

PR #27612 完整报告

sgl-project/sglang

[router] Add /flush_cache endpoint to experimental sgl-router

合并时间: 2026-06-09 09:53

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/27612>

执行摘要

该 PR 为 experimental sgl-router 新增了 `/flush_cache` 管理端点，允许运维人员一次性刷新所有 worker 的 KV 缓存。实现包含新的路由处理文件、注册表扩展、路由挂载和单元测试。这是一个标准的管理功能添加，设计上考虑了并发限制、错误报告和断路器绕过，但存在无法区分延迟刷新以及可能影响断路器状态的风险。

功能与动机

根据 PR body，该功能的动机是 'Add an endpoint to flush the cache for all the workers to the experimental router'，目的是提供一个中心化的缓存刷新接口，避免运维人员逐一登录 worker 操作。这与 sgl-model-gateway 中的 `/flush_cache` 功能一致。

实现拆解

1. 新增路由处理文件 experimental/sgl-router/src/server/routes/cache.rs，定义了 FlushCacheResult 和 FailedWorker 序列化结构体，核心函数 flush_cache 获取全量 worker 并调用 fan_out_flush 并发发送 POST 请求。fan_out_flush 使用 buffer_unordered(32) 限制并发，并直接使用共享 request::Client 绕过断路器。
2. 扩展 WorkerRegistry 在 experimental/sgl-router/src/workers/registry.rs 中添加 pub fn all(&self) -> Vec<Arc<Worker>> 方法，返回所有注册 worker 的克隆列表（包括不同模式和模型的 worker）。同时新增两个单元测试验证跨模式全量返回和空注册表行为。
3. 挂载路由在 experimental/sgl-router/src/server/app.rs 的 build_router 中通过 route("/flush_cache", post(crate::server::routes::cache::flush_cache)) 注册新端点。
4. 导出模块在 experimental/sgl-router/src/server/routes/mod.rs 中添加 pub mod cache; 声明。
5. 测试覆盖在 cache.rs 中包含了 10 个测试用例，包括全部成功返回 200、部分失败返回 502 并带 breakdown、空 fleet、不可达 worker、非 5xx 状态（如 404）处理、尾部斜杠 URL、以及不同模式 worker 等场景。

experimental/sgl-router/src/workers/registry.rs

基础数据结构扩展：新增 all() 方法，支持全量 worker 快照，是 fan-out 的前提。

```
/// Snapshot of every registered worker, across all models and modes. /// Used by
fleet-wide admin fan-out (e.g. `/flush_cache`) that targets /// every worker the router
knows about rather than one model's pool. /// Order is unspecified (iterates the
```

```
underlying `DashMap`). pubfnall(&self)->Vec<Arc<Worker>>{  
self.by_id.iter().map(|e|Arc::clone(e.value())).collect() } 对应的测试: #[test]  
fnall_returns_every_worker_across_models_and_modes(){  
let r=WorkerRegistry::default(); let _=r.add(spec("w1",WorkerMode::Plain,&["m1"]));  
let _=r.add(spec("p",WorkerMode::Prefill,&["m2"]));  
let _=r.add(spec("d",WorkerMode::Decode,&["m2"]));  
let mut ids:Vec<String>=r.all().into_iter().map(|w|w.id.0.clone()).collect(); ids.sort();  
assert_eq!(ids,vec!["d","p","w1"]); } #[test] fnall_is_empty_for_fresh_registry(){  
assert!(WorkerRegistry::default().all().is_empty()); }
```

评论区精华

当前 PR 没有 review 评论，但 PR body 中详细说明了两个关键设计决策：

- 绕过断路器：管理刷 nova 需要触及每个 worker，包括断路器打开的 worker；同时避免在断路器状态中记录管理操作。
- 已知局限：SGLang worker 的 `/flush_cache` 在等待现有请求时会返回 200 但延迟执行，本端点无法区分立即成功和延迟执行，任何非 400 响应视为成功。

风险与影响

- 风险：
 - 绕过断路器可能使运维人员忽略 worker 的实际健康状态，并可能干扰断路器的正常状态机。
 - 无法区分延迟刷新，运维人员可能误判操作效果。
 - 缺少对数千 worker 集群的并发测试，可能出现超时。
- 影响：仅影响 experimental 路由器的管理接口，不影响核心请求路径；需要适当网络隔离以防止未授权访问。

关联脉络

该 PR 是 experimental router 管理能力补全的一部分，与 sgl-model-gateway 的 `/flush_cache` 功能对标。近期仓库中也有多个管理 / 运维相关的 PR（如 #27363 添加指标），表明团队在增强 SGLang 的可运维性。