

PR #31543 完整报告

sgl-project/sglang

[PD] pool decode bootstrap HTTP sessions

合并时间: 2026-07-27 21:07

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/31543>

执行摘要

- 一句话: 复用 bootstrap HTTP 会话避免端口耗尽
- 推荐动作: 建议快速合入。本 PR 以极小的代码改动 (仅一个文件) 解决了 PD 部署中严重的生产问题, A/B 验证数据充分。值得关注的设计决策: 使用 `threading.local()` 保证 Session 线程安全、小连接池 (`pool=1`) 避免资源浪费。

功能与动机

PD 解耦部署下 decode 调度器每轮事件循环都会为每个未解析 `dp_rank` 的请求发起 `GET /route` 和 `POST /query_dp_ranks` 查询, 每次使用全新的短连接。高并发时大量 TIME-WAIT 套接字耗尽本地临时端口 (`[Errno 99] Cannot assign requested address`), 导致后续查询失败、请求永久无法解析。

实现拆解

1. 引入线程本地 Session 池: 在 `python/sglang/srt/disaggregation/common/conn.py` 模块顶部新增 `_bootstrap_sessions = threading.local()` 和 `_get_bootstrap_session(bootstrap_addr)` 函数。函数内部以 `threading.local()` 对象存储每个 `bootstrap_addr` 对应的 `requests.Session`, 确保每个线程持有自己的持久连接 (`requests.Session` 非线程安全)。
2. 配置 HTTPAdapter: `session.mount("http://", HTTPAdapter(pool_connections=1, pool_maxsize=1))`, 限制每个地址的连接池大小为 1, 避免不必要的连接膨胀; 仅挂载 `http://` 协议 (bootstrap 只走 http)。
3. 替换两处调用点:
 - `_get_bootstrap_info_from_server` 中原 `requests.get(url, timeout=5)` 改为 `_get_bootstrap_session(self.bootstrap_addr).get(url, timeout=5)`。
 - `query_prefill_dp_ranks` 中原 `requests.post(url, ...)` 改为 `_get_bootstrap_session(bootstrap_addr).post(url, ...)`。
 - 两组调用的 URL、payload、超时、错误处理均未变更, 仅连接方式从短连接变为连接池复用。
4. 删除冗余请求头: 回顾讨论后删除了 `Connection: keep-alive` 请求头 (连接池已自动保活)。测试与配置无需改动, 原 A/B 验证已覆盖回归。

关键文件:

- python/sglang/srt/disaggregation/common/conn.py (模块 解耦模块; 类别 source; 类型 core-logic; 符号 `_get_bootstrap_session`): 唯一变更文件, 新增 `_bootstrap_sessions` 线程局部变量和 `_get_bootstrap_session` 函数, 改造两处请求调用点。

关键符号: `_get_bootstrap_session`

关键源码片段

python/sglang/srt/disaggregation/common/conn.py

唯一变更文件, 新增 `_bootstrap_sessions` 线程局部变量和 `_get_bootstrap_session` 函数, 改造两处请求调用点。

```
# python/sglang/srt/disaggregation/common/conn.py

# ... (其他 import 省略)
import threading

# Thread-local storage for per-address bootstrap sessions.
# `requests.Session` is not safe for concurrent cross-thread use,
# so each thread keeps its own pool.
_bootstrap_sessions = threading.local()

def _get_bootstrap_session(bootstrap_addr: str) -> requests.Session:
    """Return a keep-alive session for the given bootstrap address.

    Sessions are cached per thread in a thread-local dict. The number of
    distinct bootstrap_addr values is bounded (one per prefill server),
    so eviction is not needed. Only http:// is mounted because bootstrap
    is HTTP-only.
    """
    # Access the per-thread dict; lazy-init if not present
    sessions = getattr(_bootstrap_sessions, "by_addr", None)
    if sessions is None:
        sessions = {}
        _bootstrap_sessions.by_addr = sessions

    session = sessions.get(bootstrap_addr)
    if session is None:
        # Create a new Session with a small connection pool (1 connection per addr)
        session = requests.Session()
        session.mount(
            "http://",
            requests.adapters.HTTPAdapter(pool_connections=1, pool_maxsize=1)
        )
        sessions[bootstrap_addr] = session
    return session
```

同时，在 `_get_bootstrap_info_from_server` 和 `query_prefill_dp_ranks` 两处方法中，将原有的 `requests.get(url, ...)` 和 `requests.post(url, ...)` 替换为 `_get_bootstrap_session(addr).get(...)` / `_get_bootstrap_session(addr).post(...)`，其余逻辑（URL、超时、异常处理）保持不变。

评论区精华

1. Session 线程安全性：kkHuang-amd 指出 `requests.Session` 不是线程安全的，初始实现使用共享字典，第二个提交将存储改为 `threading.local()`，修复了潜在的并发 bug。
 2. 连接池配置：讨论中提及 `Connection: keep-alive` 头在连接池模式下冗余，后续提交删除。
 3. Session 清理策略：明确不主动 evict session，因为 `bootstrap_addr` 数量有限（一个 prefill 服务器对应一个地址），且仅挂载 http 协议。
- Session 线程安全性 (correctness): 第二个提交将 Session 存储改为 `threading.local()`，每个线程持有独立 Session，修复了潜在并发 bug。
 - 连接池配置与冗余头 (design): 删除了冗余头，提交者添加注释说明设计选择。

风险与影响

- 风险：
 - 回归风险低：改动仅将 `requests.get/post` 替换为 `_get_bootstrap_session().get/post`，URL、参数、超时、异常处理保持不变，行为等价。
 - 线程安全：使用 `threading.local()` 确保每线程独立 Session，避免共享状态竞争。
 - 连接泄漏风险：Session 数量与 `bootstrap_addr` 数量一致（通常 1~N 个 prefill server），不会无限增长。
 - 测试覆盖缺失：当前无单元测试覆盖新函数，依赖 CI 集成测试（A/B 验证通过）。
- 影响：
 - 用户：高并发 PD 部署不再出现 `Cannot assign requested address` 错误，请求可正常完成；TTFT 和吞吐量有小幅改善（-9% TTFT p99, +4% 吞吐）。
 - 系统：decode 节点到 bootstrap 的 TCP 连接从每请求新建变为长期复用，TIME-WAIT 套接字从约 10,100 降到 0，降低系统负担。
 - 团队：改动量极小（+26/-2），风险低，可快速合入。
- 风险标记：核心路径变更，缺少测试覆盖

关联脉络

- PR #31629 [Fix] Enable graph capture and MSCCL++ for attention TP groups: 同属 disaggregation 模块的 bug 修复，改进了 PD 场景的稳定性。
- PR #32071 [BugFix][EPD] Fix Mooncake source-MR lifecycle for multi-TP /send: 另一个 PD/disaggregation 相关的 bug 修复，解决多 TP rank 并发问题。