

PR #29844 完整报告

sgl-project/sglang

[CI] Revert ModelOpt NVFP4 threshold relax

合并时间: 2026-07-04 21:17

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/29844>

执行摘要

- 一句话: 恢复 B200 NVFP4 严格阈值并加固 CUDA 13 CI 依赖管理
- 推荐动作: 此 PR 的核心价值在于 暴露真实环境问题而非放松阈值, 是 CI 可靠性加固的典型实践。值得关注 `remove_stale_cuda12_nvidia_wheels` 的批量查询实现方式和 `SGLANG_CI_EARLY_LD_LIBRARY_PATH` 的环境变量控制模式, 可作为类似 CI 依赖治理的参考。建议精读 `ci_install_dependency.sh` 中的防护逻辑。

功能与动机

29767 中放宽的 NVFP4 阈值掩盖了 B200 runner 的环境问题。实际的 bad runs 发生在 `b200-cirrascale2` 上, 该节点 `CUDA_VISIBLE_DEVICES` 为空, 且遗留的 CUDA 12 wheel 泄漏到 CUDA 13 作业中。恢复严格阈值后, CI 能正确捕获此类环境异常, 而非通过放宽容差“通过”本应失败的任务。

实现拆解

1. 还原严格阈值: 从 `python/sglang/multimodal_gen/test/server/consistency_thresholds/h100.json` 中删除 `flux2_modelopt_nvfp4_t2i` 和 `wan22_modelopt_nvfp4_t2v` 两个条目, 使 B200 恢复继承严格默认值, H100 不受影响。
2. 加固 CUDA 13 依赖安装: 在 `scripts/ci/cuda/ci_install_dependency.sh` 中新增 `remove_stale_cuda12_nvidia_wheels()` 函数, 在 CUDA 13 作业中自动卸载所有 `nvidia-*-cu12 wheel`; 函数内部通过 `pip list --format=freeze` 批量查询, 避免了逐包 `spawn` 进程的性能开销。
3. 前置 `LD_LIBRARY_PATH` 刷新: 通过环境变量 `SGLANG_CI_EARLY_LD_LIBRARY_PATH` 控制, 在 `install_sglang_kernel` 之前提前调用 `setup_ld_library_path`, 确保扩散测试的 B200 CI 能正确加载 CUDA 13 的库, 避免因 `cuDNN` 缺失导致的 `import` 失败。
4. 补全 CPU 依赖: 在 Arm64 CPU CI 镜像中安装 `pytest`, 并在 CPU 包依赖中添加 `zstandard`, 解决请求解压测试和运行时 `import parity` 所需的依赖缺失。

关键文件:

- python/sglang/multimodal_gen/test/server/consistency_thresholds/h100.json (模块 测试配置; 类别 test; 类型 test-coverage) : 删除了两个 NVFP4 测试用例 (flux2_modelopt_nvfp4_t2i、wan22_modelopt_nvfp4_t2v) 的阈值配置, 恢复严格默认值, 是回退操作的核心数据文件。
- scripts/ci/cuda/ci_install_dependency.sh (模块 CI 脚本; 类别 infra; 类型 infrastructure) : 新增 remove_stale_cuda12_nvidia_wheels 函数和前置 LD_LIBRARY_PATH 刷新逻辑, 是 CI 依赖加固的核心实现。
- .github/workflows/pr-test-multimodal-gen.yml (模块 工作流; 类别 infra; 类型 infrastructure) : 为 diffusion B200 作业设置环境变量 SGLANG_CI_EARLY_LD_LIBRARY_PATH=1, 触发脚本中的前置库路径刷新逻辑。

关键符号: remove_stale_cuda12_nvidia_wheels

关键源码片段

scripts/ci/cuda/ci_install_dependency.sh

新增 `remove_stale_cuda12_nvidia_wheels` 函数和前置 `LD_LIBRARY_PATH` 刷新逻辑, 是 CI 依赖加固的核心实现。

```
# 批量卸载遗留的 CUDA 12 wheel, 避免污染 CUDA 13 环境
remove_stale_cuda12_nvidia_wheels(){ # 仅在 CUDA 13 下执行
if["$CU_MAJOR"!="13"];then mark_step_done"${FUNCNAME[0]}" return fi # 用 pip list
--format=freeze 一次性拿到所有 wheel 名, 再用 sed 筛选
mapfile -t STALE_CUDA12_NVIDIA_WHEELS << ( python3 -m pip list --format=freeze | sed -n 's/
^\(nvidia-.*-cu12\)==.*$/1/p' ) if[${#STALE_CUDA12_NVIDIA_WHEELS[@]}-eq 0];then
echo "No stale CUDA 12 NVIDIA wheels found for ${CU_VERSION} job"
mark_step_done"${FUNCNAME[0]}" return fi echo "Removing stale CUDA 12 NVIDIA
wheels from ${CU_VERSION} job: ${STALE_CUDA12_NVIDIA_WHEELS[*]}"
$PIP_UNINSTALL_CMD "${STALE_CUDA12_NVIDIA_WHEELS[@]}" $PIP_UNINSTALL_SUFFIX
mark_step_done"${FUNCNAME[0]}" } # 主流程调用顺序中的关键插入点
main(){ ...
remove_stale_cuda12_nvidia_wheels # 在安装 flashinfer 之前清理
uninstall_stale_flashinfer install_sglang # 针对扩散 B200 作业: 提前刷新
LD_LIBRARY_PATH # 因为 install_sglang_kernel 会 import torch, 需要 cuDNN 可用
if["$SGLANG_CI_EARLY_LD_LIBRARY_PATH:-0"]="1";then setup_ld_library_path fi
install_sglang_kernel ... }
```

评论区精华

gemini-code-assist[bot] 在 review 中指出 `remove_stale_cuda12_nvidia_wheels` 实现存在性能优化空间: 当前通过循环逐包调用 `pip show` 会 spawn 大量子进程, 建议改为一次 `pip show` 查询所有包以减少开销。同时建议补充清理 `nvidia-cuda-cccl-cu12` 和 `nvidia-nvjpeg-cu12` 以提升彻底性。最终实现中已采用 `pip list --format=freeze` 一次性匹配方案, 避免了循环 spawn, 但未明确回应是否补充了上述两个包。讨论无未解决疑虑。

- 性能优化建议：避免循环 pip show (performance): 作者采用了 pip list --format=freeze | sed -n 方案，一次性获取所有包名，避免了循环 spawn，但未回复是否还采纳了补充 package 列表的建议。

风险与影响

- 风险：回归风险：删除的 NVFP4 阈值条目仅作用于 h100.json，B200 使用独立的 b200.json 或默认值，不会影响 B200 原本的严格阈值。但若未来 b200.json 继承逻辑有变更，可能意外继承到松弛值。CI 流程风险：新增的 remove_stale_cuda12_nvidia_wheels 在非 CUDA 13 环境下直接返回，不影响现有流程；但 SGLANG_CI_EARLY_LD_LIBRARY_PATH 环境变量若在非 diffusion 作业中误设，可能导致不必要的 LD_LIBRARY_PATH 重算，影响较小。兼容性：所有变更局限在 CI 脚本和测试阈值配置文件，不影响运行时功能。
- 影响：影响范围：仅限 CI 流程。影响 Diffusion 测试的 B200 作业、CUDA 13 依赖安装、Arm64 CPU CI 以及 CPU 包构建。对用户运行时无影响。影响程度：中等。修复了 CI 中环境泄漏导致的误报和漏报，提升了 B200 diffusion 测试的可靠性；同时补全了 CPU 依赖，确保相关测试和功能可正常执行。
- 风险标记：CI 流程变更，扩散测试阈值调整

关联脉络

- PR #29767 relax B200 ModelOpt NVFP4 consistency thresholds in diffusion CI: 本 PR 正是回退该 PR 的阈值放宽操作，直接相关。