

PR #33637 完整报告

sgl-project/sglang

[CI] Skip sglang-kernel and sgl-deep-gemm reinstall on version match

合并时间: 2026-08-05 16:46

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/33637>

执行摘要

- 一句话: CI 按版本门控跳过 sglang-kernel 与 sgl-deep-gemm 重装
- 推荐动作: 值得精读, 特别是 `installed_wheel_ok` 把版本、来源、文件完整性三重校验打包为单个 shell 函数的做法, 对 CI 中“跳过重装但保持可信”这类问题有通用参考价值。关注点: 版本串拼接依赖 PEP 440 local version 约定, RECORD 校验只能覆盖文件缺失, 内容损坏仍需依赖重建缓存策略。

功能与动机

PR body 指出: 安装步骤每次都在强制重装这两个 wheel, 即使已装的就是完全相同的 +cuXXX tagged 版本, 作者估算 “Saves ~4-5s per install on warm runners”。更重要的是, 作者明确警告裸版本比较不可信, 因为:

1) PEP 440 local version 会满足 pyproject 中裸 ==0.4.5 pin, 导致 `install_sglang` 放行不匹配 CUDA 的 wheel; 2) `CUSTOM_BUILD_SGL_KERNEL=true` 的 job 会在相同 +cuXXX 版本串下安装本地构建 wheel, 若无来源校验, 下一个 job 会静默测试到前一个 PR 修改过的 kernel; 3) `dist-info` 可能在部分安装后残留, 与既有 `nvidia-cusparselt guard` 防御的失败模式相同。这三个原因促使引入三重校验门禁。

实现拆解

1. `setup_pip_toolchain`: 把 `sglang-kernel` 从 `blanket uninstall` 列表中移除, 改为交给 `install_sglang_kernel` 的版本门禁管理, 避免 warm runner 每次重新物化 wheel; 仍保留对 `sgl-kernel`、`sglang`、`sgl-fa4`、`flash-attn-4` 的清理。
2. 新增 `installed_wheel_ok` 门禁函数: 用 `importlib.metadata` 检查发行版是否安装、`dist.version` 是否精确匹配期望串 (含 +cuXXX / +cu129 完整标签)、可选的 `reject-local` 是否命中 `direct_url.json` 中的 `file://` 来源、RECORD 中每个文件是否真实存在; 各失败路径都会打印原因并以非 0 退出。
3. `install_sglang_kernel` 分支改造: 非自定义构建时期望版本为 `${SGL_KERNEL_VERSION_FROM_SRT}+${CU_VERSION}`, 门禁通过则跳过 `--force-reinstall`, 否则从 `https://docs.sglang.ai/whl/${CU_VERSION}/` 强制重装; `CUSTOM_BUILD_SGL_KERNEL=true` 分支保持不变。
4. `sgl-deep-gemm` 分支改造: `CU_MAJOR=13` 时期望版本不带 tag, 否则为 `${SGL_DEEP_GEMM_VERSION}+cu129`; 只做版本 + payload 校验, 不做来源校验 (无

人本地构建，且官方 cu129 wheel 从直链安装会合法记录非 file:// 来源）。

5. 配套说明：无源码、测试或配置改动，脚本正确性由 PR CI 自身验证；4 个 commit 依次为版本门控、来源与 payload 校验、精简注释、补充失败日志。

关键文件：

- scripts/ci/cuda/ci_install_dependency.sh（模块 CI 脚本；类别 infra；类型 infrastructure；符号 installed_wheel_ok, install_sglang_kernel, setup_pip_toolchain）：唯一变更文件：移除 blanket uninstall 中的 sglang-kernel、新增 installed_wheel_ok 三重校验门禁并接入两个 wheel 的安装分支，是本次 CI 提速的核心。

关键符号：installed_wheel_ok, install_sglang_kernel, setup_pip_toolchain

关键源码片段

scripts/ci/cuda/ci_install_dependency.sh

唯一变更文件：移除 blanket uninstall 中的 sglang-kernel、新增 installed_wheel_ok 三重校验门禁并接入两个 wheel 的安装分支，是本次 CI 提速的核心。

```
# setup_pip_toolchain: 不再 blanket uninstall sglang-kernel,
# 改由 install_sglang_kernel 的版本门禁决定是否重装，避免 warm runner 重新物化 wheel。
$PIP_UNINSTALL_CMD sgl-kernel sglang sgl-fa4 flash-attn-4 $PIP_UNINSTALL_SUFFIX || true
```

```
# 三重校验门禁：版本 + CUDA 标签、来源（PEP 610）、文件完整性（RECORD）。
```

```
# 返回 0 表示可直接复用已装 wheel，非 0 表示需要重装，且打印失败原因。
```

```
installed_wheel_ok() {
    WHEEL_DIST="$1" WHEEL_WANTED="$2" WHEEL_REJECT_LOCAL="${3:-}" python3 -
    <<'EOF'
```

```
import importlib.metadata as md
```

```
import os
```

```
import sys
```

```
name = os.environ['WHEEL_DIST']
```

```
wanted = os.environ['WHEEL_WANTED']
```

```
try:
```

```
    dist = md.distribution(name)
```

```
except md.PackageNotFoundError:
```

```
    print(f'{name} 未安装（期望 {wanted}），触发重装')
```

```
    sys.exit(1)
```

```
# PEP 440: local version 满足裸 ==pin，所以必须精确比较带 +cuXXX 的完整版本串。
```

```
if dist.version != wanted:
```

```
    print(f'{name} 版本不匹配（已装 {dist.version}，期望 {wanted}），触发重装')
```

```
    sys.exit(1)
```

```
# PEP 610: 本地路径安装会在 direct_url.json 留下 file:// 来源，index 安装无记录；
```

```
# 用它拦截 CUSTOM_BUILD_SGL_KERNEL job 残留的本地构建 wheel，避免污染后续 job。
```

```
if os.environ['WHEEL_REJECT_LOCAL'] and 'file://' in (dist.read_text('direct_url.json') or ''):
```

```
    print(f'{name} 来自本地构建 wheel，改为从 index 重装')
```

```
    sys.exit(1)
```

```
if dist.files is None:
```

```

    print(f'{name} 无 RECORD 可校验, 触发重装')
    sys.exit(1)
# dist-info 可能在部分安装后残留 (与 cusparselt guard 防御的失败模式相同),
# 逐条核对 RECORD 文件是否真实存在; 文件缺失即触发重装。
missing = [str(f) for f in dist.files if not dist.locate_file(f).exists()]
if missing:
    print(f'{name} 缺失 {len(missing)} 个文件 (如 {missing[0]}), 触发重装')
    sys.exit(1)
EOF
}

# install_sglang_kernel 的接入点: 命中门禁则跳过重装, 否则强制重装匹配 CUDA 的 wheel。
if [ "${CUSTOM_BUILD_SGL_KERNEL:-}" != "true" ]; then
    SGL_KERNEL_WANTED="${SGL_KERNEL_VERSION_FROM_SRT}+${CU_VERSION}"
    if installed_wheel_ok sglang-kernel "${SGL_KERNEL_WANTED}" reject-local; then
        echo "sglang-kernel==${SGL_KERNEL_WANTED} 已安装, 跳过重装"
    else
        $PIP_CMD install "sglang-kernel==${SGL_KERNEL_VERSION_FROM_SRT}" \
            --index-url "https://docs.sglang.ai/whl/${CU_VERSION}/" \
            --force-reinstall --no-deps $PIP_INSTALL_SUFFIX
    fi
else
    echo "CUSTOM_BUILD_SGL_KERNEL=true: 保留刚构建的 sgl-kernel wheel."
fi

```

评论区精华

该 PR 没有实质 review 交锋: Fridge003 直接 APPROVED, 唯一 thread 是作者触发的 /tag-and-rerun-ci。设计要点全部写在 PR body 中, 最关键的两处表述是: “A bare version compare is not enough to trust the installed wheel”以及“the next job on that runner would silently test against the previous PR's modified kernel (the old unconditional reinstall scrubbed it every job)”, 后者解释了为什么必须做 PEP 610 来源校验。

- 三重校验替代裸版本比较 (design): 作者在第二个 commit 补齐 provenance 与 payload 校验后, reviewer 无异议直接批准; 未解决疑虑仅剩理论上内容损坏但文件齐全的极端情况。

风险与影响

- 风险: 1) PEP 440 语义依赖: 门禁依赖 index wheel 稳定携带 +cuXXX local version, 若未来 index 打包策略或 pip/uv 行为变化, 可能误判 (例如把 cu130 wheel 放行给 cu129 runner)。2) RECORD 只校验文件存在, 不校验内容: 若文件被截断但仍在磁盘上, 门禁会放行, 旧流程靠每次重装自愈的能力在此场景下丢失。3) sgl-deep-gemm 未做来源校验: 当前无人本地构建它, 但未来若引入本地构建流程, 需要补 reject-local。4) 出错模式是静默测试到错误 kernel, 因此门禁逻辑偏向保守 (任何一项失败都重装)。
- 影响: 只影响 CI 依赖安装阶段, 不影响运行时与用户请求。warm runner 每次安装节省约 4-5s, 并避免 kernel-PR 本地构建残留污染后续 job 的静默测试; 冷 runner 行为不变。团队侧收益为 CI 累积提速与更易排障 (门禁失败会打印明确原因)。

- 风险标记: CI 基础设施变更, 依赖 PEP 440 版本语义, RECORD 校验仅覆盖文件缺失, 无自动化测试

关联脉络

- PR #33619 [CI] Speed up dependency install: dual-ABI Rust ext cache and prevalidation pruning: 同属 CI 依赖安装提速系列, 且同样修改 scripts/ci/cuda/ci_install_dependency.sh, 与本 PR 形成连续的 CI 优化脉络。
- PR #33597 [CI] Extract download-rust-ext and give every install step a cache fallback: 给每个 install step 增加缓存回退, 与本 PR 的跳过重装目标一致, 共同减少 warm runner 上的重复安装开销。
- PR #33644 [CI] Free hosted-runner disk space only when it is low: 同为 CI 资源优化类变更, 反映仓库近期对 runner 效率与稳定性的持续投入。