

PR #34914 完整报告

sgl-project/sglang

Update sgl-deep-ep release workflow for DeepEP v2

合并时间: 2026-08-15 17:08

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/34914>

执行摘要

- 一句话: 适配 DeepEP v2, 更新 CUDA 13 构建依赖
- 推荐动作: 值得发布和镜像维护者精读, 理解 DeepEP v2 构建依赖的演化。关注点: 条件化 GDRCopy 安装的设计、NCCL 版本锁定方式、以及与 #34923 的边界划分。

功能与动机

DeepEP 的发布分支已改变: CUDA 13 使用 rebased DeepEP v2 分支, 需要 NCCL Gin from NCCL 2.30.7; 旧的 workflow 分支映射和 CUDA 13 GDRCopy builder 依赖已过时, 必须更新构建流程以匹配新依赖。

实现拆解

1. 更新 [.github/workflows/release-whl-deepep.yml](#) 的 release 矩阵:
 - CUDA 13 x86_64/aarch64 使用 sgl-deepep (原 sgl-deepep-x86 与 sgl-deepep-arm 合并);
 - CUDA 12.9 x86_64 使用 sgl-deepep-cu12-x86;
 - CUDA 12.9 aarch64 保留 sgl-deepep-cu12-arm。这样分支映射与 DeepEP v2 的新分支组织对齐。
2. 重写 [docker/sgl-deep-ep.Dockerfile](#) 的依赖安装逻辑:
 - 新增 ARG NCCL_VERSION=2.30.7;
 - 将 GDRCopy 的安装改为条件判断, 仅当 CUDA_TAG=cu129 时执行, 因为 CUDA 13 的 DeepEP v2 改用 NCCL Gin, 不再需要 GDRCopy;
 - 在 CUDA 13 (cu130) 下用 `pip install --force-reinstall --no-deps nvidia-nccl-cu13==2.30.7` 强制锁定 NCCL 版本;
 - 保留 NVSHMEM, 因为 DeepEP v2 extension 仍编译链接 legacy 方法。
3. 验证: 在 aarch64 GB300 devbox 上构建并安装 CUDA 13 / CPython 3.12 wheel, `import deep_ep` 成功且不依赖 `/dev/gdrdrv`; workflow 解析、actionlint、shell 语法检查均通过。

关键文件:

- [docker/sgl-deep-ep.Dockerfile](#) (模块 构建镜像; 类别 infra; 类型 infrastructure): 核心变更: CUDA 13 构建引入 NCCL Gin 依赖并跳过 GDRCopy, 条件化 legacy 路径。

- .github/workflows/release-whl-deepep.yml (模块 发布流程; 类别 infra; 类型 infrastructure) : 发布矩阵分支映射更新, 与 DeepEP v2 新分支对齐。

关键符号: 未识别

关键源码片段

`docker/sgl-deep-ep.Dockerfile`

核心变更: CUDA 13 构建引入 NCCL Gin 依赖并跳过 GDRCopy, 条件化 legacy 路径。

```
# DeepEP v2 在 CUDA 13 上使用 NCCL Gin, 不再需要 GDRCopy;
# 仅在 CUDA 12.9 的 legacy NVSHMEM/IBGDA 构建中保留 GDRCopy 用户态库。
RUN set -eux; \
    if [ "${CUDA_TAG}" = cu129 ]; then \
        git clone --depth 1 --branch "v${GDRCOPY_VERSION}" \
            https://github.com/NVIDIA/gdrCOPY.git /opt/gdrCOPY; \
        make -C /opt/gdrCOPY CUDA="${CUDA_HOME}" prefix=/usr/local lib_install; \
        printf '%s\n' /usr/local/lib > /etc/ld.so.conf.d/gdrCOPY.conf; \
        ldconfig; \
        test -f /usr/local/include/gdrapi.h; \
        ldconfig -p | grep -q libgdrapi; \
    fi

# CUDA 13 构建强制锁定 NCCL 2.30.7: DeepEP v2 依赖 NCCL Gin,
# 使用 --no-deps 避免破坏 torch 等已解析的依赖。
RUN --mount=type=cache,id=sgl-deep-ep-pip-${CUDA_TAG}-${PYTHON_TAG}-${ARCHITECTURE},
    target=/root/.cache/pip \
    set -eux; \
    "${PYTHON_BIN}" -m pip install --force-reinstall \
        "torch==${TORCH_VERSION}" \
        --index-url "https://download.pytorch.org/whl/${CUDA_TAG}"; \
    if [ "${CUDA_TAG}" = cu130 ]; then \
        "${PYTHON_BIN}" -m pip install --force-reinstall --no-deps \
            "nvidia-nccl-cu13==${NCCL_VERSION}"; \
    fi; \
    "${PYTHON_BIN}" -m pip install \
        "auditwheel>=6.0" \
    build
```

评论区精华

无 review 评论。关联 Issue 评论中 [Fridge003](#) 提出“Please delay the modification in docker/Dockerfile to #34923”, 即主镜像的运行时 NCCL 配置推迟到 #34923, 本 PR 只修改 DeepEP 专用镜像, 避免扩大影响面。

- docker/Dockerfile 修改是否应推迟 (design): 本 PR 仅修改 DeepEP 专用镜像, 主镜像的 NCCL 运行时配置由 #34923 承接。

风险与影响

- 风险:

- CUDA 13 构建移除 GDRCopy 依赖，若 DeepEP v2 后续回归需要 GDRCopy，构建产物将缺失用户态库；当前验证显示扩展只依赖 NCCL 和 NVSHMEM。
- 强制安装 `nvidia-nccl-cu13==2.30.7` 会覆盖环境中其他 NCCL 版本，可能影响其他依赖 NCCL 的组件；使用 `--no-deps` 缓解了依赖冲突。
- 分支映射变化可能影响 CI 触发路径，但已在 workflow 解析中验证 2x2 矩阵。
- 无自动化测试覆盖，主要依赖手动验证。
- 影响：影响 DeepEP 专用镜像 (`sgl-deep-ep`) 的构建产物和发布流程，对使用该镜像进行 DeepEP 开发的用户有直接影响。CUDA 13 架构（如 Blackwell）的 wheel 不再包含 GDRCopy，但运行时不再需要 `/dev/gdrdrv`。对主镜像和 CI 运行时不产生直接影响，相关变更延后至 #34923。团队需知晓新的分支命名和依赖策略。
- 风险标记：构建依赖变更，依赖版本锁定，缺少自动化测试

关联脉络

- PR #34923 主镜像和 CI 运行时 NCCL 配置（关联 PR）：PR body 与 issue 评论明确提到将运行时 NCCL 配置留给 #34923。