PR #25542 完整报告

sgl-project/sglang

Fix PD disaggregation warmup: set request_name and improve error logging

合并时间：2026-05-19 00:49

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25542

执行摘要

- 一句话：修复 PD 预热 endpoint 和日志
- 推荐动作：值得合并，修复了明显的 bug。但建议作者或后续 PR 跟进处理 review 指出的预热失败时函数返回值问题，以保持与非分离路径的行为一致。

功能与动机

PD 分离部署的预热请求因未设置 request_name 使用了错误的 endpoint（此前定义的 request_name 仅在非分离路径中赋值），导致预热可能失败或行为异常。同时，失败日志没有包含分离模式信息，不利于排查问题。

实现拆解

1. 设置 endpoint 名称：在 _execute_server_warmup 函数的分离模式分支（else 块）开头添加 request_name = "/generate"，确保预热请求发送到 /generate 端点。
2. 改进错误日志：将日志消息从 "Prefill disaggregation mode warm Up Failed, status code: {}".format(res.status_code) 改为 "Disaggregation warmup failed (mode=%s), status code: %s", server_args.disaggregation_mode, res.status_code，使用 logger 的格式化参数并包含分离模式名。
3. 仅修改一个文件：所有变更集中在 python/sglang/srt/entrypoints/http_server.py，共 4 行新增、3 行删除。

关键文件：

- python/sglang/srt/entrypoints/http_server.py（模块 入口；类别 source；类型 core-logic；符号 _execute_server_warmup）：所有变更均在此文件中，修复了 PD 分离预热的核心逻辑和日志。

关键符号：_execute_server_warmup

关键源码片段

python/sglang/srt/entrypoints/http_server.py

所有变更均在此文件中，修复了 PD 分离预热的核心逻辑和日志。

```
# 位于 _execute_server_warmup 函数内部
else:
```

```
    logger.info(f"Start of pd disaggregation warmup ...")
    # 修复：显式设置 endpoint 为 /generate，确保预热请求路由正确
    request_name = "/generate"
    json_data = {
        "sampling_params": {
            "temperature": 0.0,
            "max_new_tokens": 8,
            "ignore_eos": True,
        },
        "bootstrap_host": [FAKE_BOOTSTRAP_HOST] * server_args.dp_size,
        "bootstrap_room": [
            i * (2**63 // server_args.dp_size) + (i % server_args.tp_size)
            for i in range(server_args.dp_size)
        ],
        "input_ids": [[10, 11, 12, 13]] * server_args.dp_size,
    }
    res = requests.post(url + request_name, json=json_data, ...)
    if res.status_code == 200:
        logger.info("Disaggregation warmup request completed with status %s, resp: %s",
                    res.status_code, res.json())
        _global_state.tokenizer_manager.server_status = ServerStatus.Up
    else:
        # 修复：使用 logger 参数格式化并包含 disaggregation_mode，便于排查
        logger.info(
            "Disaggregation warmup failed (mode=%s), status code: %s",
            server_args.disaggregation_mode,
            res.status_code,
        )
        _global_state.tokenizer_manager.server_status = ServerStatus.UnHealthy
```

评论区精华

代码审查机器人指出，分离模式预热失败时（res.status_code != 200），函数并未抛出异常或返回 False，导致服务器后续仍输出 "The server is fired up and ready to roll!"，尽管状态已设为 UnHealthy。这与非分离路径的行为不一致。该问题在本 PR 中未解决。

- 预热失败后服务器状态不一致 (correctness): 未解决，本 PR 仅修复 endpoint 和日志，未处理该逻辑缺陷。

风险与影响

- 风险：低风险。变更仅影响分离部署预热路径，且改动量小。但如 review 所述，预热失败时服务器仍可能误报启动成功，存在逻辑缺陷。
- 影响：影响范围限于 PD 分离部署（server_args.disaggregation_mode != "null"）的预热阶段。修复后预热端点正确、日志更清晰，便于运维排障。
- 风险标记：预热失败逻辑不完整

关联脉络

- 暂无明显关联 PR