

PR #31576 完整报告

sgl-project/sglang

[EPD] Make encoder register/unregister health-check robust

合并时间: 2026-07-21 22:36

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/31576>

执行摘要

- 一句话: 增强编码器健康检查, 被驱逐 URL 自动复活
- 推荐动作: 值得精读。该 PR 展示了如何优雅处理分布式组件健康检查中的瞬态故障, 其“驱逐 - 探测 - 复活”模式可复用于其他需要自动恢复的服务发现场景。注意 TTL 的默认值应根据实际环境调整。

功能与动机

繁忙的编码器可能因为负载高而错过一次短暂的 2s 健康检查探针, 旧行为在连续 3 次失败后永久丢弃该 URL, 即使编码器实际健康。修改后持续探测被驱逐的 URL, 一旦恢复健康自动重新注册。

实现拆解

1. 新增环境变量 `SGLANG_ENCODER_BOOTSTRAP_EVICTED_TTL` (python/sglang/srt/envron.py), 默认 600 秒, 控制被驱逐 URL 永久抛弃的 TTL; 0 表示永远探测。
2. 重构健康检查状态 (python/sglang/srt/disaggregation/encode_receiver.py): 将 `_consecutive_failures` 拆分为 `_health_fail_counts` (持续失败计数) 和 `_evicted_urls` (驱逐时间戳字典), 并在 `__init__` 中初始化 `_evicted_ttl`。
3. 调整 `register` 和 `unregister` 行为: `register` 时清除该 URL 在 `_evicted_urls` 中的记录; `unregister` 时同时从 `_urls` 和 `_evicted_urls` 移除, 并正确返回布尔值。
4. 修改健康检查循环 (`_health_check_loop`): 对 `_urls` 和 `_evicted_urls` 中的 URL 均执行探测; 探测成功将 URL 从 `_evicted_urls` 移回 `_urls`; 超过 TTL 的 URL 被永久丢弃。
5. 未添加专用测试: 本次变更未包含对应的单元测试或集成测试文件。

关键文件:

- python/sglang/srt/disaggregation/encode_receiver.py (模块 EPD; 类别 source; 类型 core-logic; 符号 init, register, unregister, _health_check_loop): 核心变更文件, 实现健康检查驱逐及自动复活逻辑, 包括 `_evicted_urls` 集合、`_health_fail_counts` 计数器和 TTL 检查。
- python/sglang/srt/envron.py (模块 配置; 类别 source; 类型 configuration; 符号 SGLANG_ENCODER_BOOTSTRAP_EVICTED_TTL): 新增环境变量 `SGLANG_ENCODER_BOOTSTRAP_EVICTED_TTL`, 用于控制被驱逐 URL 的存活时间。

关键符号: init, register, unregister, _health_check_loop, _probe

关键源码片段

python/sclang/srt/disaggregation/encode_receiver.py

核心变更文件, 实现健康检查驱逐及自动复活逻辑, 包括 `_evicted_urls` 集合、`_health_fail_counts` 计数器和 TTL 检查。

```
# 文件: python/sclang/srt/disaggregation/encode_receiver.py
# 以下为 __init__ 中关键初始化片段
self._evicted_ttl = (
    evicted_ttl
    if evicted_ttl is not None
    else envs.SCLANG_ENCODER_BOOTSTRAP_EVICTED_TTL.get()
)
# 超过此连续失败次数后进入 evicted 集合
self._health_fail_threshold = 3
# 连续失败计数
self._health_fail_counts: Dict[str, int] = {}
# 被驱逐的 URL -> 驱逐时的时间戳 (用于 TTL 判断)
self._evicted_urls: Dict[str, float] = {}
```

register 方法: 注册 URL, 同时从 evicted 集合中移除

```
def register(self, url: str) -> bool:
    with self._lock:
        # 清除失败计数和 evicted 状态
        self._health_fail_counts.pop(url, None)
        self._evicted_urls.pop(url, None)
        if url not in self._urls:
            self._urls.append(url)
            logger.info(f"Registered encoder URL: {url}")
            return True
        logger.debug(f"Encoder URL already registered: {url}")
        return False
```

unregister 方法: 主动取消注册, 同时从 evicted 集合移除

```
def unregister(self, url: str) -> bool:
    with self._lock:
        # 检查 URL 是否存在于任一集合
        removed = url in self._urls or url in self._evicted_urls
        if url in self._urls:
            self._urls.remove(url)
        # 确保从 evicted 集合中移除, 防止自动复活
        self._evicted_urls.pop(url, None)
        self._health_fail_counts.pop(url, None)
        if removed:
            logger.info(f"Unregistered encoder URL: {url}")
        return removed
```

评论区精华

1. gemini-code-assist 指出的 unregister 返回值问题：当 URL 仅在 _evicted_urls 中时，原实现返回 False，但实际已从 evicted 集合中删除。最终代码通过 removed 变量跟踪两种集合的存在性并返回正确结果。
 2. liusy58 建议增加 revival deadline：评论指出如果没有 TTL，_evicted_urls 中的 URL 只能通过显式 unregister 清除。该建议被采纳，添加了 SGLANG_ENCODER_BOOTSTRAP_EVICTED_TTL 环境变量。
 3. liusy58 和 ShangmingCai 均批准：整体设计获认可，CI 通过后合并。
- unregister 返回值在 evicted URL 情况下不正确 (correctness): 采用建议，最终代码使用 removed = url in self._urls or url in self._evicted_urls 并返回 removed。
 - 增加被驱逐 URL 的 revival deadline (TTL) (design): 作者添加了 _evicted_ttl 属性并从环境变量 SGLANG_ENCODER_BOOTSTRAP_EVICTED_TTL 获取默认值 600 秒。

风险与影响

- 风险：
 1. 性能风险：健康检查循环现在需要探测 _urls 和 _evicted_urls 两个集合，可能增加网络开销，尤其在大量 URL 被驱逐时。默认 TTL 600 秒可以控制数量。
 2. 兼容性风险：新增环境变量不影响现有部署，但默认行为从“3 次失败永久剔除”变为“3 次失败后进入 evicted 集合并继续探测 600 秒”，可能改变用户对编码器失败的期望。
 3. 测试缺失风险：无专用测试覆盖驱逐复活逻辑，未来回归可能不易察觉。 - 影响：影响范围：EPD (Efficient Parallel Disaggregation) 场景下的编码器健康管理。用户将看到编码器在短暂压力后自动恢复，减少人工干预。影响程度：中等，变更仅涉及 EncoderBootstrapServer 类，不影响其他模块。 - 风险标记：核心路径变更，缺少测试覆盖，新增配置可能影响默认行为

关联脉络

- 暂无明显关联 PR