

# PR #39111 完整报告

sgl-project/sglang

[Router] Add a KV storage-tier coverage row to the Grafana dashboard (4/4)

合并时间: 2026-09-16 08:09

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/39111>

## 执行摘要

- 一句话: 为 sgl-router 监控看板新增 KV 分层存储覆盖率三面板
- 推荐动作: 这个 PR 最值得精读的是 PR body 而非 diff: 其中关于 "跨 namespace 相除会得到一个漂亮的错误数字 (0.54 vs 1.00)"、"检查查询的值而非仅仅检查返回行"、"max 会掩盖 quietest replica 失联" 的监控设计论证具有普遍借鉴价值, 适合负责可观测性面板与告警设计的工程师参考。建议与 #39109、#39110 按 stack 顺序一起阅读, 了解指标从产生、测试到面板化呈现的完整链路。

## 功能与动机

PR body 指出: 此前 dashboard 无法回答缓存感知路由退化时的核心问题「router 对 fleet KV cache 的视图是否与引擎实际持有一致, 按层 (tier) 看」。这个缺口曾掩盖一个持续数月的真实缺陷——tree 丢弃了 storage-medium 标签, 导致 pod 的 host 层副本对路由不可见; 从看板上看到低 cache-hit rate 与普通的引擎侧淘汰无法区分, 因为没有任何面板同时画出 router 的认知与引擎的实际持有量。作者在正文中还特别强调: 无 namespace 作用域的覆盖率表达式在真实 fleet 上会读出 0.54 而非正确的 1.00, 属于「永久、看似合理的 P1 误报」。

## 实现拆解

本 PR 仅修改 1 个文件: `experimental/sgl-router/monitoring/grafana-dashboard.json`, 整体为纯新增配置, 按以下步骤拆解:

1. 新增行容器与布局: 在 dashboard 末尾 (stale requests 面板之后) 追加 row 类型面板 "KV tree vs engine (storage-tier coverage)" (id 28, gridPos y=58), 随后挂载 3 个子面板 (id 29-31), 依次排布在 y=59 与 y=67 两行。未改动任何既有面板、模板变量或 refresh 配置, diff 保持零删除。
2. 面板 1: KV event blocks by medium (id 29): 对 `sgl_router_kv_event_blocks_total` 分别执行 `max by (namespace, event, medium)` 与 `min by (namespace, event, medium)` 双查询。设计意图是同时绘制最忙与最安静的 replica——所有 replica 都接收完整事件流, 若 quietest 曲线为 0 而 busiest 正常, 说明某个 replica 丢失了订阅、正在盲路由, 单看 max 会掩盖该故障。legend 分别标注 `{{namespace}} {{event}}/{{medium}}` 与 `(quietest replica)`。
3. 面板 2: KV tree blocks by tier (fleet) (id 30): 表达式为 `sum by (tier) (max by (worker_url, dp_rank, tier) (sgl_router_kv_tree_blocks))`, 先按 worker 取 max 以去重多

副本，再按 tier 求和。因为是绝对计数（非比率），跨 worker 求和是合法的；host 曲线为 0 说明路由完全看不到 L2 层。

4. 面板 3: Tree coverage of engine host tier (id 31)：核心覆盖率面板，分子为 router 树中 host 层块数 × `sgl_router_kv_block_size`（块大小），分母为引擎实际暴露的 `sglang:hiocache_host_used_tokens{tp_rank="0"}`，unit 设为 percentunit、min 钳制为 0。分子分母两侧都按 namespace 分组后相除，避免跨部署、跨块大小相除；`on(namespace) group_left()` 保证 block size 与树计数按 namespace 对齐。
5. 验证配套：无自动化测试（dashboard JSON 无法单元测试）。作者将所有表达式替换 `$vars` 后 POST 到存有真实序列的 Prometheus，检查的是「值」而非仅仅「返回了行」——返回行检查正是不带 namespace 作用域的 0.54 能被误判为健康的原因；同时验证了文件重新序列化后 panel id 唯一、gridPos 堆叠正确。

关键文件：

- `experimental/sgl-router/monitoring/grafana-dashboard.json`（模块 监控面板；类别 config；类型 configuration）：唯一变更文件，新增 "KV tree vs engine (storage-tier coverage)" 行及 3 个面板，是整个 HiCache 路由可观测性系列（2/4 指标 + 3/4 e2e + 4/4 面板）的最终呈现层。全部 PromQL 均按 namespace 作用域设计，核心覆盖率表达式通过 `on(namespace) group_left()` 对齐 router 树计数与引擎实际持有量。

关键符号：未识别

## 关键源码片段

### `experimental/sgl-router/monitoring/grafana-dashboard.json`

唯一变更文件，新增 "KV tree vs engine (storage-tier coverage)" 行及 3 个面板，是整个 HiCache 路由可观测性系列（2/4 指标 + 3/4 e2e + 4/4 面板）的最终呈现层。全部 PromQL 均按 namespace 作用域设计，核心覆盖率表达式通过 `on(namespace) group_left()` 对齐 router 树计数与引擎实际持有量。

以下为该行 3 个面板的核心 PromQL 表达式（整理自 dashboard JSON）。# 注释为 PromQL 合法注释，用于说明设计意图：

```
# 面板 1: KV event blocks by medium (id 29)
# max 与 min 双查询，按 namespace / event / medium 分组。
# 所有 replica 都接收完整事件流，因此 min 为 0 而 max 正常时，
# 说明某个 replica 丢失订阅、正在盲路由，单看 max 会掩盖该故障。
max by (namespace, event, medium) (rate(sgl_router_kv_event_blocks_total[$__rate_interval]))
min by (namespace, event, medium) (rate(sgl_router_kv_event_blocks_total[$__rate_interval]))

# 面板 2: KV tree blocks by tier (fleet) (id 30)
# 先按 worker 取 max 以跨副本去重，再按 tier 求和：
# 绝对计数跨 worker 求和是合法的，与比率不同。
sum by (tier) (max by (worker_url, dp_rank, tier) (sgl_router_kv_tree_blocks))

# 面板 3: Tree coverage of engine host tier (id 31)
# router 视图与引擎实际持有的比值，是区分 " 保留问题 " 与 " 可见性问题 " 的关键：
# 分子为 host 层树块数 × 块大小（按 namespace 分组，避免跨部署相除）；
```

```
# 分母为引擎 hicache host 层已用 token 数，仅取 tp_rank=0 防多 rank 重复计数。
(
  sum by (namespace) (
    max by (namespace, worker_url, dp_rank) (sgl_router_kv_tree_blocks{tier="host"})
  )
  * on(namespace) group_left() max by (namespace) (sgl_router_kv_block_size)
)
/ sum by (namespace) (sglang:hicache_host_used_tokens{tp_rank="0"})
```

## 评论区精华

唯一的 review 评论来自 sherlockwu，指出覆盖率面板分母的指标名与引擎实际暴露名不一致：

```
"Use the engine's actual metric name in the denominator. This query uses
sglang_hicache_host_used_tokens, but the engine emits
sglang:hicache_host_used_tokens (with a colon), as defined in
python/sglang/srt/observability/metrics_collector.py."
```

该问题在 PR body 中已有交代与应对：测试所用的 Prometheus scrape 管道会把指标名中的冒号重写为下划线，因此在测试环境中以 `sglang_hicache_host_used_tokens` 形式验证；面板本身保留引擎直接暴露的带冒号名称 `sglang:hicache_host_used_tokens`，修正后的表达式已重新 POST 并通过验证。sherlockwu 随后给出 APPROVED，评论无遗留未决问题。

- 覆盖率面板分母指标名不一致 (correctness): PR body 说明：测试环境所用 store 的 scrape 管道会把冒号重写为下划线，因此验证时以改写后名称测试；面板最终保留引擎原生名称 `sglang:hicache_host_used_tokens`，修正后的表达式已重新 POST 并解析通过。

## 风险与影响

- 风险：以下风险均针对该监控资产，不影响运行时逻辑：
- 指标名冒号兼容性：面板分母直接引用 `sglang:hicache_host_used_tokens`（带冒号，定义于 `python/sglang/srt/observability/metrics_collector.py`）。Prometheus 直接抓取引擎时保留冒号；但若用户监控链路经过指标名改写（如测试环境的 scrape pipeline 将 `:` 重写为 `_`），该面板分母将无数据。作者已在 PR body 中说明这一权衡并保留引擎原生名称。
- 标签假设：分母过滤 `tp_rank="0"`，依赖引擎指标携带 `tp_rank` 标签；对于不带该标签的旧版本引擎，表达式可能返回空或行为不确定。作者在 PR body 中明确承认 device-tier 覆盖需要 per-rank 标签的 `kv_evictable_tokens`，属于已知未覆盖范围。
- 依赖前置系列：本面板读取 #39109 暴露的 `sgl_router_kv_*` 系列，需按 stack 顺序合入；单独合入时所有面板显示 "No data"，可能被误判为监控故障。
- 无自动化回归：纯 JSON 配置无测试覆盖，panel id、gridPos 依赖人工检查；本次已验证 id 唯一，但后续合入其他 dashboard 修改时存在冲突可能。
- 影响：本 PR 只影响 Grafana 监控看板，对 sgl-router 运行时零影响。受益方是运维与性能工程师：新增的面板可将 "引擎侧真实淘汰 (retention problem)" 与 "router 丢标签 / 丢订阅 (visibility problem)" 两类问题直接区分开，quietest replica 曲线可暴露盲路由副本。对团队而言，这是 HiCache 分层缓存路由体系的最后一环，使其从 "有指标" 走向 "

可诊断"; 作者明确将 device-tier 覆盖面板留作后续增量, 需等待引擎侧 per-rank 分母就绪。

- 风险标记: 依赖前置系列指标, 纯 JSON 无自动化校验, 指标名冒号兼容性, 覆盖率对 tp\_rank 标签有假设

## 关联脉络

- PR #39109 [Router] Expose the KV storage-tier stream and tree occupancy on /metrics (2/4): 本 PR 所有面板读取的 sgl\_router\_kv\_event\_blocks\_total、sgl\_router\_kv\_tree\_blocks、sgl\_router\_kv\_block\_size 系列均由该 PR 在 /metrics 暴露, 是本 PR 的直接数据来源, stack 依赖关系。
- PR #39110 [Router] Real-GPU e2e coverage for storage-tier-aware cache routing (3/4): 本 PR 的 base branch 为 router-kv-tiers-3-e2e (即 #39110), e2e 测试保障了存储层感知路由在真实 GPU 上产生与面板配套的指标事件。
- PR #39280 [HiCache] Label radix-cache metrics per rank and split the "shrunk" prefetch reason: 该 PR 让 HiCache 指标按 rank 打标, 本 PR 覆盖率分母 sglang:hiocache\_host\_used\_tokens{tp\_rank="0"} 正依赖于 per-rank 标签的可用性, 属于同一条可观测性演进线。