

# PR #39169 完整报告

sgl-project/sglang

[Router] Pin to the prefix owner when the whole fleet is queueing (--saturation-queue-floor)

合并时间: 2026-09-16 16:16

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/39169>

## [Router] Pin to the prefix owner when the whole fleet is queueing (--saturation-queue-floor)

### 执行摘要

本 PR 为 Router 的 `cache_aware` 路由新增 `--saturation-queue-floor N`: 当没有任何缓存候选同时通过队列闸门与硬准入、至少一个候选是被闸门拒绝的, 且可路由 prefill 舰队中没有任何 worker 的 fresh 队列读数低于该下限时, 取消这次收益为零的分流, 把请求钉在压力最小的前缀拥有者上。改动贯穿配置面 (`config/cli.rs`、`config/types.rs`)、选择层 (`policies/selection.rs`、`server/routes/chat.rs`)、核心判定 (`policies/admission.rs`、`policies/engine_load.rs`) 与观测 (`server/metrics.rs`) , 新增 `DecisionReason::SaturationPin` 与 `saturation_pin` 决策理由。默认不设置该参数时完全保持 #39168 的行为, 属于 opt-in 的策略增强。

### 功能与动机

PR body 把问题定义得很清楚: 在舰队饱和时继续分流, 会把一次流量突发放大成「持续的缓存命中率崩塌」。但严格条件「每个 worker 都超限」在大舰队上几乎不成立——少数 worker 短暂低于限额就让「免费 worker 永远存在」, 分流于是永远触发。floor 把这个条件变成可用形式: 饱和 = 没有任何可证明低于 floor 的目的地。作者进一步论证收益方向: 分流只有在存在可证明更闲的目的地时才有回报, 否则「等待在哪里都是等待」, 而等在前缀拥有者上可以从缓存预填充, 避免一次「驱逐其他前缀并制造下一轮 miss」的全量冷预填充。

设计上还有一条显式取舍: floor 不与 #39168 的舰队饱和第二级回退叠加, 而是在设置时替换它——两者都保持前缀, 但「选哪个 owner」的排序不同 (按 uncached work 重排 vs 仅按压力), 只能有一个做决定, 因此显式旋钮优先, gate 的 tier 覆盖 floor 未设置的情况。

### 实现拆解

1. 配置面与不变量 (`config/cli.rs`、`config/types.rs`、`policies/mod.rs`) : Cli 新增 `--saturation-queue-floor` (`Option<u64>`), `AffinityConfig` 新增同名字段、默认 `None`。校验加入三条不变量: 必须同时给出 `--worker-queue-limit` («没有分流可取消»)、`floor >= 1`、`floor <= limit` (floor 高于 limit 会在闸门仍能放行部分 worker 时宣称饱和); 该 flag 计入 `tuned_cache_candidates`, 从而继承 `--policy cache_aware` 约束, 且数值校验先于 policy 校验以保证报错指向正确原因。`CacheCandidateProposal` 增加 `saturation_queue_floor` 字段, `CacheAwarePolicy` 构造 proposal 时透传。

2. 请求侧传递 (`server/routes/chat.rs` → `policies/selection.rs`) : `chat.rs` 从 `ctx.config.model.affinity` 读出 `floor` 填入 `PrefillSelectionInputs`; `selection.rs` 新增同名字段 (`None` 即关闭 `pin`) , 并新增 `SATURATION_PIN_LOG_SAMPLE = 64` 采样计数器, 与既有的「闸门盲区告警」采用同一节流策略。
3. 核心判定 (`policies/admission.rs::resolve_cache_candidates`) : 函数签名新增 `fleet: &[Arc<Worker>]` (该模型可路由的 `prefill` 池) ; `queue_gate_rejected` 从计数器改为 `Vec<&CacheCandidate>` 以保留被拒 `owner` 供排序; `queue_gate_fell_back` 增加 `proposal.saturation_queue_floor.is_none()`; 在 `work_floor` 为 `None` 的早退分支中插入 `pin` —— 先 `any_fresh_queue_below` 判断舰队是否可证明饱和, 再对被拒 `owner` 单独构建 `FreshLoadLookup`、过滤硬准入、按 `compare_prefill_pressure` 取最小者。新增 `DecisionReason::SaturationPin`。
4. 饱和证据 (`policies/engine_load.rs::any_fresh_queue_below`) : 只统计有 `fresh` 采样且 `num_waiting_reqs < floor` 的 `worker`; 无采样不算空闲 (与 `queue_gate_admits` 的 `fail-open` 相反) , 并且只扫描调用方传入的可路由舰队 `URL` , 避免 `router` 全量负载表中的 `decode` 对端、其它模型 `worker` 或不可路由 `worker` 否决 `pin`。
5. 观测与标签 (`server/metrics.rs`、`policies/selection.rs`) : `prefill_policy_reason` 新增 `DecisionReason::SaturationPin` → `"saturation_pin"`, `cache candidate` 分支的 `record_policy_decision` 从固定字面量改为走该函数; `pin` 命中时记 `CacheAwareDecision::AllQueued` 并按 `1/64` 打 `info` 日志。`metrics.rs` 顶部文档重写 `all_queued` 语义: 两种触发条件、`floor` 低于 `limit` 会放宽标签、`owner` 容量耗尽时由 `off-owner` 回退记账。
6. 测试与配套: `admission.rs` 内新增 `saturation_proposal` 辅助与 9 个饱和 `pin` 单测; `config/cli.rs` 新增 `saturation_queue_floor_requires_the_queue_gate_and_stays_below_it`。PR body 报告 `cargo test 799` 通过、`cargo fmt --check` 与 `pre-commit` 干净; 没有新增 `proxy/e2e` 层次的集成测试文件, 也没有部署或 `schema` 变更。

## experimental/sgl-router/src/policies/admission.rs

核心实现与全部 `pin` 单测所在: `resolve_cache_candidates` 新增 `fleet` 参数、把被拒候选改为保留列表、加入饱和 `pin` 分支并新增 `DecisionReason::SaturationPin`。

`resolve_cache_candidates` 中新增的饱和 `pin` 分支 (相对完整实现单元) :

```
// 饱和钉选 (SaturationPin) : 没有候选能通过队列闸门与硬准入, 且至少一个候选
// 被闸门拒绝过。此时分流只有在存在「可证明更闲」的目的地时才有收益。
let pinned = if queue_gate_rejected.is_empty() {
    None
} else {
    proposal.saturation_queue_floor.and_then(|floor| {
        // 舰队范围扫描: 只要有任一可路由 worker 的 fresh 队列读数严格低于
        // floor, 就说明存在一个分流真能少排队的落点, pin 让位。
        if snapshot
            .any_fresh_queue_below(fleet.iter().map(|worker| worker.url.as_str()), floor)
        {
            return None;
        }
    })
}
```

```

// 独立的 FreshLoadLookup, 而不是共享的 `loads`: `loads` 只覆盖闸门
// 「通过」的候选集合, 而 pin 触发时这个集合恰好是空的。空 lookup 会
// 报告无引擎覆盖, 使所有压力比较回退到 router 本地负载、并列为零,
// 最终按 worker id 决出 —— 一个按字母序 pin 的实现比不 pin 更糟。
let pin_loads = FreshLoadLookup::new(
    Some(snapshot),
    queue_gate_rejected.iter().map(|candidate| &candidate.worker),
);
queue_gate_rejected
    .iter()
    .copied()
    // 跳过同时通不过硬准入的 owner: 容量耗尽的 worker 接不下请求。
    .filter(|candidate| {
        is_cache_candidate_admitted(candidate, request_input_tokens, &pin_loads)
    })
    // 仅按压力排序, 不按匹配前缀深度重排; 同压时按 worker id 稳定排序。
    .min_by(|left, right| {
        pin_loads
            .compare_prefill_pressure(&left.worker, &right.worker)
            .then_with(|| left.worker.id.0.cmp(&right.worker.id.0))
    })
});
// 有 pin 则返回带 SaturationPin 理由的决策, 否则返回 None 交给上层回退路径。
return CacheCandidateResolution {
    decision: pinned.map(|pinned| FinalDecision {
        selected: Arc::clone(&pinned.worker),
        primary: Arc::clone(&pinned.worker),
        backup: None,
        reason: DecisionReason::SaturationPin,
        candidate_range_id: pinned.candidate_range_id.clone(),
        load_snapshot_version: snapshot.version,
    }),
    prefill_pressure_source: loads.prefill_pressure_source(),
    admission_evaluated_candidates,
    admission_rejected_candidates,
    queue_gate_rejected_candidates,
    queue_gate_best_rejected_blocks,
    queue_gate_fell_back,
    fleet_all_queued,
    pressure_guard_compared_pairs: 0,
    pressure_guard_overrides: 0,
};

```

## experimental/sgl-router/src/policies/engine\_load.rs

新增舰队级饱和判定原语 `any_fresh_queue_below`, 其 fail-closed 极性是 pin 安全性的关键, 也是与闸门 fail-open 组合语义的落点。

饱和证据的判定原语, 以及与闸门 fail-open 的对照:

```

impl EngineLoadSnapshot {
    /// 舰队里是否存在「可证明更闲」的目的地：某个 worker 的 fresh 队列读数
    /// 严格小于 `floor`。没有 fresh 采样的 worker 不计入 —— pin 问的是
    /// 「是否存在可证明更好的目的地」，未知队列不是证明。
    ///
    /// 这与队列闸门 `queue_gate_admits` 的 fail-open 极性相反；两者组合的
    /// 效果是：信号缺失时闸门保持亲和、pin 也把请求留给前缀拥有者。
    ///
    /// 调用方传入可路由舰队而非让本函数扫描全表：router 全量负载表还包含
    /// decode 对端、其它模型的 worker 以及已不可路由的 worker，PD 部署下
    /// 一个空闲的 decode 对端就足以否决每一次 pin。
    pub fn any_fresh_queue_below<'u>(
        &self,
        fleet_urls: impl IntoIterator<Item = &'u str>,
        floor: u64,
    ) -> bool {
        fleet_urls.into_iter().any(|url| {
            self.workers
                .get(url)
                .is_some_and(|load| load.num_waiting_reqs < floor)
        })
    }
}

/// 闸门判定的对照实现：`limit` 未设置或该 worker 没有 fresh 采样时一律放行。
/// `is_none_or` 就是 fail-open —— 未知队列不阻止一次缓存亲和选择。
pub(crate) fn queue_gate_admits(
    snapshot: &EngineLoadSnapshot,
    worker: &Worker,
    limit: Option<u64>,
) -> bool {
    let Some(limit) = limit else {
        return true;
    };
    snapshot
        .fresh_load_for_url(&worker.url)
        .is_none_or(|load| load.num_waiting_reqs < limit)
}

```

## experimental/sgl-router/src/config/cli.rs

新增 CLI 旋钮并定义三条不变量（依赖 `gate`、`floor`  $\geq 1$ 、`floor`  $\leq$  `limit`），配套 `saturation_queue_floor_requires_the_queue_gate_and_stays_below_it` 测试。

CLI 校验块：数值检查先于 policy 检查，避免误报原因：

```

// 数值校验先于 policy 校验：在任何 policy 下都错的取值应该先报取值本身。
if self.worker_queue_limit == Some(0) {
    return Err(anyhow!("--worker-queue-limit must be at least 1"));
}

```

```
if let Some(floor) = self.saturation_queue_floor {
    // floor 修改的是闸门的分流行为；没有闸门就没有分流可取消，
    // 这个旋钮会变成死配置。
    let Some(limit) = self.worker_queue_limit else {
        return Err( anyhow!(
            "--saturation-queue-floor requires --worker-queue-limit (there is no \
            diversion to cancel without it)"
        ));
    };
    if floor == 0 {
        return Err( anyhow!("--saturation-queue-floor must be at least 1"));
    }
    // floor <= limit 让饱和标签保持可读：floor 高于 limit 会在闸门仍能
    // 放行部分 worker 时宣称舰队饱和。
    if floor > limit {
        return Err( anyhow!(
            "--saturation-queue-floor ({floor}) must be at most --worker-queue-limit ({limit})"
        ));
    }
}
```

评论区精华

ShangmingCai (APPROVED) : Logic is correct as we discussed and verified before, looks good.

唯一的 review 是批准，说明该逻辑在合入前已经过线下讨论与验证，PR 内没有出现逐行争议或返工。随后 ShangmingCai 评论 [/tag-and-rerun-ci](#) 触发重跑（body 的 CI 状态显示 Latest PR Test (Extra) 曾失败、Latest PR Test (Base) 通过、AMD ROCm 无对应 run）。

作者本人在 body 中标记的待定设计问题同样值得记录：

分流只有在存在可证明更闲的目的地时才有回报；pin 的排序是 pressure-only，不按匹配前缀深度重排，所以「更浅但更闲」的 owner 可能胜出 —— 替代方案是 locality-first 的 [compare\\_cache\\_candidates](#)。

两条标签 caveat：floor 低于 limit 会放宽 [all\\_queued](#)；当所有被拒 owner 也容量耗尽时 pin 不产出决策，标签由 off-owner 回退绘制。

风险与影响

| 风险点        | 说明                                                                                                                       | 落点                                                              |
|------------|--------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------|
| 既有行为被开关式替换 | 启用 floor 后 <a href="#">queue_gate_fell_back</a> 恒为 false，#39168 的舰队饱和 tier 不再生效；floor 远低于 limit 时 pin 更频繁地保持亲和，TTFT 可能上升 | <a href="#">policies/admission.rs</a>                           |
| 配置耦合       | floor 只能与 gate 同开且 $\leq$ limit， <a href="#">--dp-size</a> 部署需同步放大两个值；CLI 只做静态校验，无运行时合理性告警                               | <a href="#">config/cli.rs</a> 、 <a href="#">config/types.rs</a> |

| 风险点          | 说明                                                                                                  | 落点                                                                  |
|--------------|-----------------------------------------------------------------------------------------------------|---------------------------------------------------------------------|
| 采样缺失<br>误判饱和 | <code>any_fresh_queue_below</code> 按 fail-closed 处理未知队列，部分插桩或采样中断的舰队会被整体读成饱和                        | <code>policies/engine_load.rs</code>                                |
| 指标契约<br>变化   | <code>all_queued</code> 不再等价于「每个 worker $\geq$ limit」，依赖该 label 的命中率看板 / 告警需同步改口径，且无渲染断言测试          | <code>server/metrics.rs</code> 、 <code>policies/selection.rs</code> |
| 测试层次<br>偏单元  | 只有 <code>policies/admission.rs</code> 与 <code>config/cli.rs</code> 的内联单测，缺少 proxy/e2e 验证真实路由路径与指标输出 | 测试配套                                                                |

影响面方面：只有显式配置 `--saturation-queue-floor` 的 `cache_aware` 部署行为改变，未配置的舰队零影响；开启后饱和期请求更倾向留在前缀拥有者，减少冷预填充引发的 KV 驱逐连锁，代价是排队时间被显式化而非被分流掩盖。性能方面 `pin` 只在候选集清空时触发，额外成本为一次  $O(\text{fleet})$  扫描加一个 `FreshLoadLookup` 与 `min_by`，不在热路径。

### 关联脉络

本 PR 是 Router 饱和策略栈的第三层：#39322 (selection seam, 已合入) → #39168 (队列闸门 `--worker-queue-limit`, 已合入) → 本 PR (饱和下限) → #39170 (候选采样 `--min-load-choices`, 兄弟 PR)。它与 #39170 都曾以 #39168 为基底，在 #39168 合入后各自 rebase 到 main。从演进脉络看，这条线的主题是「用显式、可观测的旋钮取代隐式的兜底 tier」：闸门先解决「忙 worker 不该凭缓存亲和夺冠」，本 PR 再解决「全队排队时分流本身就是亏本买卖」，并把决策理由与 `all_queued` / `saturation_pin` 指标分离，使饱和态在命中率看板上不可被误读为健康。后续若要继续演进，tiebreak 的压力优先 vs 局部性优先、以及 `floor` 与 `dp_size` 的缩放关系是两个明显的承接点。