PR #27410 完整报告

sgl-project/sglang

Add kv_canary PP self-test fixture and SWA divergence coverage

合并时间：2026-06-06 09:06

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/27410

执行摘要

- 一句话：为 kv_canary 添加 PP 自测夹具和 SWA divergence 测试
- 推荐动作：值得关注 CanaryPPFixture 基类的设计，它为 PP 测试提供了可复用的服务器参数配置和生命周期管理，为后续更多 PP 场景测试提供了模式参考。建议读者精读新增的扰动测试用例，了解 real-kv-hash 扰动的触发条件与断言方法。

功能与动机

为 kv_canary 功能补充 PP（pipeline parallelism）场景下的端到端自测试，确保分布式推理时 KV 缓存校验的正确性。同时增强 SWA divergence 检测的测试覆盖，避免回归。

实现拆解

1. 新增 PP 测试夹具（python/sglang/test/kv_canary/pp_fixture.py）：定义 CanaryPPFixture 基类，继承 CanaryE2EBase，配置 pp-size=2、disable-cuda-graph、SWA 池参数，并设定 workload_n_batches=2。该夹具为所有 PP 端到端测试提供统一的服务器启动与清理流程。
2. 新增 PP 基线测试（test/registered/kv_canary/test_self_e2e_pp_baseline.py）：TestPPBaselineSwa 使用 CanaryPPFixture，以 CanaryMode.LOG 运行，发送并行请求后断言无 violation，并调用 maybe_assert_swa_divergence_observed。
3. 新增 PP 扰动测试（test/registered/kv_canary/test_self_e2e_pp_perturb.py）：TestPPPerturbSwaSwa 在 setUpClass 中设置扰动环境变量（概率 0.1、目标 SWA 组、无 warmup），测试断言 verify_real_kv_hash 失败被捕获。
4. 扩展 SwaDivergenceLog 功能（python/sglang/srt/kv_canary/runner/swa_divergence.py）：添加 find_all 类方法，用正则一次性提取文本中所有匹配的日志条目。
5. 新增 find_all 单元测试（test/registered/kv_canary/test_self_unit_runner_swa_divergence.py）：TestSwaDivergenceLogFindAll 覆盖顺序、峰值幸存、空文本场景。
6. 重构 divergence 断言（python/sglang/test/kv_canary/e2e_base.py 与 test/registered/kv_canary/test_self_unit_e2e_base.py）：修改 assert_swa_divergence_observed 逻辑，将原有 uses_latest_line 测试替换为 uses_peak_out_of_window 和 checks_verify_lag_on_latest_line，增强准确性。
7. 工具微调（python/sglang/test/kv_canary/utils.py）：少量配置键调整以适应新逻辑。

关键文件：

- test/registered/kv_canary/test_self_e2e_pp_perturb.py（模块 PP 扰动测试；类别 test；类型 test-coverage；符号 TestPPPerturbSwaSwa, setUpClass, test_real_kv_used_perturbation_reports_real_kv_hash_violation）：新增 PP 扰动测试，验证 real-kv-hash 扰动的检测，是本次 PR 的核心测试之一。
- test/registered/kv_canary/test_self_unit_runner_swa_divergence.py（模块 SWA 发散测试；类别 test；类型 test-coverage；符号 TestSwaDivergenceLogFindAll, test_find_all_returns_every_sample_in_order, test_find_all_peak_survives_trailing_zero_sample, test_find_all_returns_empty_list_when_no_lines）：新增 find_all 单元测试，覆盖关键匹配逻辑，确保回归。
- test/registered/kv_canary/test_self_unit_e2e_base.py（模块 E2E 基础测试；类别 test；类型 test-coverage；符号 test_assert_swa_divergence_observed_uses_latest_line, test_assert_swa_divergence_observed_uses_peak_out_of_window, test_assert_swa_divergence_observed_checks_verify_lag_on_latest_line）：重构 divergence 断言测试，新增 out-of-window 峰值和 verify-lag 检查用例。
- python/sglang/test/kv_canary/pp_fixture.py（模块 测试夹具；类别 test；类型 test-coverage；符号 CanaryPPFixture, setUpClass）：新增的 PP 测试夹具基类，所有 PP 端到端测试依赖于此，是测试基础设施的关键。
- test/registered/kv_canary/test_self_e2e_pp_baseline.py（模块 PP 基线测试；类别 test；类型 test-coverage；符号 TestPPBaselineSwa, test_no_violation）：新增 PP 基线测试，验证无扰动时无 violation，与扰动测试形成对比。
- python/sglang/srt/kv_canary/runner/swa_divergence.py（模块 SWA 发散逻辑；类别 source；类型 core-logic；符号 find_all）：添加 find_all 方法，支持批量提取 divergence 日志，是单元测试对应的核心逻辑变更。
- python/sglang/test/kv_canary/e2e_base.py（模块 E2E 测试基类；类别 test；类型 test-coverage）：修改 assert_swa_divergence_observed 方法，支持 out-of-window 峰值和 verify-lag 检查，影响所有端到端测试的 divergence 断言行为。
- python/sglang/test/kv_canary/utils.py（模块 测试工具；类别 test；类型 test-coverage）：微调配置键，辅助测试基础设施适配。

关键符号：setUpClass, test_real_kv_used_perturbation_reports_real_kv_hash_violation, test_no_violation, test_find_all_returns_every_sample_in_order, test_find_all_peak_survives_trailing_zero_sample, test_find_all_returns_empty_list_when_no_lines, find_all, test_assert_swa_divergence_observed_uses_peak_out_of_window, test_assert_swa_divergence_observed_checks_verify_lag_on_latest_line

关键源码片段

test/registered/kv_canary/test_self_e2e_pp_perturb.py

新增 PP 扰动测试，验证 real-kv-hash 扰动的检测，是本次 PR 的核心测试之一。

```
from __future__ import annotations
```

```
import unittest
from typing import ClassVar

from sglang.srt.kv_canary.config import CanaryMode
from sglang.srt.kv_canary.perturb.config import TargetGroupKind
from sglang.test.ci.ci_register import register_cuda_ci
from sglang.test.kv_canary.pp_fixture import CanaryPPFixture

# 注册 CI 配置：2-GPU large 测试，预计 220 秒
register_cuda_ci(est_time=220, stage="extra-a", runner_config="2-gpu-large")


class TestPPPerturbSwaSwa(CanaryPPFixture):
    """测试在 PP + SWA 模式下，real-kv-hash 扰动能否被正确检测。"""

    kv_canary_mode = CanaryMode.LOG
    target_group: ClassVar[TargetGroupKind] = TargetGroupKind.SWA
    extra_server_args = ("--kv-canary-real-data", "partial")

    @classmethod
    def setUpClass(cls) -> None:
        # 设置扰动环境变量：10% 概率、目标 SWA 组、无 warmup
        cls.extra_env = {
            "SGLANG_KV_CANARY_PERTURB_REAL_KV_USED_PROB": "0.1",
            "SGLANG_KV_CANARY_PERTURB_TARGET_GROUP": str(cls.target_group),
            "SGLANG_KV_CANARY_PERTURB_WARMUP_STEPS": "0",
        }
        super().setUpClass()

    def test_real_kv_used_perturbation_reports_real_kv_hash_violation(self) -> None:
        # 发送多批并行请求，然后断言检测到 verify_real_kv_hash 失败
        for _ in range(self.workload_n_batches):
            self.send_parallel_requests()
        self.assert_per_forward_violation_reported(
            fail_reason="verify_real_kv_hash",
            target_group=self.target_group,
        )
        self.maybe_assert_swa_divergence_observed()


if __name__ == "__main__":
    unittest.main()
```

test/registered/kv_canary/test_self_unit_runner_swa_divergence.py

新增 find_all 单元测试，覆盖关键匹配逻辑，确保回归。

```
class TestSwaDivergenceLogFindAll(CustomTestCase):
    """测试 SwaDivergenceLog.find_all 方法。"""

    def test_find_all_returns_every_sample_in_order(self) -> None:
```

```
        # 构造三条递增的日志，检查 find_all 返回顺序与原文本一致
        text = "\n".join(
            SwaDivergenceLog(
                forward_ct=ct,
                verify_full=100 * ct,
                verify_swa=10 * ct,
                swa_full_idx_divergence=ct,
                swa_out_of_window_tokens=0,
            ).format()
            for ct in (20, 40, 60)
        )
        parsed = SwaDivergenceLog.find_all(text)
        self.assertEqual([p.forward_ct for p, _ in parsed], [20, 40, 60])

    def test_find_all_peak_survives_trailing_zero_sample(self) -> None:
        # 中间有一个高 out-of-window 样本，末尾为零，find_all 应保留全部
        text = "\n".join(
            SwaDivergenceLog(
                forward_ct=ct,
                verify_full=1,
                verify_swa=0,
                swa_full_idx_divergence=1,
                swa_out_of_window_tokens=oow,
            ).format()
            for ct, oow in ((20, 0), (40, 4080), (60, 0))
        )
        parsed = SwaDivergenceLog.find_all(text)
        self.assertEqual(max(p.swa_out_of_window_tokens for p, _ in parsed), 4080)
        self.assertEqual(parsed[-1][0].swa_out_of_window_tokens, 0)

    def test_find_all_returns_empty_list_when_no_lines(self) -> None:
        # 无匹配行应返回空列表
        self.assertEqual(SwaDivergenceLog.find_all("nothing here\n"), [])
```

python/sglang/test/kv_canary/pp_fixture.py

新增的 PP 测试夹具基类，所有 PP 端到端测试依赖于此，是测试基础设施的关键。

```
from __future__ import annotations

from typing import ClassVar

from sglang.test.kv_canary.consts import SWA_POOL_SERVER_ARGS
from sglang.test.kv_canary.e2e_base import CanaryE2EBase

# Pipeline parallelism 的规模
PP_SIZE: int = 2


class CanaryPPFixture(CanaryE2EBase):
```

```
    """为 PP 场景提供统一的测试基类，配置 SWA 模式、2 次 batch 和服务器参数。"""

    model_mode: ClassVar[str] = "swa"
    workload_n_batches: ClassVar[int] = 2

    @classmethod
    def setUpClass(cls) -> None:
        # 将 PP 必需的参数与子类 extra_server_args 合并
        cls.extra_server_args = (
            "--pp-size",
            str(PP_SIZE),
            "--disable-cuda-graph",
            *SWA_POOL_SERVER_ARGS,
            *cls.extra_server_args,
        )
        super().setUpClass()
```

评论区精华

该 PR 暂无实质性 review 讨论，仅有一条 bot quota 警告。变更由作者自行合并。

- 暂无高价值评论线程

风险与影响

- 风险：新增加的测试代码不涉及生产路径，风险较低。但修改了 *e2e_base*.py 和 utils.py 中的测试基类方法，可能影响已有的其他 kv_canary 测试用例，需确保向后兼容。此外，find_all 正则的引入若日志格式变化可能导致匹配失败，需同步维护。
- 影响：影响限定在 kv_canary 模块的测试集：新增 2 个端到端测试文件（基线 + 扰动）和 1 个测试基类文件，修改了 3 个现有测试文件，新增 1 个源码方法。对系统性能和用户接口无直接影响。
- 风险标记：测试基类变更，影响现有测试，正则依赖

关联脉络

- 暂无明显关联 PR