-31 10:00

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26821

执行摘要

- 一句话：增加 KV-canary 周期性统计日志与健康检查
- 推荐动作：该 PR 设计清晰，纯观察者模式值得学习；DelayedDeviceHostHandler 的异步 D2H 拷贝模式可复用于其他需要获取设备状态但不阻塞前向的场景。阅读重点：health_checker.py 中的延迟出队和增量计算逻辑。

功能与动机

PR 说明中强调两个组件均为纯观察者（pure observers of CanaryDeviceState counters and the sweep orchestrator），不依赖其输出。目的是增加可观测性，特别是检测 canary kernel wiring 退化的静默故障。

实现拆解

1. 新增 KernelRunCounterHealthChecker（health_checker.py）：构造时保存设备状态、活跃标签和步数获取器；step() 方法通过 DelayedDeviceHostHandler 异步将设备上的 kernel_run_counters 拷贝到主机；在主机端计算增量，若某个标签计数器未增加则抛出 RuntimeError。检查每 100 步执行一次，前 100 步为预热期。
2. 新增 PeriodicCanaryStatsLogger（stats_logger.py）：step() 时通过 DelayedDeviceHostHandler 读取设备上的 slot_run_counters 总和及 violation_write_index，然后在主机端打印 INFO 日志，包含步数、受保护 token 数、清扫轮次、违规数和活跃标签占比。打印频率由 CanaryConfig.stats_print_every_n_steps 控制（环境变量 SGLANG_KV_CANARY_STATS_PRINT_EVERY_N_STEPS，默认 100，设为 0 关闭）。
3. 集成配置：在 CanaryConfig（config.py）新增 stats_print_every_n_steps 字段，在 environ.py 添加环境变量默认值 100。
4. 集成到 CanaryManager（canary_manager.py）：在 __init__ 中构造两个对象，在 _post_ops_outside_graph 末尾依次调用 _health_checker.step() 和 _stats_logger.step()。
5. 测试配套：新增 test/registered/kv_canary/test_self_unit_runner_health.py，覆盖健康检查的计数器停滞检测、忽略 disable sweep 时的清扫标签、以及统计日志的间隔打印；同时修改 runner_test_base.py 和 fixtures.py 以传递新配置参数。

关键文件：

- python/sglang/srt/kv_canary/runner/health_checker.py（模块 健康检查；类别 source；类型 core-logic；符号 KernelRunCounterHealthChecker, init, step, _compute_on_device）：核心新增：KernelRunCounterHealthChecker 实现健康检查逻辑，通过 DelayedDeviceHostHandler 异步读取 kernel_run_counters 并计算增量，检测停滞。
- python/sglang/srt/kv_canary/runner/stats_logger.py（模块 统计日志；类别 source；类型 core-logic；符号 PeriodicCanaryStatsLogger, init, step, _compute_on_device）：核心新增：PeriodicCanaryStatsLogger 实现周期性统计日志，读取设备状态并打印到 INFO 日志。
- test/registered/kv_canary/test_self_unit_runner_health.py（模块 单元测试；类别 test；类型 test-coverage；符号 TestSelfUnitManagerHealth, test_kernel_run_counter_watchdog_raises_on_zero, test_kernel_run_counter_watchdog_ignores_sweep_when_sweep_is_disabled, test_periodic_stats_log_every_n_step）：新增单元测试覆盖健康检查和统计日志的核心场景。
- python/sglang/srt/kv_canary/runner/canary_manager.py（模块 管理器；类别 source；类型 dependency-wiring）：集成点：在 CanaryManager 中构造两个新对象并每步调用 step()。
- python/sglang/srt/kv_canary/config.py（模块 配置；类别 source；类型 configuration）：配置变更：新增 stats_print_every_n_steps 字段及其环境变量绑定。
- python/sglang/srt/environ.py（模块 环境变量；类别 source；类型 configuration）：环境变量注册：添加 SGLANG_KV_CANARY_STATS_PRINT_EVERY_N_STEPS 默认值。
- python/sglang/test/kv_canary/runner_test_base.py（模块 测试基础；类别 test；类型 test-coverage）：测试基础修改：适配新配置参数。
- python/sglang/test/kv_canary/fixtures.py（模块 测试 fixtures；类别 test；类型 test-coverage）：测试 fixture 修改：适配新配置参数。
- test/registered/kv_canary/test_self_unit_buffer_alloc.py（模块 缓冲区测试；类别 test；类型 test-coverage）：无关测试修改：适配新配置参数（可能因为 fixture 签名变更导致）。

关键符号：KernelRunCounterHealthChecker.init, KernelRunCounterHealthChecker.step, KernelRunCounterHealthChecker._compute_on_device, KernelRunCounterHealthChecker._postprocess_on_host, KernelRunCounterHealthChecker._expected_active_tags_for_health_check, PeriodicCanaryStatsLogger.init, PeriodicCanaryStatsLogger.step, PeriodicCanaryStatsLogger._compute_on_device, PeriodicCanaryStatsLogger._postprocess_on_host, TestSelfUnitManagerHealth.test_kernel_run_counter_watchdog_raises_on_zero, TestSelfUnitManagerHealth.test_kernel_run_counter_watchdog_ignores_sweep_when_sweep_is_disabled, TestSelfUnitManagerHealth.test_periodic_stats_log_every_n_step, TestKernelRunCounterDeltaCheck.test_first_check_raises_when_counter_never_incremented

关键源码片段

python/sglang/srt/kv_canary/runner/stats_logger.py

核心新增：PeriodicCanaryStatsLogger 实现周期性统计日志，读取设备状态并打印到 INFO 日志。

```
from __future__ import annotations

import logging
from collections.abc import Callable
from typing import Any, Optional

import torch

from sglang.jit_kernel.kv_canary.verify import CanaryLaunchTag
from sglang.srt.kv_canary.config import CanaryConfig
from sglang.srt.kv_canary.runner.future_tensor import DelayedDeviceHostHandler
from sglang.srt.kv_canary.runner.sweep import SweepOrchestrator
from sglang.srt.kv_canary.state import CanaryDeviceState

logger = logging.getLogger(__name__)


class PeriodicCanaryStatsLogger:
    """
    周期统计日志器：每隔 N 步将 canary 的关键统计信息打印到日志（INFO 级别）。
    通过 DelayedDeviceHostHandler 异步读取设备状态，不阻塞前向。
    """
    def __init__(
        self,
        *,
        config: CanaryConfig,
        device_state: CanaryDeviceState,
        active_tags: tuple[CanaryLaunchTag, ...],
        outer_step_counter_getter: Callable[[], int],
        sweep_orchestrator: SweepOrchestrator, # 用于获取清扫轮次计数
        d2h_stream: torch.cuda.Stream,
    ) -> None:
        self._config = config
        self._device_state = device_state
        self._active_tags = active_tags
        self._outer_step_counter_getter = outer_step_counter_getter
        self._sweep_orchestrator = sweep_orchestrator
        # 异步 D2H 处理器
        self._handler = DelayedDeviceHostHandler(d2h_stream=d2h_stream)

    def step(self) -> None:
        """每步调用：提交异步读取请求"""
        self._handler.step(
```

```
            compute_on_device=self._compute_on_device,
            postprocess_on_host=self._postprocess_on_host,
        )

    def _compute_on_device(self) -> Optional[dict[str, Any]]:
        """在设备上决定是否读取：间隔 <=0 或非打印步则跳过"""
        period = self._config.stats_print_every_n_steps
        if period <= 0:
            return None
        outer_step_counter = self._outer_step_counter_getter()
        if outer_step_counter == 0 or outer_step_counter % period != 0:
            return None
        device_state = self._device_state
        # 返回需要读取的设备张量（slot_run_counters 之和、violation_write_index）
        return {
            "step": outer_step_counter,
            "slot_sum": device_state.slot_run_counters.sum().view(1),
            "write_index": device_state.violation_log.violation_write_index,
        }

    def _postprocess_on_host(self, host_data: dict[str, Any]) -> None:
        """主机端后处理：打印日志，包含步数、受保护 token 数、清扫轮次、违规数、活跃标签数"""
        logger.info(
            "[canary] step=%d protected_tokens=%d sweep_passes=%d violations=%d "
            "launch_tags_active=%d/%d",
            int(host_data["step"]),
            int(host_data["slot_sum"].item()),
            self._sweep_orchestrator.sweep_passes,
            int(host_data["write_index"].item()),
            len(self._active_tags),
            len(CanaryLaunchTag),
        )
```

评论区精华

该 PR 没有人工 review 评论，仅包含自动 tag-and-rerun-ci 请求和 Gemini quota 警告，无实质性设计讨论。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. 健康检查在检测到内核停滞时抛出 RuntimeError，可能导致服务中断；但这是预期行为用于暴露问题，且可通过设置 SGLANG_KV_CANARY_STATS_PRINT_EVERY_N_STEPS=0 完全禁用（虽然该变量控制统计而非健康检查，健康检查有独立硬编码间隔 100 步，没有开关，但可以通过设置 active_tags 为空绕过）。

2. DelayedDeviceHostHandler 避免阻塞前向，但 D2H 拷贝仍占用 PCIe 带宽和流资源，高并发场景可能影响性能；不过每 100 步才执行一次，开销可接受。
3. 环境变量默认启用，用户可能不清楚新增的日志行和潜在异常，建议在文档或 changelog 中说明。 - 影响：对用户：如果启用，用户会在日志中看到 [canary] 行，出现健康检查失败时会看到 RuntimeError 堆栈。对系统：新增两个每步调用的 step() 方法，但内部大部分步骤跳过（预热期、非检查步），对性能影响极小。对团队：增加了 kv-canary 子系统的可观测性，便于定位问题。 - 风险标记：运行时异常中断，默认启用可能影响用户，D2H 拷贝带宽开销

关联脉络

- 暂无明显关联 PR