PR #2026 完整报告

radixark/miles

dashboard: scrape engines directly by default

合并时间：2026-08-13 01:44

原文链接：http://prhub.com.cn/radixark/miles/pull/2026

执行摘要

- 一句话：修复 dashboard 引擎指标静默缺失，默认抓取改为 DIRECT
- 推荐动作：值得精读，尤其适合了解观测链路问题定位方式。看点有二：一是 PR body 中先在真实 16 节点 run 上实测确认端口与 payload 双重错配，再据此决策默认值，是典型的证据驱动修复；二是把混合测试拆成 auto/ 显式 router/repoint 三组，让模式解析与重指向行为各自可回归，值得在类似配置解析代码中复用。

功能与动机

默认路由是 sglang router，而原有 auto 解析逻辑按 use_miles_router 决定抓取模式，导致引擎指标每轮 404、engine_series 流从未创建。PR body 原话：Engine metrics were silently missing from the dashboard in every run that used the default (sglang) router。并给出两个根因：一是 sglang router 的指标端口是独立 prometheus_port（随机 4000-5000），scraper 只拿到主端口；二是即使指对端口，payload 也只有网关自身的 smg_* 计数，没有 worker_addr 与 sglang: 系列。失败又是静默的：hooks.register_router 以 fire-and-forget 方式调用 remote，404 只出现在 collector actor 内部日志，所以需要让 auto 默认直接抓各引擎 /metrics。

实现拆解

1. 修正模式解析（miles/dashboard/collector.py）：set_router 删除 use_miles_router 分支，auto 恒定走 ScrapeMode.DIRECT；只有显式 --dashboard-sglang-scrape-mode router 才使用 ROUTER。这样默认路径直接抓各引擎 /metrics，绕开端口与 payload 双重错配。
2. 同步注册链路（miles/dashboard/hooks.py）：register_router 调用 set_router.remote(addr) 时不再传 use_miles_router，与新签名对齐；args.use_miles_router 仍由路由选择主逻辑使用，调用方行为不变。
3. 更新参数文档（miles/dashboard/args.py）：--dashboard-sglang-scrape-mode 帮助文本改为 auto 抓各引擎 /metrics、router 抓 {router}/engine_metrics，避免继续误导。
4. 测试拆分与适配：tests/fast/dashboard/test_collector.py 把原 test_scraper_mode_resolution_and_repoint 拆为 test_auto_scrape_mode_is_direct、test_explicit_router_scrape_mode_is_honoured、test_scraper_repoint，新增 _router_mode_collector 辅助；tests/fast/dashboard/test_hooks.py 与 tests/fast/dashboard/test_core_integration.py 同步移除假参数，断言 set_router 不再携

带 use_miles_router。

关键文件：

- miles/dashboard/collector.py（模块 采集器；类别 source；类型 core-logic；符号 set_router）：核心修复：set_router 删除 use_miles_router 分支，auto 固定解析为 DIRECT，使默认 sglang router 下的引擎指标可正常抓取。
- miles/dashboard/hooks.py（模块 钩子；类别 source；类型 core-logic）：register_router 不再向 set_router 传递 use_miles_router，API 简化的同步配套。
- miles/dashboard/args.py（模块 参数解析；类别 source；类型 configuration）：同步更新 --dashboard-sglang-scrape-mode 帮助文本，反映 auto 现在抓取各引擎 /metrics。
- tests/fast/dashboard/test_collector.py（模块 采集器；类别 test；类型 test-coverage；符号 test_scraper_mode_resolution_and_repoint, _router_mode_collector, test_auto_scrape_mode_is_direct, test_explicit_router_scrape_mode_is_honoured）：将原混合测试拆成 auto 解析、显式 router 模式、repoint 三组测试，并新增 _router_mode_collector 辅助，防行为回归。
- tests/fast/dashboard/test_core_integration.py（模块 集成测试；类别 test；类型 test-coverage）：从 fake args 中移除 use_miles_router 字段以匹配新 API。
- tests/fast/dashboard/test_hooks.py（模块 钩子；类别 test；类型 test-coverage）：更新 _router_args 与断言以适配去掉 use_miles_router 的 set_router 调用。

关键符号：set_router, register_router

关键源码片段

miles/dashboard/collector.py

核心修复：set_router 删除 use_miles_router 分支，auto 固定解析为 DIRECT，使默认 sglang router 下的引擎指标可正常抓取。

```
def set_router(self, router_addr: str) -> None:
    """注册 sglang router 并启动（或重新指向）scraper。"""
    # auto 模式固定走 DIRECT：sglang router 自身不提供聚合的 engine_metrics，
    # ROUTER 模式必然 404；即使改打 prometheus 端口，也只有网关自己的 smg_* 计数。
    mode = ScrapeMode.DIRECT if self.config.scrape_mode == 'auto' else ScrapeMode(self.
    config.scrape_mode)
    # 停 scraper 时不能持有锁：其线程可能正阻塞在 _append sink 的同一把锁上（死锁）
    with self._lock:
        previous = self._scraper
        if previous is not None and previous.router_addr == router_addr and previous.mode ==
        mode:
            return # 相同 router 与模式：保持现有 scraper，不重建
        self._scraper = None
    if previous is not None:
        previous.stop()
    kwargs = dict(
        mode=mode,
        router_addr=router_addr,
```

```
            engine_addrs=self._current_engine_addrs,
            interval=self.config.scrape_interval_seconds,
            whitelist=self.config.metric_whitelist,
        )
        if self._scraper_http_get is not None:
            kwargs['http_get'] = self._scraper_http_get
        scraper = SglangScraper(self._append, **kwargs)
        with self._lock:
            self._scraper = scraper
        scraper.start()
        logger.info('dashboard scraper started in %s mode against %s', mode, router_addr)
```

tests/fast/dashboard/test_collector.py

将原混合测试拆成 auto 解析、显式 router 模式、repoint 三组测试，并新增 _router_mode_collector 辅助，防行为回归。

```
def _router_mode_collector(tmp_path):
    # 显式 router 模式：给真正会聚合引擎指标的网关使用
    config = CollectorConfig(
        dashboard_dir=str(tmp_path / 'dashboard'),
        run_name='collector-test',
        start_ts=1.0,
        scrape_mode='router',
    )
    return make_collector(tmp_path, config=config, scraper_http_get=lambda url, timeout:
    ROUTER_FIXTURE)


def test_auto_scrape_mode_is_direct(tmp_path):
    # 默认 auto 必须解析为 DIRECT：sglang router 不提供 engine_metrics 端点
    collector = make_collector(tmp_path, scraper_http_get=lambda url, timeout: ROUTER_
    FIXTURE)
    collector.set_router('http://router:3000')
    assert collector._scraper.mode == ScrapeMode.DIRECT
    collector.shutdown()


def test_explicit_router_scrape_mode_is_honoured(tmp_path):
    collector = _router_mode_collector(tmp_path)
    collector.set_router('http://router:3000')
    assert collector._scraper.mode == ScrapeMode.ROUTER
    collector.shutdown()


def test_scraper_repoint(tmp_path):
    collector = _router_mode_collector(tmp_path)
    collector.set_router('http://router:3000')
    first = collector._scraper
```

```
    collector.set_router('http://router:3000') # no-op：相同地址与模式不重建
    assert collector._scraper is first

    collector.update_topology(TopologySnapshot(ts=1.0, engines=[_engine('http://e:1')]))
    collector.set_router('http://router2:3000') # router 重启后重新指向
    assert collector._scraper is not first
    assert first._stop_event.is_set()
    # 先有 actor 注册的引擎，再保留上一轮 scrape 记忆的外部引擎
    assert collector._scraper.engine_addrs() == [
        'http://e:1',
        'http://10.0.0.1:15000',
        'http://10.0.0.1:15004',
    ]
    collector.shutdown()
```

评论区精华

本 PR 没有实质 review 讨论：唯一审核人 Zhichenzzz 直接 APPROVED，review body 为空；关联 Issue 仅有一条 Gemini Code Assist 停用公告。信息量集中在 PR body 的根因诊断：Wrong port（sglang router 指标端口是独立 prometheus_port，scraper 始终只打主端口，每 tick 404）与 Wrong payload（实测 16 节点 GLM-5.2：prometheus 端口 /engine_metrics 返回 200，但只有 smg_* 网关计数，无 worker_addr 与 sglang: 系列）——这两点直接决定了默认 DIRECT 的修复方向，且未引发任何争议。

- 暂无高价值评论线程

风险与影响

- 风险：风险集中在四处。其一，默认行为变更：未显式指定 --dashboard-sglang-scrape-mode 的存量运行会从 ROUTER 自动切换为 DIRECT，若环境依赖 sgl-model-gateway 聚合语义需显式传 router 模式。其二，抓取压力变化：DIRECT 从打一个 router 端点变为打每个引擎的 /metrics，16 节点下每 2 秒一轮的请求量成 N 倍增长，需观察引擎 prometheus 端点的负载。其三，API 兼容：collector.py 的 set_router 移除了 use_miles_router 关键字参数，仓库内调用点已同步，但外部直接调用方会收到 TypeError。其四，测试覆盖：fast 测试用 fixture 模拟 /metrics，没有覆盖真实 sglang router 双端口与网关聚合的 e2e 场景，端口漂移类问题仍可能回归。
- 影响：影响范围集中在 dashboard 可观测链路。对用户：开启 --use-miles-dashboard 且使用默认 sglang router 的运行将首次出现 engine_series 指标流，引擎视图不再缺数据。对系统：每 tick 404 的无效请求消失，指标真实落盘，collector 行为与参数文档语义对齐。对团队：dashboard 注册链路中 use_miles_router 相关分支收窄，miles router 弃用后的死代码面进一步缩小，为后续清理铺路。
- 风险标记：默认行为变更，抓取端点扩展，API 兼容性，e2e 覆盖缺失

关联脉络

- PR #2353 dashboard: report model FLOPs utilization: 同为 dashboard 引擎侧观测能力补全，改动 dashboard/advisory.py、train_metric_utils.py 等，与本 PR 共用 dashboard 前后端链路。
- PR #2024 dashboard: read open phase markers regardless of age: 同为 dashboard 数据可见性修复，改动 miles/dashboard/store.py，与本次 engine_series 流缺失同属看板空窗问题。