PR #23593 完整报告

sgl-project/sglang

[Docker] Prep for torch 2.11: cu129 fix, image validator, dep cleanup

合并时间：2026-05-04 15:37

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/23593

执行摘要

- 一句话：为 Torch 2.11 准备 Docker 镜像并修复 cu129 构建
- 推荐动作：对于大多数开发者，此 PR 属于基础设施变更，无需深入阅读。但对于 CI/CD 维护者和 Docker 镜像消费者，建议关注标签策略迁移部分。设计模式值得注意：使用版本别名实现向后兼容过渡，以及通过 --no-deps 精确控制依赖版本的方法。

功能与动机

Torch 2.11 使 cu130 wheels 成为 PyPI 默认，导致 cu129 Docker 镜像构建路径出现两个 install-time bug：sgl-kernel 安装缺少 --force-reinstall --no-deps，使 pip 静默拉取了 cu130 torch；主依赖安装使用 --extra-index-url 不足以强制 cu12x 解析。同时，Torch 2.11 已内置了之前手动覆盖的 NVIDIA 软件包，移除这些覆盖可避免静默降级。此 PR 修复这些问题，完成 #21247 中 Dockerfile 侧的工作。

实现拆解

1. 修复 cu129 torch 解析 (docker/Dockerfile)：在 cu128/cu129 分支为 sgl-kernel 安装添加 --force-reinstall --no-deps；对主 sglang 依赖安装，先用 --index-url 从 pytorch.org cu12x 索引预装 torch/torchvision/torchaudio。
2. 移除过时的 NVIDIA 软件包覆盖 (docker/Dockerfile)：删除 nvidia-nccl-cu12/cu13、nvidia-cudnn-cu12/cu13、nvidia-cublas、nvidia-cutlass-dsl 的强制重新安装。
3. 修复 nixl 重复安装 (docker/Dockerfile)：从通用依赖列表移除 nixl stub 包，改为在 CUDA 版本分支中直接安装 nixl-cu12 或 nixl-cu13，并保留 stub 包确保 import nixl 可用，使用 --no-deps。
4. 调整 Docker 镜像标签策略 (.github/workflows/release-docker-dev.yml, release-docker.yml, release-docker-runtime.yml)：将默认 dev/latest 标签从 cu129 镜像迁移到 cu130 镜像，旧标签作为别名保留。
5. 删除废弃的 docker/diffusion.Dockerfile：不再被引用。 此外，更新默认 ARG CUDA_VERSION 为 13.0.1，收紧 nvidia-cutlass-dsl 约束，移除 deadsnakes PPA。

关键文件：

- docker/Dockerfile（模块 Docker 构建；类别 infra；类型 infrastructure）：核心修复：修复 cu129 torch 解析和 sgl-kernel 安装，添加 nixl 重复安装修复，移除过时的 NVIDIA 覆盖

- .github/workflows/release-docker-dev.yml（模块 CI 工作流；类别 infra；类型 infrastructure）：调整 dev 镜像标签策略，将默认标签从 cu129 切换到 cu130，并添加别名。
- docker/diffusion.Dockerfile（模块 Docker 构建；类别 infra；类型 deletion）：删除废弃的 diffusion Dockerfile，不再被工作流或文档引用。
- .github/workflows/release-docker.yml（模块 CI 工作流；类别 infra；类型 infrastructure）：调整 release 框架镜像标签策略，cu130 标签成为默认等价标签，cu129 添加后缀。
- .github/workflows/release-docker-runtime.yml（模块 CI 工作流；类别 infra；类型 infrastructure）：调整 runtime 镜像标签策略，与 release 镜像标签对齐。
- .github/workflows/trivy-scan-dev.yml（模块 CI 工作流；类别 infra；类型 infrastructure）：将默认扫描标签从 dev-cu13 改为 dev-cu12，与标签迁移对齐。
- scripts/ci/utils/docker_build_metadata_args.py（模块 CI 脚本；类别 infra；类型 infrastructure）：更新构建元数据脚本中的默认标签引用。

关键符号：未识别

关键源码片段

.github/workflows/release-docker-dev.yml

调整 dev 镜像标签策略，将默认标签从 cu129 切换到 cu130，并添加别名。

```
# 构建标签配置，dev-cu13 / nightly-dev-cu13 作为 cu130 镜像的别名发布，
# 用于向后兼容之前固定的名称
if [ -z "${SUFFIX}" ]; then
  # 夜间构建：dev 标签从 cu129 切换到 cu130，保留 cu12 别名
  TAG_CONFIG='[{"cuda":"cu129","tags":["dev-cu12","nightly-dev-cu12-{date}-{short_sha}"]},
  {"cuda":"cu130","tags":["dev","dev-cu13","nightly-dev-{date}-{short_sha}","nightly-dev-cu13-{date}
  -{short_sha}"]}]'
else
  TAG_CONFIG="[{\"cuda\":\"cu129\",\"tags\":[\"dev-cu12${SUFFIX}\"]},{\"cuda\":\"cu130\",\"tags\":
  [\"dev${SUFFIX}\",\"dev-cu13${SUFFIX}\"]}]"
fi
```

评论区精华

审查中主要讨论了 nixl stub 包的安装方式：

- ovidiusm指出 stub 包 nixl 是必要的，因为某些代码路径会 import nixl，建议在 CUDA 12 和 13 分支中同时安装 nixl 和对应的 nixl-cuX，使用 --no-deps 避免版本冲突。
- mmangkad建议使用 nixl[cu12] 和 nixl[cu13] 的语法简化安装，但最终实现采用了 ovidiusm 的方案。
- 另一讨论围绕标签重定向策略，最终决定保留旧标签别名以确保向后兼容。
- nixl 安装方式 (design): 最终采用 ovidiusm 的方案，保留 stub 包并同时安装特定 CUDA 库，使用 --no-deps 避免版本冲突。

风险与影响

- 风险：
  - 构建中断风险：--force-reinstall --no-deps 和预安装 torch 改动改变了依赖解析顺序，若 torch 2.11 未来更新时可能需要调整。
  - 标签兼容性风险：将默认标签从 cu129 切换到 cu130 可能影响已硬编码旧标签（如 dev）的用户。添加了别名缓解，但用户若使用 latest 标签可能无感知地切换了 CUDA 版本。
  - 依赖版本风险：移除了 NVIDIA 覆盖，若 Torch 2.11 后续版本不再捆绑这些库，可能引入版本不匹配。但可能性低。
  - 无运行时风险：变更仅限于构建脚本和 CI 配置，不影响模型推理路径。
- 影响：
  - 用户影响：使用 Docker 镜像的用户可能注意到默认 dev 标签现在指向 cu130 镜像。使用显式标签（如 dev-cu12）的用户无影响。运行时行为不变。
  - 系统影响：Docker 构建将使用新的依赖策略，减少手动覆盖，更加稳健。
  - 团队影响：CI/CD 维护者需注意标签策略变化，确保自动化脚本匹配新标签模式。
  - 影响程度：中等。涉及镜像交付，但通过别名保持了向后兼容。
  - 风险标记：依赖解析变更，标签策略迁移，构建过程变更，无运行时影响

关联脉络

- PR #21247 [Dependency] Upgrade to Torch 2.11.0: 该 PR 是 torch 2.11 升级的 Dockerfile 侧配套，处理了 Dockerfile 的依赖修复和清理。