

PR #2248 完整报告

THUDM/slime

Support PYTORCH_ALLOC_CONF

合并时间: 2026-08-03 19:38

原文链接: <http://prhub.com.cn/THUDM/slime/pull/2248>

执行摘要

- 一句话: PYTORCH_ALLOC_CONF 支持, 隔离 SGLang
- 推荐动作: 改动小但有针对性, 结合 colocation 显存优化线 (#2180 等) 值得快速阅读。
关注点: 环境变量隔离位置是否合理, 以及 Dockerfile 依赖锁定策略——建议后续改为固定 commit 或构建产物校验, 避免构建漂移。

功能与动机

PR 标题与 body 均未提供详细说明, 动机主要来自代码注释: expandable segments 能帮助 colocated 训练 actor 容忍反复的 cache 释放, 但 SGLang 的 allocator/sleep 路径不支持该选项; rollout 侧 Ray actor 会继承 job 环境变量, 因此需要在 spawn SGLang server 前移除。配套将 torch_memory_saver 切换到作者 fork, 推测是为了让镜像内包含与 PYTORCH_ALLOC_CONF 兼容的 allocator 版本。

实现拆解

1. 环境变量隔离 (入口清理): 在 slime/backends/sglang_utils/sglang_engine.py 的 launch_server_process 函数开头新增两行 os.environ.pop(...), 清理 PYTORCH_CUDA_ALLOC_CONF 与 PYTORCH_ALLOC_CONF。该函数是所有 SGLang server (含 encoder-only、disaggregated) 的统一启动入口, 在这里清理能覆盖所有子进程; 对于未设置这些变量的场景是 no-op, 不影响原有行为。
2. 镜像依赖调整: docker/Dockerfile 中 torch_memory_saver 的安装地址从 fzyzcjy 仓库固定 commit (a193d9dd...) 改为 zhuzilin fork 且不锁版本, 保证镜像内含与新配置兼容的 allocator 实现, 但引入了可复现性风险。
3. 版本号同步: docker/version.txt 从 nightly-dev-20260722a 更新为 nightly-dev-20260803a, 与新构建的镜像对应。
4. 测试与配置配套: 本次没有新增测试文件, 也没有新增命令行参数或文档说明。用户只需在提交 Ray job 时设置 PYTORCH_ALLOC_CONF (如 expandable_segments:True) 即可生效, SGLang 侧不受影响。

关键文件:

- slime/backends/sglang_utils/sglang_engine.py (模块引擎启动; 类别 source; 类型 core-logic; 符号 launch_server_process): 核心改动文件, 在统一的服务启动入口清理 allocator 配置环境变量, 决定整个 rollout 侧 SGLang 进程的显存分配行为。

- docker/Dockerfile (模块 构建镜像; 类别 infra; 类型 infrastructure) : 切换 torch_memory_saver 安装源到作者 fork, 未锁定 commit, 影响镜像可复现性与 allocator 行为。
- docker/version.txt (模块 版本信息; 类别 docs; 类型 documentation) : 同步更新 nightly 镜像版本标识, 与 Dockerfile 变更配套。

关键符号: launch_server_process

关键源码片段

slime/backends/sglang_utils/sglang_engine.py

核心改动文件, 在统一的服务启动入口清理 allocator 配置环境变量, 决定整个 rollout 侧 SGLang 进程的显存分配行为。

```
def launch_server_process(server_args: ServerArgs) -> multiprocessing.Process:
    # expandable segments 能帮助 colocated 训练 actor 容忍反复的 cache 释放,
    # 但 SGLang 的 allocator/sleep 路径并不支持该配置。
    # rollout 侧 Ray actor 会继承 job 环境变量, 因此在 spawn 每个 SGLang
    # server 及其子进程之前移除这些选项, 避免不兼容配置泄漏到服务进程。
    os.environ.pop("PYTORCH_CUDA_ALLOC_CONF", None)
    os.environ.pop("PYTORCH_ALLOC_CONF", None)

    # encoder-only 场景直接复用 SGLang 自带的服务启动逻辑
    if getattr(server_args, "encoder_only", False):
        from sglang.srt.disaggregation.encode_server import launch_server_process as sglang_
        launch_server_process

    return sglang_launch_server_process(
        server_args,
        start_method="spawn",
        wait_for_server=True,
    )

from sglang.srt.entrypoints.http_server import launch_server

# 使用 spawn 方式启动独立进程, 确保清理后的环境变量在子进程生效
multiprocessing.set_start_method("spawn", force=True)
server_args.host = server_args.host.strip("[]")
p = multiprocessing.Process(target=launch_server, args=(server_args,))
p.start()

# 非 rank 0 节点无需等待健康检查
if getattr(server_args, "node_rank", 0) != 0:
    return p

_wait_server_healthy(
    base_url=server_args.url(),
    api_key=server_args.api_key,
```

```
is_process_alive=lambda: p.is_alive(),
)

return p
```

评论区精华

该 PR 没有任何 review 评论或 issue 讨论，设计权衡主要内嵌在代码注释中。核心决策是：为了让 colocated 训练 actor 从 expandable segments 中获益，同时避免把不兼容选项传递给 SGLang server，选择在 spawn 前 pop 环境变量，而不是在 SGLang 侧做兼容。这样把变化收敛在 slime 的启动层，不需要改 SGLang patch，但代价是若未来 SGLang 支持该配置，这一清理逻辑需要重新评估。

- 暂无高价值评论线程

风险与影响

- 风险：
 - 环境变量被无条件移除：如果未来有用户希望给 SGLang server 本身配置 `PYTORCH_ALLOC_CONF`，当前实现无法透传。短期内 SGLang 不支持，风险可控。
 - Docker 依赖未锁定 commit: zhuzilin/torch_memory_saver 未指定版本，镜像重建可能随时间漂移，存在供应链与可复现性风险。
 - 缺少测试覆盖：改动落在 `launch_server_process` 这一 rollout 核心路径上，但没有任何单测或集成测试验证 "清理生效" 和 "配置可用于 actor"，重构时容易回归。
 - 对普通用户透明：未设置相关变量时，两行 pop 是 no-op，不影响现有行为。
 - 影响：对使用 colocated 训练 + SGLang 推理的用户，允许设置 `PYTORCH_ALLOC_CONF` 提升显存稳定性；对普通用户是透明 no-op。对团队，需要维护对 zhuzilin fork 的依赖和版本跟进；该改动与 #2180 的 release-train 配合，能减少反复 cache 释放带来的显存碎片问题。影响范围集中在 rollout 服务启动路径和 Docker 构建，影响程度中等偏低。
- 风险标记：缺少测试覆盖，依赖未锁定版本，核心启动路径变更

关联脉络

- PR #2180 Add --release-train: 本 PR 注释提及 colocated 训练 actor 反复 cache 释放，正是 --release-train 引入的释放 / 重建场景；该 PR 为训练侧显存释放机制，本 PR 为其提供 expandable_segments 支持并隔离 SGLang 进程。
- PR #2228 [docker] upgrade sglang to v0.5.15.post1: 同涉 docker/version.txt 与 sglang_engine.py；SGLang 升级后其 allocator/sleep 路径行为变化，本 PR 的环境变量清理与之配套。
- PR #2208 Support reloading the default process group: 同涉 rollout 进程生命周期与环境管理，reloadable process group 与 spawn 环境清理共同影响多进程显存环境。