

PR #39280 完整报告

sgl-project/sglang

[HiCache] Label radix-cache metrics per rank and split the "shrunk" prefetch reason

合并时间: 2026-09-16 08:54

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/39280>

执行摘要

- 一句话: HiCache 指标按 rank 打标并拆分 shrunk 归因, 修复多 rank 求和失真
- 推荐动作: 值得精读。如果团队将 HiCache 作为生产级分层缓存使用, 本 PR 的指标语义约定 (per-shard vs logically replicated、attempt 级诊断 vs 最终 miss 计数) 是解读后续 HiCache 面板和告警的前提。特别值得学习的设计决策: 一是 rank 标签与 storage collector 键约定对齐, 避免 L2/L3 两层指标无法关联; 二是在重构导致某些 reason 失去代码路径后, 保留 label 预置以维持 dashboard 稳定、同时明确文档口径; 三是把 "窗口字节 delta / duration sum" 定义为活跃带宽, 区分纯 rate 含空闲时间的口径。

功能与动机

PR body 明确指出两个可观测性缺陷: 其一, "Every scheduler rank runs its own cache over its own KV shard, so the multiprocess registry summed the ranks and the series reported TP x the logical token count", 且 storage (L3) collector 已带 tp_rank / pp_rank / dp_rank 标签, 两层指标无法对齐; 其二, "storage_prefetch_unfulfilled_tokens_total{reason="shrunk"} covered four different terminal outcomes, which made it impossible to tell an anchor loss apart from a splice-time trim". 因此目标是让 L2 序列带上与 L3 一致的 rank 键, 并把模糊的 shrunk 桶拆成可区分的诊断原因。

实现拆解

1. 新增 rank 标签派生函数: 在 python/sglang/srt/observability/metrics_collector.py 中新增 radix_cache_metric_labels()。DP attention 开启时取 parallel.attn_tp_rank / parallel.attn_dp_rank, 否则取 parallel.tp_rank 且 dp_rank=0, 并带上 pp_rank, 键名与 storage collector 的 DP-aware 约定完全一致, 这样同一 rank 的 L2 与 L3 序列可直接对齐合并。
2. 接入缓存初始化: python/sglang/srt/mem_cache/base_prefix_cache.py 的 BasePrefixCache.init_metrics_collector() 改为调用 radix_cache_metric_labels(self.__class__.__name__, get_parallel(), is_dp_attention_enabled()) 构造 labels, 其中 is_dp_attention_enabled() 延迟导入以避免循环依赖, extra_metric_labels 仍可叠加。
3. 拆分 shrunk reason: python/sglang/srt/mem_cache/unified_radix_cache.py 的 _handle_storage_prefetch_anchor_loss() 将 reason="shrunk" 改为 reason="anchor_lost" (设备 anchor 在 IO 提交前消失, buffer 模式);

python/sglang/srt/managers/schedule_policy.py 的

_account_prefill_cache_admission() 将非 device_capacity 的准入缺口 reason 从shrunk 改为 cache_admission_shortfall (L3 加载到 L2 的 span 在 cache-mode 准入时已不可复用)。metrics_collector.py 同步扩展 reason 预置集合, 并保留aux_window_trim / device_overlap 两个无代码路径的 label 预置以维持 dashboard 稳定, 文档改写为 attempt 级诊断口径, 明确它不是最终 prefill cache-miss 计数。

4. 重写 L2 字节计数器文档: load_back_bytes_total 与 hicache_backup_bytes_total 明确为 per-shard (sum ranks 得物理总量), token 计数器 (load_back_tokens_total / hicache_backup_tokens_total) 为 logically replicated (read one rank), 并给出 "窗口字节 delta 除以同窗口 duration sum 为字节加权活跃带宽, plain rate 含空闲时间" 的计算口径。
5. 测试配套: test/registered/unit/observability/test_stat_loggers_di.py 新增 TestRadixCacheMetricLabels 覆盖 DP attention 开 / 关两种标注派生;
test/registered/unit/mem_cache/test_unified_radix_cache_unittest.py 将 anchor-loss 用例改名为 test_positive_hit_with_lost_anchor_reports_anchor_lost 并断言新 reason;
test/registered/unit/managers/test_prefill_adder.py 断言 cache_admission_shortfall。
作者本地跑通 65 passed / 1 skipped。

关键文件:

- python/sglang/srt/observability/metrics_collector.py (模块 指标采集; 类别 source; 类型 core-logic; 符号 radix_cache_metric_labels): 核心改动文件: 新增 radix_cache_metric_labels() 派生 rank 标签, 扩展 storage_prefetch_unfulfilled_tokens_total 的 reason 预置集合 (shrunk 拆分为 anchor_lost / aux_window_trim / device_overlap / cache_admission_shortfall), 并重写 L2 字节计数器的 per-shard 与带宽文档口径。
- python/sglang/srt/mem_cache/base_prefix_cache.py (模块 缓存基类; 类别 source; 类型 dependency-wiring; 符号 init_metrics_collector): radix-cache collector 的初始化入口: init_metrics_collector() 从固定 cache_type 标签改为调用 radix_cache_metric_labels() 注入 rank 标签, 并延迟导入 is_dp_attention_enabled() 避免循环依赖。
- python/sglang/srt/managers/schedule_policy.py (模块 调度策略; 类别 source; 类型 core-logic; 符号 _account_prefill_cache_admission): cache-mode 准入侧 reason 拆分: _account_prefill_cache_admission() 将非 device_capacity 的准入缺口从 shrunk 改为 cache_admission_shortfall, 与 anchor_lost 形成两个可区分的 attempt 级诊断。
- python/sglang/srt/mem_cache/unified_radix_cache.py (模块 统一缓存; 类别 source; 类型 core-logic; 符号 _handle_storage_prefetch_anchor_loss): buffer 模式 anchor 丢失路径的 reason 改名: _handle_storage_prefetch_anchor_loss() 从 shrunk 改为 anchor_lost, 是 reason 拆分在统一缓存侧的对应改动。
- test/registered/unit/observability/test_stat_loggers_di.py (模块 指标测试; 类别 test; 类型 test-coverage; 符号 TestRadixCacheMetricLabels, test_labels_follow_the_storage_collector_rank_keys): 新增 TestRadixCacheMetricLabels: 用 SimpleNamespace 模拟 parallel 状态, 覆盖 DP

attention 开启 / 关闭两种 rank 标签派生，验证与 storage collector 键约定对齐。

- test/registered/unit/mem_cache/test_unified_radix_cache_unittest.py (模块 缓存单元测试; 类别 test; 类型 test-coverage; 符号 test_positive_hit_with_lost_anchor_reports_anchor_lost) : anchor-loss 用例从 is_reported_as_shrunk 改名为 reports_anchor_lost, 断言 _finish_storage_prefetch 收到 reason="anchor_lost", 验证统一缓存侧的 reason 拆分。
- test/registered/unit/managers/test_prefill_adder.py (模块 准入测试; 类别 test; 类型 test-coverage; 符号 test_storage_prefetch_fulfillment_resolves_at_admission) : PrefillAdder 准入结算用例把期望 reason 从 shrunk 改为 cache_admission_shortfall, 覆盖调度策略侧的 reason 拆分。

关键符号: radix_cache_metric_labels, BasePrefixCache.init_metrics_collector, PrefillAdder._account_prefill_cache_admission, UnifiedRadixCache._handle_storage_prefetch_anchor_loss, UnifiedRadixCache.finish_storage_prefetch_admission

关键源码片段

python/sglang/srt/observability/metrics_collector.py

核心改动文件: 新增 radix_cache_metric_labels() 派生 rank 标签, 扩展 storage_prefetch_unfulfilled_tokens_total 的 reason 预置集合 (shrunk 拆分为 anchor_lost / aux_window_trim / device_overlap / cache_admission_shortfall), 并重写 L2 字节计数器的 per-shard 与带宽文档口径。

metrics_collector.py —— rank 标签派生与 reason 预置

```
def radix_cache_metric_labels(
    cache_type: str, parallel: Any, dp_attention_enabled: bool
) -> Dict[str, Any]:
    # 每个 scheduler rank 都维护自己的 cache 分片; 若不携带 rank 标签,
    # multiprocessing registry 会把所有 rank 的 series 求和, 上报值变成
    # TP x 逻辑 token 数, 且与带 rank 标签的 L3 storage 序列无法对齐。
    # 这里复用 storage collector 的 rank 键约定, 保证同一 rank 的
    # L2 与 L3 序列能直接合并计算。
    if dp_attention_enabled:
        # DP attention 下 KV 按注意力域切分, rank 取注意力域内编号。
        tp_rank, dp_rank = parallel.attn_tp_rank, parallel.attn_dp_rank
    else:
        # 非 DP attention: 直接用 TP rank, dp_rank 恒为 0。
        tp_rank, dp_rank = parallel.tp_rank, 0
    return {
        "cache_type": cache_type,
        "tp_rank": tp_rank,
        "pp_rank": parallel.pp_rank,
        "dp_rank": dp_rank,
    }
```

RadixCacheMetricsCollector.__init__ 内的 reason 预置:

预置全部 reason 标签, 保证各路终态在首次上报前就有稳定 series。

```

# anchor_lost / cache_admission_shortfall 是本 PR 从 shrunk 拆分出的
# attempt 级诊断：前者表示设备侧 anchor 在 IO 提交前被驱逐，后者表示
# L3 加载到 L2 的 span 在 cache-mode 准入时已不可复用。
# aux_window_trim / device_overlap 是历史语义，当前重构后已无代码路径，
# 保留预置是为了让既有 dashboard 的 label 集合保持稳定。
for reason in (
    "below_threshold",
    "host_capacity",
    "device_capacity",
    "device_covered",
    "storage_transfer",
    "anchor_lost",
    "aux_window_trim",
    "device_overlap",
    "cache_admission_shortfall",
    "dropped",
):
    self.storage_prefetch_unfulfilled_tokens_total.labels(
        **self.labels, reason=reason
    )

```

python/sglang/srt/mem_cache/base_prefix_cache.py

radix-cache collector 的初始化入口：init_metrics_collector() 从固定 cache_type 标签改为调用 radix_cache_metric_labels() 注入 rank 标签，并延迟导入 is_dp_attention_enabled() 避免循环依赖。

base_prefix_cache.py —— 所有 radix 缓存（含 UnifiedRadixCache）共用
的指标初始化入口，本 PR 在此注入 rank 标签。

```

def init_metrics_collector(self):
    # 延迟导入避免模块级循环依赖；is_dp_attention_enabled 决定
    # 取 attn_tp_rank/attn_dp_rank 还是普通 tp_rank + dp_rank=0。
    from sglang.srt.layers.dp_attention import is_dp_attention_enabled

    labels = radix_cache_metric_labels(
        self.__class__.__name__, get_parallel(), is_dp_attention_enabled()
    )
    # extra_metric_labels 仍可叠加，兼容既有自定义标签能力。
    if get_observability().extra_metric_labels:
        labels.update(get_observability().extra_metric_labels)
    radix_cache_cls = resolve_collector_class(
        STAT_LOGGER_ROLE_RADIX_CACHE,
        RadixCacheMetricsCollector,
    )
    self.metrics_collector = radix_cache_cls(labels=labels)

```

python/sglang/srt/managers/schedule_policy.py

cache-mode 准入侧 reason 拆分: _account_prefill_cache_admission() 将非 device_capacity 的准入缺口从 shrunk 改为 cache_admission_shortfall, 与 anchor_lost 形成两个可区分的 attempt 级诊断。

schedule_policy.py —— cache-mode 准入后的统计结算:

L3 命中的 token 若未全部兑现为可用 prefill 前缀, 按缺口原因记账。

```
def _account_prefill_cache_admission(self, req: Req, prefix_len: int) -> None:
```

```
    if req.retracted_stain:
```

```
        # 回缩归属有意暂不计入统计; 丢弃其生命周期状态, 避免后续
```

```
        # abort 误报为 drop。
```

```
        self.tree_cache.discard_storage_prefetch_accounting(
```

```
            req.cache_request_handle
```

```
        )
```

```
    return
```

```
if prefix_len > 0:
```

```
    device_hit, host_hit, storage_hit = split_cached_prefix_by_tier(
```

```
        prefix_len=prefix_len,
```

```
        host_hit_len=req.materialized_host_hit_len(),
```

```
        storage_hit_len=req.storage_hit_length,
```

```
        storage_hit_start=req.storage_hit_start,
```

```
        host_hit_is_storage=req.host_hit_is_storage,
```

```
    )
```

```
    self.log_device_hit_tokens += device_hit
```

```
    self.log_host_hit_tokens += host_hit
```

```
    self.log_storage_hit_tokens += storage_hit
```

```
fulfilled_storage_hit = req.fulfilled_storage_hit_len(prefix_len)
```

```
reason = None
```

```
if fulfilled_storage_hit < req.storage_hit_length:
```

```
    # 拆分后的语义: host 容量不足仍记 device_capacity; 其余不足
```

```
    # 路径统一归为 cache_admission_shortfall, 表示 L3 加载到 L2 的
```

```
    # span 在 cache-mode 准入时已不可复用。这是 attempt 级诊断,
```

```
    # 不是最终 prefill cache-miss 计数。
```

```
    reason = (
```

```
        "device_capacity"
```

```
        if req.needs_host_load_back()
```

```
        and req.host_loaded_length < req.host_hit_length
```

```
        else "cache_admission_shortfall"
```

```
    )
```

```
self.tree_cache.finish_storage_prefetch_admission(
```

```
    req.cache_request_handle,
```

```
    fulfilled_tokens=fulfilled_storage_hit,
```

```
    reason=reason,
```

```
)
```

评论区精华

本 PR 没有 reviewer 行内评论，hzh0425 直接 APPROVED（无正文）。核心设计取舍记录在作者 xiezhq-hermann 的两条 rebase 评论中：

- 第一次 rebase 到 main (2fca6d69aa) 解决与 #39318 (attempt-scoped CacheRequestHandle) 的冲突: "the split discard reasons (aux_window_trim / device_overlap / anchor_lost) now key their release and finish calls by the request handle", 即 reason 记账按 request handle 组织。
- 第二次 rebase 到 #39283 / #39480 合并后的 main: "#39283 replaced the admission-time splice trim with a re-fetch (prepare_staged_prefetch / _refetch_staged)... the aux_window_trim / device_overlap outcomes this PR originally introduced have no code path left; this rebase drops the helper and keeps the two remaining splits (anchor_lost at IO-commit anchor loss, cache_admission_shortfall in the cache-mode adder) plus the per-rank L2 labels".

关键结论: label 预置保留但无代码路径, 是为了让既有 dashboard 的 label 集合保持稳定; _refetch_staged 仍会把 shrunk 重取报为 shrunk。这两条评论是作者在 rebase 过程中主动收缩方案范围, 避免提交死代码。

- aux_window_trim / device_overlap 因 #39283 重构失去代码路径的处理 (design): 保留 aux_window_trim / device_overlap 的 label 预置与文档描述 (作为历史语义), 实际只保留 anchor_lost 与 cache_admission_shortfall 两个新拆分; _refetch_staged 继续把 shrunk 重取报为 shrunk。
- 与 #39318 attempt-scoped CacheRequestHandle 的冲突解决 (design): rebase 完成, discard reason 记账统一以 CacheRequestHandle 为键, 测试同步更新。
- 指标语义文档化: per-shard vs logically replicated、attempt-level vs final miss (documentation): 文档已按新口径重写, 并在每个计数器上标明读取方式。

风险与影响

- 风险:
 1. 指标 series 基数上升: 每个 tp/pp/dp rank 组合都会产生独立 series, 在大规模 TP/PP/DP 集群下 Prometheus 序列数量会成倍增长; 这是修复指标正确性所必需的代价, 但监控存储与查询开销需关注。
 2. 既有 dashboard 不兼容: 旧 reason="shrunk" 被拆分为 anchor_lost / cache_admission_shortfall, 依赖旧标签的告警规则和面板需要同步更新; 好在 _refetch_staged 仍输出 shrunk, 系列不会完全断流。
 3. 文档与代码路径偏差隐患: aux_window_trim / device_overlap 仅存在于计数器文档和 label 预置中, 当前没有任何代码路径产生这两个值; 未来若相关行为回归, 需同步清理文档, 否则会误导排障人员。
 4. 初始化时序: BasePrefixCache.init_metrics_collector() 依赖 get_parallel() 与 is_dp_attention_enabled() 在缓存初始化时已就绪; 若未来缓存初始化早于 parallel 状态装配, 可能取到错误的 rank。
 5. 影响面有限: 纯指标路径变更, 不触碰推理数值 (PR 声明 metrics-only), 回归风险集中在 collector 初始化和 reason 字符串的测试断言上。 - 影响: 对运维与平台团队:

HiCache L2 (本机 DRAM) 与 L3 (远端存储) 指标首次可按 rank 对齐, 能正确计算物理总量、逐 rank 差异和活跃带宽, 为容量规划与带宽瓶颈定位提供准确数据基础; 但 dashboard 与告警需适配新 rank 标签和拆分后的 reason。对系统: 纯可观测性改动, 无推理路径数值影响。对团队: 这是 HiCache 分层缓存可观测性建设的关键一环, 与 Router 侧存储感知路由指标形成互补。 - 风险标记: 指标 series 基数随 rank 组合增长, reason 标签拆分影响既有 dashboard, 保留无代码路径的 label 预置, 纯指标改动无数值回归面

关联脉络

- PR #39283 [HiCache] Rework the buffer-mode storage prefetch pipeline and retry bookkeeping: 本 PR 的 reason 拆分直接依赖 #39283 的行为变化: admission 期 splice trim 被 re-fetch 替代后, aux_window_trim / device_overlap 失去代码路径, 最终只保留 anchor_lost 与 cache_admission_shortfall 两个拆分。
- PR #39480 [HiCache] Optimize buffer-mode storage existence bookkeeping: 作者第二次 rebase 说明中明确提到基于 #39283 与 #39480 合并后的 main 重做; #39480 优化了 buffer-mode 存储存在性记账, 与 cache_admission_shortfall 的准入判断链路相关。
- PR #39318 attempt-scoped CacheRequestHandle: 作者第一次 rebase 评论明确提及 #39318: 本 PR 的 discard reason 按 CacheRequestHandle 组织 release/finish 调用, 测试断言也基于 handle。该 PR 不在本次历史列表中, 但 rebase 记录是可靠证据。
- PR #39109 [Router] Expose the KV storage-tier stream and tree occupancy on /metrics: 同为 HiCache 分层缓存可观测性建设: Router 侧暴露存储分层指标, 本 PR 让引擎侧 L2/L3 指标可对齐, 两者共同构成 HiCache 的可观测闭环。