PR #26816 完整报告

sgl-project/sglang

Add the KV-canary perturb framework for fault-injection self-tests

合并时间：2026-05-31 09:58

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26816

执行摘要

- 一句话：引入 KV-canary 扰动框架用于故障注入自测试
- 推荐动作：该 PR 是 KV-canary 测试框架的重要基础，可精读以理解设计意图。重点关注 WarmupGate 和配置解析模式，未来扩展其他扰动类型时可以参考。

功能与动机

通过注入可控的 KV 损坏，验证 KV-canary 的校验机制确实能够检测到异常，提升系统的可测试性和可靠性。

实现拆解

1. 配置层(perturb/config.py)：定义 TargetGroupKind 枚举（FULL/SWA）和 PerturbConfig 冻结数据类，通过 from_env 类方法从环境变量 SGLANG_KV_CANARY_PERTURB_TARGET_GROUP 和 SGLANG_KV_CANARY_PERTURB_WARMUP_STEPS 解析。解析函数进行大小写不敏感检查并严格拒绝非法值。
2. 工具模块(perturb/utils.py)：实现 WarmupGate 类，根据全局步数判断是否处于预热窗口（前 N 步不扰动），并在首次进入 / 退出时记录日志。提供 should_run_perturbation 函数检查概率和前置条件，以及 pick_target_group 函数根据目标类型从 buffer groups 中随机选取一个目标。
3. 管理器(perturb/manager.py)：PerturbManager 类聚合配置、buffer groups、预热门控，并预留 perturb 方法（当前为空实现），后续具体扰动子类将重写。
4. 集成：在 canary_manager.py 的 __init__ 中创建 PerturbManager 实例，在 _pre_ops_outside_graph 中调用 perturb，在 attach_radix_cache 中传递 radix cache；在 api.py 的 install_canary 入口创建 PerturbConfig 并传递给 CanaryManager；在 environ.py 注册两个新的环境变量。
5. 测试配套：test_self_unit_perturb.py 覆盖配置解析的各种边界情况（有效值、非法值、缺失环境变量）；runner_test_base.py 添加 make_perturb_config 工厂方法；violation_assert_mixin.py 扩展断言辅助。

关键文件：

- python/sglang/srt/kv_canary/perturb/utils.py（模块 扰动框架；类别 source；类型 dependency-wiring；符号 WarmupGate, init, is_in_warmup,

_log_warmup_disabled_once）：实现了 WarmupGate 类、should_run_perturbation 和 pick_target_group 工具函数，是扰动框架的核心工具模块。

- python/sglang/srt/kv_canary/perturb/config.py（模块 配置；类别 source；类型 dependency-wiring；符号 TargetGroupKind, str, PerturbConfig, from_env）：定义 TargetGroupKind 枚举和 PerturbConfig 数据类，包含从环境变量解析配置的逻辑。
- python/sglang/srt/kv_canary/perturb/manager.py（模块 管理器；类别 source；类型 core-logic；符号 PerturbManager, init, attach_radix_cache, perturb）：PerturbManager 类，协调扰动执行，但目前 perturb 方法是空实现。
- test/registered/kv_canary/test_self_unit_perturb.py（模块 单元测试；类别 test；类型 test-coverage；符号 TestParseTargetGroupKind, test_parse_target_group_kind_accepts_valid_values_case_insensitively, test_parse_target_group_kind_rejects_invalid_value, test_parse_target_group_kind_rejects_missing_or_any）：单元测试验证配置解析的正确性，包括大小写不敏感、非法值拒绝和缺失环境变量处理。
- python/sglang/srt/kv_canary/runner/canary_manager.py（模块 管理器集成；类别 source；类型 dependency-wiring）：集成 PerturbManager 到 CanaryManager，在 pre_ops_outside_graph 中调用 perturb。
- python/sglang/srt/kv_canary/api.py（模块 入口；类别 source；类型 entrypoint）：在 install_canary 入口创建 PerturbConfig 并传递给 CanaryManager。
- python/sglang/srt/environ.py（模块 环境配置；类别 source；类型 core-logic）：添加两个环境变量定义以支持 perturb 配置。
- python/sglang/test/kv_canary/violation_assert_mixin.py（模块 测试辅助；类别 test；类型 test-coverage；符号 assert_sweep_violation_reported）：增加辅助函数以在测试中方便使用 perturb 配置。
- python/sglang/test/kv_canary/runner_test_base.py（模块 测试基类；类别 test；类型 test-coverage；符号 make_perturb_config）：提供 make_perturb_config 工厂方法以便测试创建配置。

关键符号：WarmupGate, should_run_perturbation, pick_target_group, PerturbConfig.from_env, _parse_target_group_kind, PerturbManager.init, PerturbManager.perturb, CanaryManager.init, install_canary

关键源码片段

python/sglang/srt/kv_canary/perturb/utils.py

实现了 WarmupGate 类、should_run_perturbation 和 pick_target_group 工具函数，是扰动框架的核心工具模块。

```
# WarmupGate 控制扰动启用的预热窗口
class WarmupGate:
    def __init__(self, *, config: PerturbConfig, outer_step_counter_getter: Callable[[], int]) ->
    None:
        self._config = config
        self._outer_step_counter_getter = outer_step_counter_getter
```

```
        self._warmup_disable_logged = False
        self._warmup_enable_logged = False

    def is_in_warmup(self) -> bool:
        step = self._outer_step_counter_getter()
        if step < self._config.warmup_steps:
            if not self._warmup_disable_logged:
                logger.info("kv_canary perturb: disabled during warmup window (first %d forward
                steps)", self._config.warmup_steps)
                self._warmup_disable_logged = True
            return True
        if not self._warmup_enable_logged:
            logger.info("kv_canary perturb: enabled after warmup window at step=%d", step)
            self._warmup_enable_logged = True
        return False

# 根据目标类型从 buffer groups 中随机选择一个组
def pick_target_group(*, buffer_groups: tuple[CanaryBufferGroup, ...], target_kind:
TargetGroupKind) -> Optional[CanaryBufferGroup]:
    if target_kind == TargetGroupKind.FULL:
        want = PoolKind.FULL
    elif target_kind == TargetGroupKind.SWA:
        want = PoolKind.SWA
    else:
        raise ValueError(f"Unsupported target_group_kind: {target_kind!r}")
    filtered = [g for g in buffer_groups if g.kind == want]
    if not filtered:
        return None
    return random.choice(filtered)
```

python/sglang/srt/kv_canary/perturb/manager.py

PerturbManager 类，协调扰动执行，但目前 perturb 方法是空实现。

```
# PerturbManager 类
class PerturbManager:
    def __init__(self, *, config: PerturbConfig, buffer_groups: tuple[CanaryBufferGroup, ...],
                 outer_step_counter_getter: Callable[[], int], swa_window_size: int = 0, sweep_
                 interval: int = 0) -> None:
        self._config = config
        self._buffer_groups = buffer_groups
        self._outer_step_counter_getter = outer_step_counter_getter
        self._swa_window_size = swa_window_size
        self._sweep_interval = sweep_interval
        self._radix_cache: Optional["BasePrefixCache"] = None
        self._warmup_gate = WarmupGate(config=config, outer_step_counter_getter=outer_step_
        counter_getter)

    def attach_radix_cache(self, radix_cache: "BasePrefixCache") -> None:
        self._radix_cache = radix_cache
```

```
    def perturb(self, *, maybe_inaccurate_forward_batch: Optional["ForwardBatch"]) -> None:
        pass # 当前为空实现，后续子类将重写
```

评论区精华

本 PR 无 review 评论，由作者自行合并。

- 暂无高价值评论线程

风险与影响

- 风险：当前 perturb 方法为空实现，未来实现可能引入性能开销，需关注 forward 路径的调用频率。环境变量未设置时不会启用扰动，对正常用户无影响。测试配置解析较为完善，降低了配置错误风险。整体影响面有限，主要在于可测试性的提升。
- 影响：对用户：无直接影响，扰动仅在环境变量明确设置时启用。对系统：增加了少量启动开销。对团队：为测试提供了基础设施，未来实现具体扰动类型后能更全面验证 canary。
- 风险标记：核心路径新增空实现可能被忽略，环境变量配置可能导致意外启用

关联脉络

- PR #26817 Add real-data KV verification to the KV-canary: 同一功能线，均为 KV-canary 添加可观测性 / 验证能力
- PR #26818 Add token-id verification to the KV-canary: 同一功能线，增加 token-id 验证，与扰动框架配合使用
- PR #26819 Add the KV-canary perturb modes and PD-disaggregation e2e tests: 后续 PR，添加扰动模式和 PD 拆分测试，扩展本 PR 的框架
- PR #26820 Add a sliding-window-attention divergence reporter for the KV-canary: 同一功能线，SWA divergence 报告与扰动框架互补
- PR #26821 Add periodic KV-canary stats logging and kernel-run-counter health check: 同一功能线，日志和健康检查，增强可观测性