PR #22659 完整报告

sgl-project/sglang

Add sleep/wake support for diffusion engine

合并时间：2026-06-25 09:37

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/22659

执行摘要

- 一句话：为 diffusion 引擎添加 sleep/wake 支持，可释放 GPU 内存而不重启服务
- 推荐动作：值得精读其设计权衡，特别是 all-or-nothing 移动回滚机制和与 LayerwiseOffload 的交互。但若只是使用 diffusion 引擎，可快速阅读 API 文档即可。

功能与动机

This PR adds coarse-grained sleep/wake support for the multimodal diffusion engine so the server can temporarily release GPU memory without restarting the process. 主要面向 RL 等场景，当引擎暂时空闲时释放 GPU 资源供其他任务使用。

实现拆解

1. 新增 MemoryOccupationController（memory_occupation_controller.py）：
   - 核心类管理睡眠 / 唤醒状态，提供 _move_modules（all-or-nothing 移动与回滚）、_offload_active_modules_to_cpu（跳过 LayerwiseOffload 模块）、_restore_modules_from_map 等方法。
   - 辅助函数 _get_module_device、_move_unregistered_tensors 处理未注册张量。
2. GPUWorker 集成（gpu_worker.py）：
   - 添加 memory_occupation 属性，延迟初始化 MemoryOccupationController。
   - 实现 is_sleeping、_get_memory_occupation、release_memory_occupation、resume_memory_occupation 方法。
3. 调度器路由（scheduler.py）：
   - 在 request_handlers 中注册 ReleaseMemoryOccupationReqInput 和 ResumeMemoryOccupationReqInput 的处理函数。
   - 修改 _handle_generation 在睡眠时返回 OutputBatch(error='Server is sleeping...')，而不是抛出异常。
   - 同时修改 _handle_update_weights_from_disk 在睡眠时拒绝更新权重。
4. API 端点（weights_api.py）：
   - 新增两个 POST 端点 /release_memory_occupation 和 /resume_memory_occupation，调用调度器并处理响应。
   - 每个端点独立实现，以便后续独立迭代。

5. 请求结构体（io_struct.py）：
   - 新增 ReleaseMemoryOccupationReqInput 和 ResumeMemoryOccupationReqInput 两个 dataclass，作为请求载体。

关键文件：

- python/sglang/multimodal_gen/runtime/managers/memory_managers/memory_occupation_controller.py（模块 扩散引擎；类别 source；类型 core-logic；符号 _get_module_device, _move_unregistered_tensors, move_tensors, _is_layerwise_offload_managed）：核心新文件，实现 MemoryOccupationController 类，包含模块移动、回滚、跳过 LayerwiseOffload 等关键逻辑
- python/sglang/multimodal_gen/runtime/entrypoints/post_training/weights_api.py（模块 扩散引擎；类别 source；类型 entrypoint；符号 release_memory_occupation, resume_memory_occupation）：新增两个 POST 端点，作为用户调用的入口
- python/sglang/multimodal_gen/runtime/managers/gpu_worker.py（模块 扩散引擎；类别 source；类型 core-logic；符号 is_sleeping, _get_memory_occupation, release_memory_occupation, resume_memory_occupation）：GPUWorker 中添加 sleep/wake 方法，是实际执行模块移动的地方
- python/sglang/multimodal_gen/runtime/managers/scheduler.py（模块 扩散引擎；类别 source；类型 core-logic；符号 _handle_update_weights_from_disk, _handle_release_memory_occupation, _handle_resume_memory_occupation）：调度器注册新的请求类型，添加处理函数，并在生成请求时检查睡眠状态
- python/sglang/multimodal_gen/runtime/entrypoints/post_training/io_struct.py（模块 扩散引擎；类别 source；类型 core-logic；符号 ReleaseMemoryOccupationReqInput, ResumeMemoryOccupationReqInput）：新增两个 dataclass 用作请求结构

关键符号：MemoryOccupationController.init, MemoryOccupationController._move_modules, MemoryOccupationController._offload_active_modules_to_cpu, MemoryOccupationController.release_memory_occupation, MemoryOccupationController.resume_memory_occupation, GPUWorker.is_sleeping, GPUWorker._get_memory_occupation, GPUWorker.release_memory_occupation, GPUWorker.resume_memory_occupation, Scheduler._handle_release_memory_occupation, Scheduler._handle_resume_memory_occupation, Scheduler._handle_generation

评论区精华

Review 中主要讨论了以下设计权衡：

主题	类别	讨论	结论
error_status_code 字段必要性	design	Rockdu 认为应复用现有错误路径而非新增字段	已移除该字段，恢复统一错误处理
共享 handler 是否合理	design	Rockdu 建议拆分 release/resume 端点以便独立迭代	已拆分为两个独立 API handler

主题	类别	讨论	结论
跳过 Layerwise Offload 模块	correctness	Rockdu 指出需要跳过已由 layerwise offload 管理的模块	已添加 _is_layerwise_offload_managed 检查
日志中对 HTTPException 的特殊处理	design	Rockdu 质疑是否必要	已移除，恢复原始日志行为
调度器睡眠时返回 OutputBatch vs. raise	correctness	Rockdu 认为应直接 raise，作者认为 OutputBatch 更符合请求失败的正常路径	保留 OutputBatch 方式，保持请求处理一致性
MemoryOccupationController 的初始化时机	performance	mickqian 询问非 RL 场景是否需要早初始化	改为延迟初始化，避免 startup 开销

- error_status_code 字段必要性 (design): 移除 error_status_code 字段，恢复统一错误处理
- 共享 _handle_memory_occupation_request 是否合理 (design): 已拆分为两个独立 API handler
- 跳过 LayerwiseOffload 模块 (correctness): 添加 _is_layerwise_offload_managed 检查并跳过
- 日志中 HTTPException 特殊处理 (design): 已移除，恢复原始日志行为
- 调度器睡眠时返回 OutputBatch vs raise (correctness): 保留 OutputBatch 方式，保持请求处理一致性
- MemoryOccupationController 初始化时机 (performance): 改为延迟初始化，避免 startup 开销

风险与影响

- 风险：
  1. 状态不一致风险：如果 release 或 resume 操作在分布式环境中失败（如某个 worker 掉线），可能导致部分模块在 GPU 部分在 CPU，但代码提供了 all-or-nothing 回滚，降低了风险。
  2. 测试覆盖缺失：PR 没有包含直接针对 sleep/wake 流程的单元测试，仅依赖 CI 集成测试。可能出现边缘情况未被覆盖（如并发请求处理时的状态竞争）。
  3. API 端点暴露：/release_memory_occupation 和 /resume_memory_occupation 是公开端点，如果未加鉴权，外部攻击者可能恶意触发睡眠，导致服务不可用。
  4. LayerwiseOffload 交互：睡眠 / 唤醒跳过 LayerwiseOffload 管理的模块，但如果 LayerwiseOffload 状态在睡眠期间变化（如权重更新），可能导致不一致。 - 影响：用户影响：用户可通过调用 API 临时释放 GPU 内存，适用于 RL 或分时复用场景，无需重启服务。若在睡眠时发送生成请求，会收到 400 错误和明确提示。系统影响：新增

FastAPI 路由和调度器请求类型，增加代码库维护成本。内存占用在睡眠期间显著降低（模块移至 CPU），但恢复时需重新加载，增加延迟。团队影响：该功能由单一作者实现，但经历 82 次提交和多人 reviewer，代码质量经过多次重构。需要后续文档和维护。

- 风险标记：缺少测试覆盖，API 端点无鉴权可能被滥用，睡眠期间权重更新可能引发不一致，分布式环境下部分模块移动失败回滚后需清理

关联脉络

- PR #19152 Add sleep/wake support for diffusion engine (original): 此 PR 继承自 #19152，基于其 rebase 并继续开发，几乎所有变更都源自该 PR