

PR #28573 完整报告

sgl-project/sglang

support MP Server and embedded server for granian to enable multi tokenizer worker

合并时间: 2026-06-19 08:59

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/28573>

执行摘要

- 一句话: HTTP/2 支持多 tokenizer worker
- 推荐动作: 该 PR 值得精读, 尤其是 `_run_granian_server` 的两种模式设计、以及如何通过移除旧环境变量简化代码。设计决策关注: 嵌入式 vs 多进程服务器在不同 tokenizer worker 数下的选择逻辑。

功能与动机

此前 `--enable-http2` 与 `--tokenizer-worker-num > 1` 无法同时使用 (`server_args.py` 中会抛出 `ValueError`)。该 PR 旨在移除这一限制, 让用户在启用 HTTP/2 时也能使用多个 tokenizer worker 实例以提升吞吐。

实现拆解

1. 重构 Granian 启动入口: 在 `python/sglang/srt/entrypoints/http_server.py` 中重写 `_run_granian_server` 函数, 当 `tokenizer_worker_num == 1` 时, 使用 Granian 的嵌入式模式将 ASGI app 作为 asyncio task 运行在当前进程内, 直接复用全局状态; 当 `tokenizer_worker_num > 1` 时, 使用 Granian 的 MP Server 启动多个 worker 进程, 每个 worker 通过共享内存读取参数并调用 `init_multi_tokenizer` 初始化自己的 `TokenizerManager`。
2. 移除旧的环境变量与初始化路径: 删除 `python/sglang/srt/envron.py` 中的 `SGLANG_GRANIAN_PARENT_PID` 配置, 删除 `http_server.py` 中的 `_init_granian_worker` 函数和 `_close_main_process_sockets` 函数, 简化 `get_main_process_id` 逻辑 (`multi_tokenizer_mixin.py`)。
3. 解除 `server_args` 中的组合限制: 在 `python/sglang/srt/server_args.py` 的 `_handle_ssl_validation` 中移除对 `--enable-http2` 与 `--tokenizer-worker-num > 1` 同时使用的 `ValueError` 校验。
4. 新增测试覆盖: 在 `test/registered/openai_server/basic/test_http2_server.py` 中增加 `TestHTTP2ServerMultiTokenizer` 类, 使用 `--enable-http2 --tokenizer-worker-num 2` 启动服务, 并继承基础测试用例, 验证多 tokenizer 路径的请求处理。同时调整 CI 注册的预估时间。

关键文件:

- python/sglang/srt/entrypoints/http_server.py (模块 HTTP 服务; 类别 source; 类型 dependency-wiring; 符号 `_run_granian_server`, `serve`, `lifespan`) : 核心变更文件, 重写 Granian 启动函数 (`_run_granian_server`), 添加两种运行模式; 移除旧的初始化逻辑 (`_init_granian_worker`, `_close_main_process_sockets`); 调整 `lifespan` 控制流。
- python/sglang/srt/server_args.py (模块 配置层; 类别 source; 类型 core-logic; 符号 `_handle_ssl_validation`) : 移除 `--enable-http2` 与 `--tokenizer-worker-num > 1` 不兼容的校验, 是功能解锁的关键配置变更。
- test/registered/openai_server/basic/test_http2_server.py (模块 集成测试; 类别 test; 类型 test-coverage; 符号 `TestHTTP2ServerMultiTokenizer`, `setUpClass`) : 新增 `TestHTTP2ServerMultiTokenizer` 测试类, 覆盖 multi tokenizer worker + HTTP/2 场景, 是保证新路径质量的关键测试。
- python/sglang/srt/managers/multi_tokenizer_mixin.py (模块 Token 管理; 类别 source; 类型 dependency-wiring; 符号 `get_main_process_id`) : 简化 `get_main_process_id` 实现, 移除环境变量覆盖逻辑, 体现清理旧路径。
- python/sglang/srt/envron.py (模块 环境配置; 类别 source; 类型 core-logic; 符号 `SGLANG_GRANIAN_PARENT_PID`) : 移除了 `SGLANG_GRANIAN_PARENT_PID` 环境变量定义, 是清理工作的一部分。

关键符号: `_run_granian_server`, `serve`, `lifespan`, `init_multi_tokenizer`, `get_main_process_id`

关键源码片段

python/sglang/srt/entrypoints/http_server.py

核心变更文件, 重写 Granian 启动函数 (`_run_granian_server`), 添加两种运行模式; 移除旧的初始化逻辑 (`_init_granian_worker`, `_close_main_process_sockets`); 调整 `lifespan` 控制流。

```
# python/sglang/srt/entrypoints/http_server.py (head)
# lifespan 函数中移除旧分支, 统一走 init_multi_tokenizer
@asynccontextmanager
async def lifespan(fast_api_app: FastAPI):
    if getattr(fast_api_app, "is_single_tokenizer_mode", False):
        server_args = fast_api_app.server_args
        warmup_thread_kwargs = fast_api_app.warmup_thread_kwargs
        thread_label = "Tokenizer"
    else:
        # 不再检查 envs.SGLANG_GRANIAN_PARENT_PID, 所有 worker 进程
        # 都通过共享内存的 init_multi_tokenizer 初始化自身状态
        server_args = await init_multi_tokenizer()
        warmup_thread_kwargs = dict(server_args=server_args)
        thread_label = f"MultiTokenizer-{{global_state.tokenizer_manager.worker_id}}"

# _run_granian_server 新签名的简要示意 (具体实现因不完整未展开)
def _run_granian_server(
    host, port, log_level, tokenizer_worker_num=1,
```

```

ssl_certfile=None, ssl_keyfile=None, ssl_ca_certs=None,
ssl_keyfile_password=None, ssl_verify=False,
backlog=2048, backpressure=2048,
):
    """
    当 tokenizer_worker_num == 1 时使用 Granian 嵌入式模式（当前进程内 asyncio task）；
    当 tokenizer_worker_num > 1 时使用 Granian MPSTServer 启动多个 worker 进程。
    """
    # ...

```

python/sglang/srt/server_args.py

移除 `--enable-http2` 与 `--tokenizer-worker-num > 1` 不兼容的校验，是功能解锁的关键配置变更。

```

# python/sglang/srt/server_args.py (head)
# 在 _handle_ssl_validation 中，原代码抛出错误的部分已删除
if self.enable_http2:
    # ... granian 检查和 ssl_refresh 检查保持不变
    # 下方删除的 6 行语句：
    # if self.tokenizer_worker_num > 1:
    #     raise ValueError(
    #         "--enable-http2 does not yet support --tokenizer-worker-num > 1. "
    #         "Multi-worker HTTP/2 support will be added in a future release."
    #     )

```

评论区精华

- SSL 参数映射问题（评论者：merrymercy）：指出 `ssl_client_verify=ssl_verify` 用法错误。`ssl_verify` 是用于 `requests.verify` 的值（True/False/CA 路径），而 Granian 的 `ssl_client_verify` 是服务器端布尔值，用于客户端证书验证。直接传入可能导致类型错误或意外启用双向 TLS。建议默认设为 False，除非显式添加 mTLS 选项。
- 测试覆盖要求（评论者：ispobock）：询问是否测试了 multi tokenizer 路径。作者 rainj-me 回复已添加对应测试。
- SSL 参数传递错误 (correctness): 应改为 `False` 或添加独立选项。未在合并前修复。
- Multi tokenizer 测试覆盖 (testing): 新增 `TestHTTP2ServerMultiTokenizer` 测试类覆盖此路径。

风险与影响

- 风险：
 - SSL 配置风险：http_server.py 中 `ssl_client_verify=ssl_verify` 的参数传递可能引发类型错误或意外启用 mTLS，导致正常请求失败（该问题在合并前未被修复）。
 - 嵌入式服务器稳定性：嵌入式模式直接在主 asyncio 事件循环中运行 server，若有阻塞操作将影响整体请求处理，需要保证所有处理为异步协程。
 - 回归风险：重构 Granian 启动流程、删除旧函数和环境变量，可能影响依赖 SGLANG_GRANIAN_PARENT_PID 或 `_init_granian_worker` 的第三方集成。

- 影响：

- 用户影响：现在可以同时使用 HTTP/2 与多个 tokenizer worker，提升高并发场景下的吞吐能力。单 worker 用户无感知。
- 系统影响：嵌入式模式避免了额外的进程间通信开销，可能轻微改善延迟和 CPU 使用。
- 团队影响：代码量大致持平（+136/-133），但移除了旧路径，降低了维护负担。
- 风险标记：SSL 参数错误，嵌入式服务阻塞风险，旧环境变量移除兼容性

关联脉络

- 暂无明显关联 PR