PR #2024 完整报告

radixark/miles

dashboard: read open phase markers regardless of age

合并时间：2026-08-12 12:27

原文链接：http://prhub.com.cn/radixark/miles/pull/2024

执行摘要

- 一句话：移除相位读取下界，修复停滞 run 视图空白
- 推荐动作：值得精读（改动很短）：这是一个小而精准的 bugfix，展示了数据流分区策略（按 END hour / START hour 分区）与查询窗口假设耦合导致的数据可见性问题。值得关注的设计决策：对低速率流主动放弃时间下界换取正确性，并用量化成本（约 40KB/h）支撑该决策。后续在 phases 流上做其他窗口查询时应延续 ' 无下界 ' 约定。

功能与动机

compute-utilization 视图在 run 停滞时显示 waiting for phase data… 且 lane 时间线空白：一个 744B run 因 rollout 请求超时停滞，在最后一个训练步完成后停留于单个 open train_wait 达 11.5h。phase 流数据完好（open marker 在磁盘上），但 _phase_events 用 t0 - MAX_WINDOW_S 作为下界、假设 open marker 至多 4h 旧；而 open marker 按 START hour 分区，一旦超龄即落出所有后续窗口读取。正如 PR body 所说：'This is exactly the situation where the phase view matters most: a fleet-wide stall rendered as no data instead of 100% train_wait.'

实现拆解

1. 定位根因：miles/dashboard/store.py 中 _phase_events 构造读取窗口时，lower = t0 - self.MAX_WINDOW_S 隐含假设 open phase marker 最老 4h；但 open marker 按其 START hour 分区，停滞 11.5h 后已超过该下界，导致每次窗口读取都漏掉该事件。
2. 修改读取逻辑：将分区读取改为 window(None, upper)，只保留上界 t1 + MAX_WINDOW_S；上界同时覆盖 closed phase 的 END hour 分区与 open marker 的 START hour 分区，并删除原注释中以 MAX_WINDOW_S 为前提的说明（第二次 commit 移除残留注释）。
3. 成本评估与验证：phases 流低速率（约 40KB/h，15h 16 节点运行仅 563KB），读取到上界的全部分区成本可忽略；gpu_util、engine_series 等高速流不受影响。作者在停滞 run 的 dump 上验证尾部 4h fleet 组成从空（waiting for phase data…）恢复为 {train_wait: 0.50, rollout: 0.50}，与实际停滞匹配。
4. 测试与配套：未新增测试文件，依赖既有的 tests/fast/dashboard/test_store.py（14 项通过）；无配置、schema 或部署配套改动。

关键文件：

- miles/dashboard/store.py（模块 仪表盘；类别 source；类型 core-logic；符号 _phase_events）：唯一变更文件，_phase_events 移除 phases 分区读取的下界，修复长时间 open marker 读不到的问题

关键符号：_phase_events

关键源码片段

miles/dashboard/store.py

唯一变更文件，_phase_events 移除 phases 分区读取的下界，修复长时间 open marker 读不到的问题

```
def _phase_events(self, t0: float | None, t1: float | None) -> list[PhaseEvent]:
    # 只保留上界：closed phase 按 END hour 分区，查询窗口向前留 MAX_WINDOW_S 的松弛量；
    # open marker 按 START hour 分区，可能远早于窗口起点（例如停滞 11.5h 的 train_wait），
    # 因此不再设置下界，保证任何年龄的 open marker 都能被读回。
    upper = None if t1 is None else t1 + self.MAX_WINDOW_S
    return self._readers[Stream.PHASES].window(None, upper)
```

评论区精华

PR 无实质 review 讨论：唯一评论来自 gemini-code-assist[bot]，内容为通知其 GitHub consumer 版本已 sunset、代码审查活动已停止；人工评审 Zhichenzzz 直接 APPROVED 且未留评论，说明改动小而清晰。

- 自动化代码审查服务停止通知 (other): 无实质审查内容，不影响 PR 评审结论；人工评审 Zhichenzzz 已 APPROVED。

风险与影响

- 风险：主要风险集中在读取范围扩大与回归覆盖：_phase_events 现在会读取窗口上界之前全部分区，对长时间运行的 dashboard 实例，phases 分区累积后读取量线性增长，但该流低速率（约 40KB/h），实际成本可忽略；行为上依赖 reader 的 window 按时间戳过滤语义，既有的 14 项测试通过，但缺少针对 ' 超过 4h 的 open marker' 的直接回归测试，后续改动可能重新引入同类假设；不影响 gpu_util、engine_series 等其他流，也无 API 或 schema 变更。
- 影响：影响范围集中在 dashboard 的 compute-utilization 相位视图：停滞 run 现在能显示真实相位（如 100% train_wait）而非空白，所有使用 dashboard 的用户受益。改动仅涉及 phases 流读取路径，gpu_util、engine_series 等高速流不受影响，无 API、schema 或配置变更，兼容性风险低。对团队而言，该修复补齐了渲染端 #1965 对应的读端缺口。
- 风险标记：相位读取语义变更，缺少直接回归测试

关联脉络

- PR #1965 dashboard: render open phase through to the data edge: PR body 中明确引用：渲染端已支持将 open phase 绘制到数据边缘，本 PR 补齐读端交付，二者构成完整闭环。

- PR #2359 docs: dashboard advanced features and example visualization: 同为 dashboard 模块近期变更，反映 dashboard 功能持续演进。