PR #25184 完整报告

sgl-project/sglang

[SMG] Fix cache-aware policy pool isolation in PD mode

合并时间：2026-05-14 11:07

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25184

执行摘要

本 PR 修复了 sgl-model-gateway 在 PD（prefill/decode）模式下 CacheAwarePolicy 缓存感知路由退化为随机翻转的 bug。通过将 trie 键改为 (pool, model) 复合键隔离各池的缓存树，并在 worker 注册 / 移除时正确初始化 / 清理 PD 池的策略实例，恢复了 PD 模式下缓存感知路由的正确行为。

功能与动机

在 PD 模式下，prefill 和 decode 工作者共享同一个 CacheAwarePolicy trie，且 trie 键仅包含 model_id。每个 select_worker 调用末尾的 tree.insert(text, url) 导致针对同一 prompt 交替的 prefill 和 decode 请求互相覆盖租户条目，缓存感知路由退化为随机工作者选择（flip-flop）。本 PR 旨在为 prefill、decode 和 regular 池提供完全隔离的缓存树。

实现拆解

1. 引入池隔离的 trie 键（cache_aware.rs）：新增 pool_tag、make_tree_key、tree_key_for_worker 三个辅助函数，将 trie 的键从单一的 model_id 改为 pool::model 复合字符串。修改 init_workers、add_worker、remove_worker、select_worker 等方法，使其在访问 trie 时使用复合键；移除了不再需要的 add_worker_by_url 方法。
2. 添加 PD 工作者移除分发（registry.rs）：新增 remove_pd_worker_from_cache_aware 方法，根据 WorkerType 将移除请求分发给 prefill_policy 或 decode_policy（而非统一的 model_policies），对 Regular 类型直接返回。
3. 工作者注册时初始化 PD 池策略（update_policies.rs）：在 per-model 的策略初始化循环之后，额外检查 prefill 和 decode 池是否配置了 cache_aware 策略，若是则调用 init_pd_cache_aware_policies 为两个池的 CacheAwarePolicy 实例播种 worker。
4. 工作者移除时同步清理 PD 池（remove_from_policy_registry.rs）：在原有的 remove_worker_from_cache_aware 调用之后，新增对 remove_pd_worker_from_cache_aware 的调用，确保 PD 池的 trie 也同步清理。
5. 添加回归测试（cache_aware.rs tests 模块）：新增三个测试用例：两个池分别使用独立策略时的隔离、单个共享策略回归场景（使用复合键仍保持隔离），以及 PD worker 移除后另一池的 trie 不受影响。

sgl-model-gateway/src/policies/cache_aware.rs

核心变更文件，引入池隔离的 trie 键，修改所有相关方法

```
/// 根据 WorkerType 返回池标签，用于构建隔离的 trie 键
```

```
fn pool_tag(worker_type: &WorkerType) -> &'static str {
    match worker_type {
        WorkerType::Regular => "regular",
        WorkerType::Prefill { .. } => "prefill",
        WorkerType::Decode => "decode",
    }
}

/// 构建复合键 `pool::model`
fn make_tree_key(pool: &str, model: &str) -> String {
    format!("{}::{}", pool, model)
}

/// 从 worker 直接获取其 trie 键
fn tree_key_for_worker(worker: &dyn Worker) -> String {
    make_tree_key(
        pool_tag(worker.worker_type()),
        normalize_model_key(worker.model_id()),
    )
}

// 在 init_workers 中，改为按复合键分组
pub fn init_workers(&self, workers: &[Arc<dyn Worker>]) {
    // 按 (pool, model) 分组，确保每个池拥有独立的 trie
    let mut grouped: std::collections::HashMap<String, Vec<&Arc<dyn Worker>>> =
        std::collections::HashMap::new();
    for worker in workers {
        grouped
            .entry(tree_key_for_worker(worker.as_ref()))
            .or_default()
            .push(worker);
    }
    // 为每个复合键创建或获取对应的 trie 并插入所有 worker URL
    for (tree_key, pool_workers) in grouped {
        let tree = self
            .trees
            .entry(tree_key)
            .or_insert_with(|| Arc::new(Tree::new()));
        for worker in pool_workers {
            tree.insert("", worker.url());
        }
    }
}
```

sgl-model-gateway/src/policies/registry.rs

新增 PD 工作者从 *cache-aware* 策略中移除的分发方法

```
/// 根据 WorkerType 将 PD worker 从对应池的 cache-aware 策略中移除
pub fn remove_pd_worker_from_cache_aware(&self, worker: &dyn Worker) {
```

```
    // 根据 worker 类型选择正确的池策略实例
    let policy = match worker.worker_type() {
        crate::core::WorkerType::Prefill { .. } => self.prefill_policy.get(),
        crate::core::WorkerType::Decode => self.decode_policy.get(),
        crate::core::WorkerType::Regular => return, // 常规 worker 由 model_policies 处理
    };
    if let Some(policy) = policy {
        // 仅当策略为 cache_aware 时才执行移除
        if policy.name() == "cache_aware" {
            if let Some(cache_aware) = policy.as_any().downcast_ref::<CacheAwarePolicy>() {
                cache_aware.remove_worker(worker);
                debug!(
                    "已从 cache-aware 策略中移除 PD worker {} ({})",
                    worker.url(),
                    worker.worker_type()
                );
            }
        }
    }
}
```

评论区精华

本 PR 未产生实质 review 讨论，reviewer slin1237 直接批准了此 PR。PR 描述中详尽分析了根因和修复方案。

风险与影响

- 风险：当前 mesh 同步使用的 apply_remote_tree_operation 仍以 model_id 为键，若启用网格集群且使用 PD + cache-aware，可能存在键不匹配风险；复合键变更导致运行中升级需重启 worker 重建 trie；remove_pd_worker_from_cache_aware 的正确性依赖 WorkerType 正确设置。
- 影响：修复了 PD 模式下缓存感知路由失效的问题，使 prefill 和 decode 池的缓存布局互不干扰。仅影响启用了 PD 模式且配置了 cache-aware 策略的用户。对常规 Router 无影响。

关联脉络

此 PR 与近期 sgl-model-gateway 相关的 PR（如 #24375 K8s 集成测试、#24719 PyO3 绑定）无直接关联，属于独立的 bugfix 工单。后续可关注 mesh 同步侧是否需适配复合键以支持完整集群场景。