PR #24089 完整报告

sgl-project/sglang

[Feat][LMCache] Support LMCache mp mode

合并时间：2026-05-28 10:15

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/24089

执行摘要

- 一句话：支持 LMCache 多进程模式，解耦缓存进程与推理进程
- 推荐动作：值得精读，尤其是两阶段加载设计、模式枚举抽象和基于 YAML 的配置方式。对于理解 SGLang 缓存层扩展机制和有状态的推理系统解耦有借鉴意义。

功能与动机

PR body 指出：'This PR adds the SGLang-side wiring needed to use LMCache in multi-process (MP) mode.' 之前仅支持进程内（IP）模式，缓存生命周期与推理进程绑定。MP 模式允许 LMCache 作为独立守护进程，支持跨 SGLang 重启的缓存持久化，便于弹性部署和故障恢复。

实现拆解

1. CLI 参数扩展：在 server_args.py 中添加 --lmcache-config-file 可选参数，用于指定 LMCache YAML 配置文件路径。
2. 模式枚举与数据结构：在 lmc_radix_cache.py 中新增 LMCacheMode 枚举（MP/IP）和 _LMCacheLoadBackMarker 数据类，承载两阶段加载时从 match_prefix 传递到 init_load_back 的元数据。
3. 构造函数重构：LMCRadixCache.__init__ 默认设置 self._mode = LMCacheMode.MP；若未提供 --lmcache-config-file 则抛出 ValueError；通过 lmcache_get_config 解析 YAML 中的 mp_host/mp_port 创建 LMCacheMPConnector；IP 模式下回退到 LMCacheLayerwiseConnector。
4. 两阶段加载（MP 模式）：match_prefix 分派到 _mp_match_prefix，仅执行 LOOKUP 查询是否命中并返回 host_hit_length；init_load_back 阶段执行实际 RETRIEVE，将 KV 从 LMCache 守护进程拉取到预分配 GPU 槽位。IP 模式仍使用原来的单阶段 _ip_match_prefix 直接发起 start_load_kv。
5. 配套变更：重命名 example_config.yaml 为 example_config_ip.yaml，新增 example_config_mp.yaml；更新 README.md 和官方文档说明两种模式用法；调整单元测试，移除环境变量设置改为显式传递 IP 配置文件。

关键文件：

- python/sglang/srt/mem_cache/storage/lmcache/lmc_radix_cache.py（模块 缓存层；类别 source；类型 dependency-wiring；符号 _LMCacheLoadBackMarker,

LMCacheMode, reset, match_prefix）：核心变更文件，引入 MP/IP 模式选择、两阶段加载协议和对称方法重构，改动最大（+278/-57）。

- python/sglang/srt/server_args.py（模块 启动配置；类别 source；类型 core-logic）：新增 lmcache_config_file 字段和 --lmcache-config-file 参数，提供 YAML 配置入口。
- python/sglang/srt/mem_cache/storage/lmcache/unit_test.py（模块 单元测试；类别 test；类型 test-coverage）：更新单元测试，移除环境变量配置改为显式传递配置文件，保证 IP 模式测试可复现。
- python/sglang/srt/mem_cache/storage/lmcache/example_config_mp.yaml（模块 配置文件；类别 config；类型 configuration）：新增 MP 模式 YAML 配置文件，定义 mp_host 和 mp_port。
- python/sglang/srt/mem_cache/storage/lmcache/README.md（模块 文档；类别 docs；类型 documentation）：详细说明 MP/IP 两种模式的部署和使用方法，更新示例命令。

关键符号：init, match_prefix, _mp_match_prefix, _ip_match_prefix, init_load_back, _load_back, reset

评论区精华

Review 由 Oasis-Git 主导，提出 5 条关键评论：

- 将 from sglang.srt.server_args import get_global_server_args 移到文件顶部（导入顺序规范）。
- 使用 LMCacheMode 枚举替代布尔量 _mp_mode，使模式分支对称清晰。
- match_prefix 中的内部函数应提取为 _mp_match_prefix 和 _ip_match_prefix 两个独立方法，便于维护。
- 将 --lmcache-mp-host/--lmcache-mp-port 参数移除，改为通过 YAML 配置文件统一管理服务器地址（关注点分离）。 所有评论均在最终代码中得到采纳。
- 代码组织与设计抽象 (design): 最终代码采纳所有建议：导入在顶部，使用 LMCacheMode 枚举，创建 _mp_match_prefix 和 _ip_match_prefix 两个方法，通过 YAML 配置 MP host/port。

风险与影响

- 风险：
  1. 默认行为变更：默认启用 MP 模式，但要求 --lmcache-config-file 参数，未提供则抛出 ValueError，现有 LMCache 用户升级后可能中断。
  2. IP 模式隐藏：IP 模式需在代码层面设置 self._mode = LMCacheMode.IP，对仅使用配置的运维人员不友好。
  3. 网络依赖：MP 模式依赖 ZMQ 通信，守护进程故障或网络延迟可能阻塞推理，现有 LayerTransferCounter 需要确保流同步正确。
  4. 测试覆盖不足：单元测试仅覆盖 IP 模式基本路径，MP 模式缺乏自动化测试（需依赖外部 LMCache 守护进程）。
  5. 回退逻辑缺失：init_load_back 中 RETRIEVE 失败时未定义明确的降级行为。

- 影响：
  - 用户影响：使用 LMCache 的用户需重新配置，决定采用 MP 或 IP 模式；MP 模式需额外启动 lmcache server 守护进程。
  - 系统影响：MP 模式解耦缓存进程与推理进程，推理服重启不丢失缓存，但运维复杂度增加。
  - 团队影响：维护两种连接器实现和配置路径，增加代码复杂度，但促进关注点分离和可测试性。
  - 风险标记：默认行为变更，核心路径变更，新增 YAML 配置依赖，缺少 MP 模式单元测试

关联脉络

- 暂无明显关联 PR