PR #35248 完整报告

sgl-project/sglang

[Metrics] Discount queued prefill load by recent cache hits when waiting-queue matching is off

合并时间：2026-08-19 02:27

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/35248

执行摘要

- 一句话：等待队列 prefill 负载按近期缓存命中率折减，修复 autoscaling 高估
- 推荐动作：值得精读。该 PR 展示了「精确信号缺失时用近期统计量做有界估计」的工程思路：既保留精确路径（匹配开启、chunked 请求），又为退化场景（LPM 超限、cache-agnostic）提供平滑近似；同时刻意维持导出指标语义不变，降低观测兼容性冲击。适合作为调度器指标设计的参考案例，也建议阅读 test_load_inquirer.py 中对 LPM 128 队列边界和命中率估计数值的断言方式。

功能与动机

PR body 明确指出：get_num_waiting_uncached_tokens 在 waiting-queue prefix matching 不可用（cache-agnostic policy without fast prefix matching, or LPM past its 128-request queue limit）时按完整 seqlen 计数，导致 On cache-friendly multi-turn workloads this overstates queued prefill demand by an order of magnitude, inflating autoscaling signals derived from the load snapshot exactly when the queue is deep. 即队列越深、缓存命中率越高，负载信号失真越严重，直接影响基于负载快照的自动扩缩容决策。

实现拆解

实现分 5 步展开，核心是引入「窗口化缓存命中率」作为等待队列负载的估计依据：

1. 新增环境变量（python/sglang/srt/environ.py）：在调度器 token 预算区新增 SGLANG_CACHE_HIT_RATE_WINDOW_SECONDS = EnvFloat(15.0)，用于控制命中率窗口长度，默认 15 秒。
2. 窗口化命中率统计（python/sglang/srt/managers/scheduler_components/metrics_reporter.py）：新增 _CacheHitRateWindow 类，用 deque 保存 (now, hit_tokens, total_tokens) 样本，add() 时追加样本并弹出超出窗口的过期样本，返回窗口内 token 加权命中率。SchedulerMetricsReporter.__post_init__ 初始化 cache_hit_rate_window 和 recent_cache_hit_rate = 0.0；report_prefill_stats 在计算 effective hit/input tokens 后调用 cache_hit_rate_window.add(...) 更新窗口值。注意导出的 stats.cache_hit_rate 仍是 per-report 瞬时值，窗口值只用于负载估计。
3. 等待队列匹配谓词（python/sglang/srt/managers/schedule_policy.py）：新增 waiting_queue_prefix_matched(waiting_queue)，调用 _determine_active_policy(waiting_queue) 获取实际生效 policy（LPM 超过 128 请求时降级为 FCFS），返回「

policy 是否为 CacheAwarePolicy，或 tree_cache 支持 fast prefix matching」。

4. 负载估计逻辑改造（python/sglang/srt/managers/scheduler_components/load_inquirer.py）：SchedulerLoadInquirer 新增 waiting_queue_prefix_matched 和 get_recent_cache_hit_rate 两个 Callable 字段；get_num_waiting_uncached_tokens 改为：匹配开启时使用精确的 seqlen - num_matched_prefix_tokens，匹配关闭时使用 int(seqlen * (1 - recent_cache_hit_rate)) 估计，活动 chunked 请求始终走精确路径。
5. 依赖注入（python/sglang/srt/managers/scheduler.py）：init_load_inquirer 中注入两个 lambda，分别连接 self.policy.waiting_queue_prefix_matched(self.waiting_queue) 与 self.metrics_reporter.recent_cache_hit_rate。

测试配套：新增 test/registered/unit/managers/test_load_inquirer.py，覆盖 cache-agnostic policy 对 fast matching 的依赖、LPM 128/129 队列上限边界、估计值与精确值的计算；test/registered/unit/observability/test_forward_pass_metrics.py 增加 _CacheHitRateWindow 的 15 秒窗口过期边界用例。

关键文件：

- python/sglang/srt/managers/scheduler_components/metrics_reporter.py（模块 指标上报；类别 source；类型 core-logic；符号 _CacheHitRateWindow, init, add）：新增 _CacheHitRateWindow 窗口化命中率统计，并在 report_prefill_stats 中更新 recent_cache_hit_rate，是本次估计逻辑的数据源。
- python/sglang/srt/managers/scheduler_components/load_inquirer.py（模块 负载查询；类别 source；类型 core-logic；符号 get_num_waiting_uncached_tokens）：get_num_waiting_uncached_tokens 是核心行为变更点：匹配关闭时改用 seqlen * (1 - recent_cache_hit_rate) 估计，并新增两个依赖注入 Callable。
- python/sglang/srt/managers/schedule_policy.py（模块 调度策略；类别 source；类型 core-logic；符号 waiting_queue_prefix_matched）：新增 waiting_queue_prefix_matched 谓词，统一表达「等待队列请求是否携带精确 num_matched_prefix_tokens」，并正确处理 LPM 128 队列上限的降级场景。
- python/sglang/srt/managers/scheduler.py（模块 调度器；类别 source；类型 core-logic）：init_load_inquirer 注入 waiting_queue_prefix_matched 与 get_recent_cache_hit_rate 两个 lambda，打通 schedule_policy 与 metrics_reporter 到负载查询器的数据通路。
- python/sglang/srt/environ.py（模块 环境配置；类别 source；类型 configuration）：新增 SGLANG_CACHE_HIT_RATE_WINDOW_SECONDS 环境变量，控制窗口化命中率的统计时长，默认 15 秒。
- test/registered/unit/managers/test_load_inquirer.py（模块 负载查询；类别 test；类型 test-coverage；符号 TestSchedulePolicyWaitingQueueMatching, make_policy, test_cache_agnostic_policy_requires_fast_matching, test_lpm_queue_limit_can_disable_matching）：新增测试覆盖谓词边界（cache-agnostic 需要 fast matching、LPM 128/129 队列上限）以及估计 / 精确两种 waiting token 记账路径。
- test/registered/unit/observability/test_forward_pass_metrics.py（模块 前向指标；类别 test；类型 test-coverage；符号 test_cache_hit_rate_window_keeps_last_15s_of_tokens）：补充 _CacheHitRateWindow 的 15 秒窗口过期边界测试，验证窗口滑动后命中率正确

更新。

关键符号：_CacheHitRateWindow.add, SchedulerLoadInquirer.get_num_waiting_uncached_tokens, SchedulePolicy.waiting_queue_prefix_matched

关键源码片段

python/sglang/srt/managers/scheduler_components/metrics_reporter.py

新增 _CacheHitRateWindow 窗口化命中率统计，并在 report_prefill_stats 中更新 recent_cache_hit_rate，是本次估计逻辑的数据源。

```
# 模块顶部读取配置：窗口秒数（默认 15 秒）
CACHE_HIT_RATE_WINDOW_SECONDS = envs.SGLANG_CACHE_HIT_RATE_WINDOW_SECONDS.get()


class _CacheHitRateWindow:
    """维护 token 加权的近端缓存命中率。

    只统计最近 CACHE_HIT_RATE_WINDOW_SECONDS 秒内的 prefill 样本，
    避免把很久以前的命中率混入当前等待队列的负载估计。
    """

    def __init__(self) -> None:
        self.samples = deque() # 元素为 (now, hit_tokens, total_tokens)
        self.hit_tokens = 0
        self.total_tokens = 0

    def add(self, hit_tokens: int, total_tokens: int, now: float) -> float:
        # 只有真正有 prefill token 的 report 才入窗，避免稀释窗口
        if total_tokens > 0:
            self.samples.append((now, hit_tokens, total_tokens))
            self.hit_tokens += hit_tokens
            self.total_tokens += total_tokens

        # 弹出过期样本，保证返回的是最近窗口内的加权命中率
        cutoff = now - CACHE_HIT_RATE_WINDOW_SECONDS
        while self.samples and self.samples[0][0] <= cutoff:
            _, expired_hit_tokens, expired_total_tokens = self.samples.popleft()
            self.hit_tokens -= expired_hit_tokens
            self.total_tokens -= expired_total_tokens

        return self.hit_tokens / self.total_tokens if self.total_tokens > 0 else 0.0
```

python/sglang/srt/managers/scheduler_components/load_inquirer.py

get_num_waiting_uncached_tokens 是核心行为变更点：匹配关闭时改用 seqlen * (1 - recent_cache_hit_rate) 估计，并新增两个依赖注入 Callable。

```
def get_num_waiting_uncached_tokens(self) -> int:
```

```
    """Estimate input tokens waiting for prefill compute."""
    if self.disaggregation_mode == DisaggregationMode.DECODE:
        return 0

    # 等待队列是否做过精确前缀匹配：匹配开启时用精确值，否则用估计值
    waiting_queue_prefix_matched = self.waiting_queue_prefix_matched()
    cache_miss_rate = 1.0 - self.get_recent_cache_hit_rate()

    num_tokens = 0
    for req in self.get_waiting_queue():
        if waiting_queue_prefix_matched:
            num_tokens += max(0, req.seqlen - req.num_matched_prefix_tokens)
        else:
            # 没有匹配信息时，按近端缓存命中率估计未命中部分
            num_tokens += int(req.seqlen * cache_miss_rate)

    # 正在分块执行的请求总是有精确的 prefix_indices，直接精确计算
    cr = self.get_chunked_req()
    if cr is not None:
        num_tokens += max(0, cr.seqlen - len(cr.prefix_indices))
    return num_tokens
```

python/sglang/srt/managers/schedule_policy.py

新增 waiting_queue_prefix_matched 谓词，统一表达「等待队列请求是否携带精确 num_matched_prefix_tokens」，并正确处理 LPM 128 队列上限的降级场景。

```
def waiting_queue_prefix_matched(self, waiting_queue: List[Req]) -> bool:
    # 先判定当前实际启用的 policy（LPM 超过 128 个请求时降级为 FCFS）
    policy = self._determine_active_policy(waiting_queue)
    # CacheAwarePolicy 一定做等待队列前缀匹配；
    # 对 CacheAgnosticPolicy，只有 tree_cache 支持 fast match 时才有匹配信息
    return (
        isinstance(policy, CacheAwarePolicy)
        or self.tree_cache.supports_fast_match_prefix()
    )
```

评论区精华

该 PR 无实质 review 讨论，review_comments 为 0，仅有一条作者触发的 CI 重跑命令「/tag-and-rerun-ci」。决策主要由 PR body 自述驱动：明确区分「导出的 cache_hit_rate 保持 per-report 语义」和「窗口化命中率只用于等待队列负载估计」，避免破坏既有监控语义。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. 估计偏差风险：等待队列的未匹配请求使用全局 recent_cache_hit_rate 近似，而该窗口值来自最近处理完成的 prefill batch。若队列中的请求与最近批次请求分布差异大（例

如新会话突发涌入），命中率估计会低估或高估实际 uncached tokens，进而影响 prefill 排队时间和 autoscaling 反应。

2. int() 截断风险：load_inquirer.py 中 int(req.seqlen * cache_miss_rate) 向下取整，短请求（如 seqlen 个位数）在命中率高时可能被截成 0，完全忽略其 prefill 负载。
3. 指标语义变化：get_num_waiting_uncached_tokens 影响 LoadSnapshot 与 /v1/loads，依赖该数值的 DP 负载均衡和 autoscaler 会观察到数值跳变，需要监控对比。
4. 新配置项默认值：SGLANG_CACHE_HIT_RATE_WINDOW_SECONDS 默认 15 秒，在突发流量下窗口可能过短导致命中率波动大；该配置为 EnvFloat，非法值会在启动期暴露。 - 影响：影响范围为调度器负载上报链路：SchedulerLoadInquirer、SchedulerMetricsReporter、SchedulePolicy 三者联动，波及 DP 负载均衡快照、/v1/loads 接口以及依赖负载信号的 autoscaling 组件。对用户而言，缓存友好的多轮对话、长前缀共享场景下排队负载不再被量级性高估，扩缩容行为更贴近真实 prefill 需求；对团队而言，load 指标语义从「精确计数」变为「可估计」，需要关注新增 recent_cache_hit_rate 与 waiting_queue_prefix_matched 两个新的依赖注入点，后续调试负载异常时需同时检查这两个信号。 - 风险标记：估计逻辑替代精确计数，autoscaling 信号语义变化，全局命中率近似等待队列，int() 截断可能低估短请求，新增配置项默认 15 秒

关联脉络

- PR #35191 [Scheduler] Cap prefill-delayer queue target by admission capacity: 同为调度器负载与排队信号相关的改动，涉及 prefill 队列目标与 admission capacity，与本次负载估计调整共同影响 prefill 排队行为和 autoscaling 信号解读。
- PR #35049 [PD] Deferred decode-side KV release for aborts mid-transfer: 同为调度与负载相关修复，涉及 disaggregation 下队列与 KV 释放逻辑，虽不在同文件，但同属负载与调度正确性演进脉络。