

PR #39109 完整报告

sgl-project/sglang

[Router] Expose the KV storage-tier stream and tree occupancy on /metrics (2/4)

合并时间: 2026-09-16 04:30

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/39109>

执行摘要

本 PR 为 `experimental/sgl-router` 的 `/metrics` 端点新增一套 KV 分层存储 (HiCache tier) 可观测系列: 事件计数 (按 `event × medium`)、树占用 (按 `worker × tier`)、块大小换算系数、丢批计数、记账错误计数与树维护模式开关。纯增量改动, 不改变任何路由行为, 所有系列均拉取式渲染; 它与 #39108 (分层树)、#39110 (e2e 测试) 构成 4 连 PR 的第 2 环, 目标是把“router 丢弃 `medium` 标签”和“引擎从不发送标签”从同一种低命中率症状中分离出来。

功能与动机

PR body 直言: #39108 修复的 tier bug 存活数月而无人察觉, 因为此前“nothing plotted the router's view of the fleet's KV cache against what the engines actually held”。router 丢弃 `medium` 标签与引擎从不发送标签, 症状完全相同——低 `cache-hit` 率——且都无法与普通引擎侧驱逐区分。本 PR 提供区分两者的时间序列: `sgl_router_kv_tree_blocks * sgl_router_kv_block_size` 与引擎自身占用的比值即覆盖率, `~1` 表示树镜像引擎、`~0` 表示引擎持有路由看不见的 tier、`>1` 表示有批事件在传输中丢失 (ZMQ 高水位丢弃); `block_stored/unknown` 非零则是“引擎新增存储层、请升级 router”的信号。

实现拆解

1. 变更入口与装配: `main.rs` 调用 `kv_index.metrics_source()` 写入 `app_ctx.kv_metrics`。
`KvEventIndex::metrics_source()` 仅在 `maintain_tree == true` 时返回 `KvIndexMetrics` (`tree + tally` 两个只读句柄), `metadata-only` 模式 (外部 `Indexer` 为路由信号) 返回 `None`, 整块系列从 `scrape` 中省略——因为结构性全零会被 `HELP` 文本误读为“tier 流未到达”。
2. 新增 `tally.rs` 计数模块: `EventKind + EventTally`, `medium` 折叠为固定 6 值防伪造 `series`; `record()` 常见路径无锁无分配; 未知 `medium` `warn-once` 且有 16 个字符串上限; `slot()/label()` 用穷举 `match` 而非判别式转型, 新增枚举变体直接编译失败。
3. `tree.rs` 增量记账与加固: `TreeState` 新增 `occupancy: HashMap<KvWorkerId, TierCounts>` 与 `accounting_errors`, 在 `insert / remove / clear_worker / evict_lru` 四个突变点记账; `clear_worker` 通过 `account_remove_counts()` 一次整 `carrier` 释放, 避免失败路径在持有写锁时每 `node` 打一行 `error!`; `TreeState::insert` 两个不变量违反路径补记已写入的占用增量; 常量断言把 `Tiers::ALL` 钉死为 `Tiers::SLOTS` 的并集。
4. `pump` 接线与丢批检测: `pump_loop` 按 `kind × medium` 计数事件与 `block` 数; `seq` 缺口计为 `record_lost_batches()` 并打 `warn`; 乱序 `batch` 在计数前被过滤。

5. /metrics 渲染: render_kv_tiers() 输出 sgl_router_kv_block_size、sgl_router_kv_tree_blocks{worker_url,dp_rank,tier}、sgl_router_kv_events_total、sgl_router_kv_event_blocks_total、sgl_router_kv_event_batches_lost_total、sgl_router_kv_tree_accounting_errors_total{reason}; sgl_router_kv_tree_maintained 无条件输出。
6. 配套: monitoring/README.md 文档化每个系列与 coverage 三种读数; escape_label 提升为 pub(crate); kv_events/mod.rs 补齐导出。测试全部内嵌 #[cfg(test)], cargo test -p sgl-router 718 个用例通过 (566 lib + 152 integration), 含随机游走不变量测试、/metrics 双向门控测试、丢批与 unknown medium 折叠测试, cargo clippy --all-targets 干净。

experimental/sgl-router/src/policies/kv_events/tree.rs

树在 4 个突变点增量记账按 worker × tier 的占用, 并修复 insert 不变量路径漏记账; 新增 Tiers::SLOTS 及编译期断言, 是本次改动风险最集中的文件。

```
/// TierCounts 以 Tiers::SLOTS 的位置为索引: device / host / disk / external。
/// 一个 node 同时落在 device 与 host 时, 两个槽位都计数。
pub type TierCounts = [u64; TIER_SLOT_COUNT];
```

```
/// 把一个 node 的 tier 位累加进 counts。
fn tally_tiers(counts: &mut TierCounts, bits: Tiers) {
    for (slot, (tier, _)) in Tiers::SLOTS.iter().enumerate() {
        if bits.contains(*tier) {
            counts[slot] += 1;
        }
    }
}
```

```
/// 树内可变状态: 新增 occupancy 与 accounting_errors 两个账本。
/// occupancy 在四个突变点 (insert / remove / clear_worker / evict_lru)
/// 记账, 因此 /metrics 抓取只需一次读锁, 不需要整树遍历。
struct TreeState {
    nodes: HashMap<NodeId, Node>,
    by_hash: HashMap<i64, HashSet<NodeId>>,
    next_id: NodeId,
    occupancy: HashMap<KvWorkerId, TierCounts>,
    /// 记账自相矛盾的次数, 按原因分槽。release 构建下 debug_assert
    /// 被编译掉, 计数器是区分 "worker 无上报" 与 "记账漏记" 的唯一途径。
    accounting_errors: [u64; ACCOUNTING_REASONS.len()],
}
```

```
impl TreeState {
    /// 按整个 chain 记账: delta 含一条链上所有 node 的 tier 增量,
    /// 一次行查找完成, 而不是每 block 一次。
    fn account_add(&mut self, worker: &KvWorkerId, delta: TierCounts) {
        if delta.iter().all(|&c| c == 0) {
            return;
        }
    }
}
```

```

}
match self.occupancy.get_mut(worker) {
    Some(counts) => {
        for (acc, d) in counts.iter_mut().zip(delta) {
            *acc += d;
        }
    }
    None => {
        self.occupancy.insert(worker.clone(), delta);
    }
}
}
}

```

/// 释放整个 carrier 的记账：clear_worker 一次调用完成整棵树的释放。

/// 若按 node 逐行记账，失败路径会在持有树写锁时每 node 打一行

/// error! ——数十万行日志把每个路由决策都堵在后面。

```

fn account_remove_counts(&mut self, worker: &KvWorkerId, delta: TierCounts) {
    if delta.iter().all(|&c| c == 0) {
        return;
    }
    let Some(counts) = self.occupancy.get_mut(worker) else {
        // 释放一个没有占用行的 carrier：属于记账漏记的 bug。
        // 计数 + error! 而不是 panic，避免 pump 长驻任务整体宕掉。
        self.accounting_errors[REASON_MISSING_ROW] += 1;
        error!(
            worker = %worker.url,
            dp_rank = worker.dp_rank,
            "tree invariant violation: releasing tiers for a carrier with no occupancy row",
        );
        return;
    };
    let mut underflowed = false;
    for (acc, d) in counts.iter_mut().zip(delta) {
        underflowed |= *acc < d;
        // 饱和减：绝不让 gauge 回绕成负数；但若 underflow 导致
        // 全零，下面的 all-zero 判断会删掉一个仍持有 node 的 carrier
        // 的行——所以必须有 accounting_errors 兜底解释这一点。
        *acc = acc.saturating_sub(d);
    }
    if underflowed {
        self.accounting_errors[REASON_UNDERFLOW] += 1;
        error!(
            worker = %worker.url,
            dp_rank = worker.dp_rank,
            "tree invariant violation: occupancy underflow",
        );
    }
    if counts.iter().all(|&c| c == 0) {
        self.occupancy.remove(worker);
    }
}

```

```
}  
}  
}
```

评论区精华

sherlockwu 在 [monitoring/README.md](#) 第 57 行提出 P2 建议:

Use the engine's exported metric names in the guidance. For example, `SchedulerMetricsCollector` declares `sglang:kv_used_tokens`, `sglang:kv_evictable_tokens`, and `sglang:hicache_host_used_tokens` (with a colon), and the standard Prometheus exporter preserves that colon.

即 README 里用于覆盖率换算的 `sglang_kv_used_tokens` 等名称与引擎实际导出的带冒号名称不一致, 按文档配置面板会踩空。该建议未强制修改, PR 随后被 Approve; 运维落地面板时需按带冒号的真实导出名对齐。

风险与影响

- 热路径记账: TreeState 的 4 个突变点新增占用记账, 设计为按 chain 记账控制开销, 但仍属路由决策核心路径上的写操作, 后续备份密度上升需复查。
- 丢批推断假设: `sgl_router_kv_event_batches_lost_total` 依赖发布方 seq 稠密假设, 发布方序号语义变化会产生误报; 重建游标后合法重启的场景已通过 `last_applied` 比较处理。
- 覆盖率口径: $\text{kv_tree_blocks} \times \text{kv_block_size} \div \text{引擎占用}$ 需要运维自行保证两侧同一 deployment, 跨部署相除得到无意义比值; HELP 文本与 README 已强调但无法强制。
- 静默类故障: underflow 饱和后 carrier 行会被删除, 面板会读成“worker 无上报”, 只有 `accounting_errors_total` 能区分; 这是刻意的设计取舍, 但要求面板同时监控该计数器。
- 对用户与团队: 路由行为零变化, 影响集中在可观测性面; 这是 4 连 PR 栈中 Grafana 面板 (4/4) 与真实 GPU e2e (3/4) 的观测基础。`sgl_router_kv_tree_maintained` 等“HELP 文本即契约”的设计将约束后续新增 tier 时必须同步维护三张表。
- CI 状态: PR Test (Base) 通过; PR Test (Extra) 运行 #34999679570 失败, 材料未给出原因, 合入前建议确认与本次改动无关。

关联脉络

本 PR 是 [\[Router\] KV 分层存储路由](#) 4 连栈的第 2 环: #39108 先引入带 tier 语义的树 (tagged removal 只清自身 tier 的新语义正是 $\text{coverage} > 1$ 读数的来源), 本 PR 把该树的占用与事件流暴露到 [/metrics](#), #39110 (3/4) 用真实 GPU e2e 测试驱动这些系列, 4/4 (本 PR 不包含) 是配套 Grafana 面板。与仓库近期 HiCache 演进 (#39283 预取管线重构、#39480 存在缓存优化) 同属分层缓存方向: 前者保障引擎侧 tier 事件发布的正确性, 后者保障 router 侧对 tier 事件的可观测性, 两者共同收口“引擎侧故障被误读为路由器 bug”这一类长期难以定位的问题。