

PR #33527 完整报告

sgl-project/sglang

[FIX] [benchmark] Fix flush_cache failure after warmup by waiting for server idle

合并时间: 2026-08-05 21:27

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/33527>

执行摘要

- 一句话: 防 race condition: flush_cache 增加 timeout 等待服务空闲
- 推荐动作: 简单且聚焦的 bugfix, 适合快速阅读。它展示了如何处理客户端与服务端之间的时序竞态, 以及如何通过参数让服务端主动等待空闲。对 benchmark 使用者和 CI 维护者有价值。

功能与动机

PR body 中明确说明: Fix race condition where flush_server_cache gets HTTP 400 right after benchmark warmup. Client warmup gather complete does not guarantee the server scheduler is fully idle. Add timeout param to let server wait until is_fully_idle() passes before flushing cache. 也就是说, benchmark 脚本在 warmup 结束后立即调用 flush_cache, 但服务端调度器可能还在处理残留状态, 导致请求被拒绝, 通过 timeout 参数让服务端等待空闲, 从根源上解决时序竞态。

实现拆解

1. 修改 python/sglang/benchmark/serving.py 中的 flush_server_cache 函数: 原先直接 post 请求, 现在针对非 vllm 后端 (即 sglang 系列) 构造 params={"timeout": 10.0}, 并作为 query 参数传给 /flush_cache 端点。vllm 后端对应 /reset_prefix_cache 不传 timeout。
2. 同步更新 test/registered/bench_fn/test_benchmark_datasets_api.py 中的 test_embedding_cache_flush_uses_the_engine_specific_endpoint 单元测试, 断言 vllm 调用时 params 为空字典, sglang 调用时 params 为 {"timeout": 10.0}, 确保新行为的回归覆盖。
3. 该改动只影响 benchmark 脚本的缓存清理路径, 不涉及推理运行时逻辑, 风险面小。

关键文件:

- python/sglang/benchmark/serving.py (模块 基准脚本; 类别 source; 类型 core-logic; 符号 flush_server_cache): 核心修改文件, flush_server_cache 函数增加 timeout 参数, 解决 warmup 后 flush_cache 的竞态失败。
- test/registered/bench_fn/test_benchmark_datasets_api.py (模块 基准测试; 类别 test; 类型 test-coverage; 符号 test_embedding_cache_flush_uses_the_engine_specific_endpoint): 更新测试断言, 确保 vllm 和 sglang 后端在 flush_server_cache 时传递不同参数

，防止回归。

关键符号：flush_server_cache

关键源码片段

python/sglang/benchmark/serving.py

核心修改文件，flush_server_cache 函数增加 timeout 参数，解决 warmup 后 flush_cache 的竞态失败。

```
def flush_server_cache(base_url: str, backend: str) -> None:
    """Flush an engine's prefix cache after benchmark warmup.

    SGLang 服务端在 warmup 刚结束时可能还没完全空闲，直接 flush 会收到
    HTTP 400，因此对非 vllm 后端传入 timeout 查询参数，让服务端在超时
    窗口内轮询 is_fully_idle()，等调度器清理完残留状态后再刷新缓存。
    """
    # vllm 使用 /reset_prefix_cache，其余（sglang 系）使用 /flush_cache
    cache_endpoint = (
        "/reset_prefix_cache" if backend.startswith("vllm") else "/flush_cache"
    )
    # 只有 sglang 后端需要 timeout，vllm 后端不支持该参数，传空 dict 保持原行为
    params = {"timeout": 10.0} if not backend.startswith("vllm") else {}
    response = requests.post(
        base_url + cache_endpoint,
        headers=get_auth_headers(),
        params=params,
    )
    response.raise_for_status()
```

评论区精华

review 只有 sglang-npu-bot 的 APPROVED，无实质的人工讨论。Issue 评论主要是 CI 重跑命令 (/rerun-failed-ci)，说明 CI 曾失败过，但最终通过。

- 暂无高价值评论线程

风险与影响

- 风险：主要风险包括：硬编码 timeout=10.0 可能在服务极慢时不足，但也可能掩盖服务长时间无响应的问题；参数以 query string 传递，需确认服务端 /flush_cache 支持 timeout 参数（从代码上下文看是支持的，否则会被忽略）；本次只更新了单元测试 mock 断言，没有集成测试验证真实服务端行为。整体风险较低，因为只影响 benchmark 工具。
- 影响：影响 benchmark 工具的运行稳定性，特别是 CI 中跑 benchmark 的场景，减少偶发失败；对生产服务无影响，因为不涉及运行时逻辑。团队维护 benchmark 脚本的人需要知道 flush 现在可能等待最多 10 秒。
- 风险标记：硬编码 timeout 值，单测未覆盖真实服务端，参数可能被忽略

关联脉络

- 暂无明显关联 PR