

PR #2027 完整报告

radixark/miles

dashboard: advisory v2 — run-health alarms before config tuning

合并时间: 2026-08-13 02:27

原文链接: <http://prhub.com.cn/radixark/miles/pull/2027>

执行摘要

- 一句话: advisory v2 以健康告警优先, 门控并抑制误导性调优建议
- 推荐动作: 值得精读。核心设计决策值得学习: 按“主张”分层 + 告警门控调优; stall watchdog 用自身中位数基线且对“常开相位”不作主张; counter 差分容忍引擎重启; 缺失数据与退化组的语义区分。建议读者关注评审中 DumpStillWriting 边界是否跟进, 以及 follow-up 的 per-dp-rank 标签粒度问题。

功能与动机

PR body 的 Motivation 记录了一次真实事故: 744B agentic 运行中 rollout 管线崩溃 (dp-rank 饥饿 → client-timeout 风暴 → 整组丢弃 → backfill 反馈循环), 训练侧在 open train_wait 停留 11.5 小时、35% 引擎请求被 client 中止, 而 v1 面板只给出 "Peak concurrency (14) stayed under 30% of --sglang-max-running-requests (256); consider lowering it"。作者核心判断是低利用率在饥饿场景下是症状而非余量——降低上限会把真正的瓶颈锁死。因此 v2 按“规则所主张的内容”排序, 并用健康门控保证同一观察在健康与饥饿运行上不会导出相反结论。

实现拆解

1. 入口重构与规则分层: `miles/dashboard/advisory.py` 将 `compute_advisories` 重写为“告警优先、调优门控”结构: 先收集 `_stall_advisories` (critical 管线停滞)、`_engine_alarm_advisories` (warning 引擎侧)、`_rollout_advisories` (warning dump 侧), 任一告警存在即短路返回, 只有全空时才进入 `_tuning_advisories` 输出 v1 的 info 级调优建议。合并 main 时与 #2353 的 MFU 规则、#2023 的 dp-imbalance 规则冲突, 最终 MFU 作为调优层种子 (无 ENGINE_SERIES 时仍可触发), dp-imbalance 保留在调优层尾部。
2. Stall watchdog (停滞看门狗): `_stall_advisories` 基于新增的 `MetricStore.phase_events()` (返回未展开的原始相位事件) 计算每个 open 相位的年龄, 与同名相位历史中位数对比: 超过 `STALL_FACTOR=6x`、不低于 `STALL_MIN_AGE_S=1800s`、且至少 3 个 closed 样本才判 critical。关键边界: 设计上永不关闭的相位 (fully-async rollout) 没有基线、不作主张; 用 `meta.start_ts` 过滤上一次崩溃 attempt 遗留的 open marker, 防止断点续跑后永久锁死 critical; 带配对 closed 事件的 open marker 不算停滞。train_wait 相位额外附加“训练被 rollout 数据饿死”的提示。

3. 引擎与 rollout 告警: `_engine_alarm_advisories` 从累计计数器 `sglang_num_requests_total / slang_num_aborted_requests_total` 计算最近 30 分钟 abort 比, `_counter_delta` 按区间累加正增量以容忍引擎重启导致计数重置; 告警文本展开 group-discard 放大效应 (`n_samples_per_prompt` 整组丢弃 + backfill 重提交)。
KV-pool-bound 签名 (token 使用率 >90% 且峰值并发 <50% 上限) 输出“提高 `--sglang-mem-fraction-static`, 不要降低请求上限”的 warning。`_rollout_advisories` 检查最新训练 rollout 的 zero-std 组比例 ($\geq 50\%$ 告警, 全零奖励额外提示系统性失败) 与截断比例 ($\geq 25\%$ 告警, 引用 `--rollout-max-response-len`)。
4. 支撑修复与配套: `DumpReader.groups()` 的 `zero_std` 判定加入 `reward_mean.is_not_null()` 条件, 区分“训练 dump 缺失导致的 null 奖励”与真实退化组; `miles/dashboard/args.py` 将 `rollout_max_response_len` 持久化到 meta args 快照; `miles/dashboard/server.py` 的 `/api/advisory` 现在把 `DumpReader` 传入 `compute_advisories`; 前端 `views_timeline.js` 面板更名“Run advisory”, info 淡显、critical 加粗。
5. 测试配套: `test_advisory.py` 新增 13 个用例 (停滞判定、双子 marker、跨 attempt、无基线相位、abort 风暴及引擎重启、风暴消退、低流量窗口、KV 绑定、zero-std、全零奖励、null 奖励、截断引用), 用 `_StubReader` 隔离 dump 侧规则; `test_dump_reader_views.py` 新增 null-reward 用例; `test_server.py` 更新为预期截断 warning。PR body 声称 60 项 dashboard 测试通过。

关键文件:

- `miles/dashboard/advisory.py` (模块 告警引擎; 类别 source; 类型 core-logic; 符号 `compute_advisories`, `_fmt_duration`, `_stall_advisories`, `_counter_delta`): advisory v2 的全部核心逻辑: `compute_advisories` 分层重构、stall watchdog、abort 风暴与 KV-pool-bound 告警、rollout 侧规则、调优门控。
- `tests/fast/dashboard/test_advisory.py` (模块 告警测试; 类别 test; 类型 test-coverage; 符号 `_phase`, `_closed_steps`, `_StubReader`, `test_stalled_open_phase_is_critical_and_suppresses_tuning`): 13 个新增用例覆盖 v2 全部规则分支与边界 (停滞、双子 marker、跨 attempt、abort 风暴重启、KV 绑定、null 奖励等), 是规则语义的权威文档。
- `miles/dashboard/dump_reader.py` (模块 数据读取; 类别 source; 类型 bugfix; 符号 `groups`): 修复 `zero_std` 误判: null 奖励 (训练 dump 缺失 / 未同步) 不再被当作退化组, 这是 rollout 视图出现虚假 '8/8 zero-std groups' 的根因。
- `miles/dashboard/store.py` (模块 存储层; 类别 source; 类型 core-logic; 符号 `phase_events`): 新增 `phase_events()` 公共访问器, 暴露未展开的原始相位事件 (含 open marker), stall watchdog 依赖它。
- `miles/dashboard/server.py` (模块 服务 endpoint; 类别 source; 类型 dependency-wiring; 符号 `advisory`): `/api/advisory` 端点把 `DumpReader` 传入 `compute_advisories`, 使 rollout 侧规则生效; 是 v2 功能的接线点。
- `miles/dashboard/static/views_timeline.js` (模块 前端面板; 类别 source; 类型 frontend): 面板从 'Config advisory' 更名 'Run advisory' 并按级别渲染 (info 淡显、critical 加粗), 是用户可见的变更入口。

- miles/dashboard/args.py (模块 参数层; 类别 source; 类型 configuration) : 将 rollout_max_response_len 持久化进 meta args 快照, 截断告警才能引用真实配置值。
- tests/fast/dashboard/test_dump_reader_views.py (模块 读取测试; 类别 test; 类型 test-coverage; 符号 test_groups_null_rewards_are_not_zero_std) : 新增 test_groups_null_rewards_are_not_zero_std, 锁死缺失数据与退化组的语义。
- tests/fast/dashboard/test_server.py (模块 服务测试; 类别 test; 类型 test-coverage; 符号 test_advisory_endpoint) : advisory 端点断言从“无告警”更新为“截断 warning”, 验证 reader 接线。

关键符号: compute_advisories, _stall_advisories, _engine_alarm_advisories, _rollout_advisories, _tuning_advisories, _counter_delta, MetricStore.phase_events, DumpReader.groups

关键源码片段

miles/dashboard/advisory.py

advisory v2 的全部核心逻辑: compute_advisories 分层重构、stall watchdog、abort 风暴与 KV-pool-bound 告警、rollout 侧规则、调优门控。

advisory.py —— v2 入口: 先算告警, 有告警直接返回; 调优建议只在健康运行时输出。

同一观察 (低并发、低缓存命中) 在健康与饥饿运行上含义相反, 必须被告警门控。

```
def compute_advisories(
    store: MetricStore,
    reader: DumpReader | None = None,
    *,
    t0: float | None = None,
    t1: float | None = None,
    mfu: dict | None = None,
    low_mfu: float = DEFAULT_LOW_MFU,
) -> list[Advisory]:
    args = store.meta.args if store.meta else {}
    alarms = _stall_advisories(store)
    alarms += _engine_alarm_advisories(store, args, t0, t1)
    if reader is not None:
        alarms += _rollout_advisories(reader, args)
    if alarms:
        return alarms # 有活动告警时, 调优建议会指向错误方向, 直接抑制
    return _tuning_advisories(store, args, t0, t1, mfu=mfu, low_mfu=low_mfu)
```

```
def _stall_advisories(store: MetricStore) -> list[Advisory]:
```

```
    """Open 相位超过自身历史中位数 6 倍 (且至少 30 分钟) 视为停滞。
```

```
    以全 run 最新数据时间为判断基准, 而非查询窗口: 停滞是当前状态的陈述,
```

```
    open marker 可能比用户查看的任意窗口都旧得多。"""
```

```
    time_range = store.time_range()
```

```
    if time_range is None:
```

```
        return []
```

```

edge = time_range[1]
# 断点续跑会 append 到同一 dump 目录, crash attempt 留下的 open marker
# 永不关闭; 用本次 attempt 时钟去量它们会永久锁死 critical。meta.json 每
# 次 attempt 重写, start_ts 是天然边界, 同时把上次 attempt 的 closed 时长
# 挡在 median 基线之外。
attempt_start = store.meta.start_ts if store.meta else float("-inf")
events = [event for event in store.phase_events() if event.t0 >= attempt_start]
# 有配对 closed 事件的 open marker 说明相位已正常关闭 (乱序写入), 不算停滞
closed_keys = {(e.node, e.rank, e.name, e.t0) for e in events if not e.open}
durations: dict[str, list[float]] = {}
open_ages: dict[str, list[float]] = {}
for event in events:
    if event.open:
        if (event.node, event.rank, event.name, event.t0) not in closed_keys:
            open_ages.setdefault(event.name, []).append(edge - event.t0)
    else:
        durations.setdefault(event.name, []).append(event.t1 - event.t0)

out = []
for name, ages in sorted(open_ages.items()):
    closed = durations.get(name, [])
    # fully-async 的 rollout 相位设计上永不关闭: 没有 closed 基线就没有主张
    if len(closed) < STALL_MIN_CLOSED:
        continue
    typical = median(closed)
    age = max(ages)
    if age <= max(STALL_FACTOR * typical, STALL_MIN_AGE_S):
        continue
    message = (
        f"Phase '{name}' has been open for {_fmt_duration(age)} on {len(ages)} rank(s) — "
        f"{age / typical:.0f}x its median duration ({_fmt_duration(typical)}); "
        "the pipeline is stalled, not slow"
    )
    if name == "train_wait":
        message += " (training is starved of rollout data — check the data buffer and engine health)"
    out.append(Advisory(level="critical", message=message))
return out

```

```

def _counter_delta(series: list[dict], *, since: float = float("-inf")) -> float:
    """累计计数器在窗口内的总增量。

```

按区间逐段累加并丢弃负增量: 引擎在同一地址重启会中途重置计数器, last-minus-first 会把重启前后的历史一并抹掉 (与 MetricStore 派生序列同一条重置规则); since 只保留其后结束的区间。"""

```

total = 0.0
for one in series:
    for ts, (before, after) in zip(one["ts"][1:], pairwise(one["value"]), strict=True):

```

```

    if ts >= since:
        total += max(after - before, 0.0)
    return total

```

miles/dashboard/dump_reader.py

修复 zero_std 误判：null 奖励（训练 dump 缺失 / 未同步）不再被当作退化组，这是 rollout 视图出现虚假 '8/8 zero-std groups' 的根因。

```

# dump_reader.py —— zero_std 语义修复：缺失数据不是退化组。
def groups(self, rollout_id: int, *, evaluation: bool = False) -> pl.DataFrame:
    """Per-GRPO-group 聚合；zero_std 标记退化组（组内奖励全同，优势消失）。"""
    reward_column = "reward" if evaluation else "raw_reward"
    return (
        self.summary(rollout_id, evaluation=evaluation)
        .group_by("group_index")
        .agg(
            n=pl.len(),
            reward_mean=pl.col(reward_column).mean(),
            reward_std=pl.col(reward_column).std(),
            response_length_mean=pl.col("response_length").mean(),
            truncated_frac=pl.col("truncated").cast(pl.Float64).mean(),
        )
        # reward_mean 为 null 表示训练 dump 缺失 / 未同步，是缺失数据而非退化组；
        # reward_mean 有值但 std 为 null 表示单样本组，那才是真正退化
        .with_columns(
            zero_std=pl.col("reward_mean").is_not_null() & (pl.col("reward_std").fill_null(0.0) <= 1e-12)
        )
        .sort("group_index")
    )

```

评论区精华

唯一一条实质 review 评论来自合并者 Zhichenzzz ([miles/dashboard/advisory.py:72](#))：
[_rollout_advisories](#) 使用 `train_ids[-1]`，最新 rollout 最可能仍在写盘，`summary()` 会抛 `DumpStillWriting` 并被 server 映射为 503；前端 follow 模式以 4 请求 /4.5s 的成本重试后 `catch {}` 静默保留旧列表——面板冻结在过期快照，而丢失的恰好是只在 live run 上重要的 stall/abort 告警。他建议用 `try/except (DumpStillWriting, FileNotFoundError, KeyError)` 隔离 dump 侧规则，并顺带指出 `groups()` 内部已调 `summary()`，同一 parquet 每请求读两遍、可传一次减半。从最终 head 代码看，`compute_advisories` 仍为裸调用 `_rollout_advisories(reader, args)`，未见该包裹，此建议尚未落入实现。另有流程性评论询问 2022-2027 各 dashboard PR 的合入顺序，无实质技术内容。

- 最新 rollout 写盘中的 `DumpStillWriting` 会拖垮 liveness 告警 (correctness)：从最终 head 代码看 `compute_advisories` 仍为裸调用 `_rollout_advisories(reader, args)`，未见 `try/except` 包裹，建议未落入实现，属未解决边界。
- 2022-2027 系列 dashboard PR 的合入顺序 (question)：未在评论区展开；从提交历史看各 PR 最终陆续合入 main，且 v2 在合并中吸收 #2353 与 #2023 的冲突。

风险与影响

- 风险：正确性风险：_rollout_advisories 的 DumpStillWriting 异常会拖垮整个告警链——不仅 503，还会让前端静默保留旧快照，且丢失的是最重要的 liveness 告警；该路径目前无测试覆盖。性能风险：follow 模式下 /api/advisory 每 tick 重复读同一 parquet (groups() → summary() 双层)，异常时额外 3 次重试，大 dump 下磁盘 I/O 放大。兼容性：Advisory.level 新增 critical 枚举，API 响应结构不变；面板对 critical 仅加粗，视觉区分度有限。阈值风险：STALL_FACTOR、ABORT_RATIO_WARN、ZERO_STD_FRAC_WARN 等均为硬编码，v1 已声明“阈值需依真实运行调参”，新规则在真实运行覆盖前存在误报 / 漏报空间。已知缺口：dp-rank 队列不平衡告警因 scraper 标签粒度不足 (collector 将引擎标签降为 engine_type) 无法实现，作者自述留作 follow-up。
- 影响：用户影响：训练 / 推理工程师在面板上获得三层告警，能第一时间区分“管线停滞”与“计算慢”，避免按误导性调优建议（如降低请求上限）锁死真正瓶颈。系统影响：dashboard 服务新增 dump 侧读取路径与一个 meta 参数键 (rollout_max_response_len)，无破坏性 API 变更。团队影响：advisory 规则按层级组织、阈值集中在文件顶部常量区，后续新增规则需明确归层并补短路测试；评审指出的 DumpStillWriting 边界是合入后应优先跟进的项目。
- 风险标记：DumpStillWriting 未隔离（告警面板可能静默冻结），硬编码阈值需真实运行调参，follow 模式重复读盘 + 重试开销，per-dp-rank 粒度缺失，dp 不平衡告警缺位

关联脉络

- PR #2353 dashboard: report model FLOPs utilization: 合并 main 时与 v2 在 compute_advisories 中冲突：MFU 规则被归入 _tuning_advisories 层，并保持无 ENGINE_SERIES 时仍可触发。
- PR #2026 dashboard: scrape engines directly by default: 同一 dashboard/observability 功能线，为本 PR 的引擎 counter 数据可靠性打基础。
- PR #2024 dashboard: read open phase markers regardless of age: 修复相位 marker 读取边界，与 stall watchdog 共享相位事件数据路径。