

PR #28143 完整报告

sgl-project/sglang

fix(health): make health-check rid unique across tokenizer workers

合并时间: 2026-06-14 15:01

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/28143>

执行摘要

- 一句话: 修复健康检查 ID 跨进程重复 bug
- 推荐动作: 值得立即合并。改动小、风险低、针对一个已复现的生产 bug。可以作为多 worker 部署的最佳实践参考——始终使用 uuid 或 pid + counter 保证请求 ID 跨进程唯一。

功能与动机

Issue #28142 报告了在 `--tokenizer-worker-num 8` 的多 worker 模式下, 健康检查频繁抛出 `KeyError: 'HEALTH_CHECK_1781301554.5131953'`。根本原因是 `time.time()` 精度不足以区分同一时刻多个 worker 的请求, 导致 rid 碰撞, 进而使 `BatchTokenIDOutput` 中出现重复 rid, `DetokenizerManager` 无法找到对应的 decode status 而崩溃。

实现拆解

这是一个极简的两步修复:

1. 在 `http_server.py` 中替换 rid 生成 (第 610→613 行): 在 `/health` 和 `/health_generate` 端点中, 将 `rid = f"{HEALTH_CHECK_RID_PREFIX}_{time.time()}"` 改为 `rid = f"{HEALTH_CHECK_RID_PREFIX}_{uuid.uuid4().hex}"`, 并添加说明注释。新增了 `import uuid`。改动量仅 +4/-1。
2. 在 `encode_server.py` 中同步修复 (第 2554→2555 行 和 第 3761→3762 行): 同样的 rid 生成方式替换, 覆盖 DP worker 内部健康编码 (`_dp_worker_health_encode`) 和 `encode dispatcher` 侧的 `health_generate` 函数。新增了 `import uuid`。改动量 +5/-2。

没有引入配置、测试或文档变更。

关键文件:

- `python/sglang/srt/entrypoints/http_server.py` (模块 HTTP 入口; 类别 source; 类型 dependency-wiring): 主入口 HTTP server, 是健康检查端点的核心实现, 修复直接影响所有 `/health` 和 `/health_generate` 请求的 rid 唯一性。
- `python/sglang/srt/disaggregation/encode_server.py` (模块 编码服务器; 类别 source; 类型 dependency-wiring; 符号 `_dp_worker_health_encode`, `health_generate`): 分解模式下的 `encode server`, 包含 DP worker 内部健康编码和 `dispatcher` 端的健康检查, 两处都需要唯一 rid。

关键符号: `health_generate`, `_dp_worker_health_encode`

关键源码片段

python/sglang/srt/entrypoints/http_server.py

主入口 HTTP server，是健康检查端点的核心实现，修复直接影响所有 `/health` 和 `/health_generate` 请求的 rid 唯一性。

```
# python/sglang/srt/entrypoints/http_server.py

import uuid # 新增导入

@app.get("/health")
@app.get("/health_generate")
async def health_generate(request: Request) -> Response:
    """
    Check the health of the inference server by sending a special request
    to generate one token.
    """
    # ... 前置检查省略 ...

    sampling_params = {"max_new_tokens": 1, "temperature": 0.0}
    # uuid 保证 rid 在多个 tokenizer worker 间唯一；
    # 纯 time.time() 在同一纳秒可能碰撞，
    # 导致共享的 DetokenizerManager 抛 KeyError。
    rid = f"{HEALTH_CHECK_RID_PREFIX}_{uuid.uuid4().hex}"

    if _global_state.tokenizer_manager.is_generation:
        gri = GenerateReqInput(
            rid=rid,
            input_ids=[0],
            sampling_params=sampling_params,
            log_metrics=False,
        )
        # ... 分解模式配置 ...
    else:
        gri = EmbeddingReqInput(
            rid=rid,
            input_ids=[0],
            sampling_params=sampling_params,
            log_metrics=False,
        )
    # ... 后续不变 ...
```

python/sglang/srt/disaggregation/encode_server.py

分解模式下的 encode server，包含 DP worker 内部健康编码和 dispatcher 端的健康检查，两处都需要唯一 rid。

```
# python/sglang/srt/disaggregation/encode_server.py

import uuid # 新增导入
```

```

async def _dp_worker_health_encode(enc: MMEncoder) -> None:
    """DP worker 内部的健康探测，执行一次 dummy encode。"""
    # ... 跳过忙碌 worker 和模态选择 ...

    # uuid 保证不同 worker 进程的 rid 不会碰撞；
    # bare time.time() 在并发时可能产生重复。
    req_id = f"{HEALTH_CHECK_RID_PREFIX}_{uuid.uuid4().hex}"
    try:
        _, _, _, error_msg, error_code = await enc.encode(
            mm_items=mm_items,
            modality=modality,
            req_id=req_id,
            num_parts=1,
            part_idx=0,
        )
    finally:
        enc.embedding_to_send.pop(req_id, None)
    # ... 错误处理 ...

# dispatcher 侧的 /health_generate 端点同样替换
# req_id = f"{HEALTH_CHECK_RID_PREFIX}_{uuid.uuid4().hex}"

```

评论区精华

无实质性 review 讨论。自动化 bot `gemini-code-assist[bot]` 仅做了总结性评论，没有提出修改意见。第二位提交者 `hnyls2002` 在第二次 commit 中将最初的 `os.getpid()` 方案改为了 `uuid.uuid4().hex`，并补充了 `encode_server.py` 的覆盖。

- 暂无高价值评论线程

风险与影响

- 风险：风险极低。uuid4 碰撞概率在可忽略范围内，且 rid 仅用于内部路由和 `decode status` 查找，不依赖单调性或可读性。唯一潜在风险是 uuid 生成相比单调递增 ID 或 `f"{pid}_{time_ns()}"` 可能带来微小的额外开销，但在健康检查低频场景下完全可接受。返回的 Response 内容和逻辑流程未变。
- 影响：直接修复了多 tokenizer worker 模式下健康检查的稳定性问题，影响用户为所有使用 `--tokenizer-worker-num > 1` 的部署，包括 DP 分解（disaggregation）场景。对其他单 worker 配置无影响。修复本身不涉及模型推理路径，不触发回归。
- 风险标记：低风险

关联脉络

- PR #28142 [Bug] `KeyError: 'HEALTH_CHECK_1781301554.5131953'`: 直接关联的 issue，报告了本 PR 修复的 bug，提供了复现环境和错误日志。