

PR #2257 完整报告

THUDM/slime

[docker] support cuda 13

合并时间: 2026-08-07 21:48

原文链接: <http://prhub.com.cn/THUDM/slime/pull/2257>

执行摘要

- 一句话: 新增 CUDA 13 (Blackwell) Docker 构建路径
- 推荐动作: 值得精读 docker/Dockerfile 的 CUDA 分支与 docker/justfile 的发布目标, 可作为“多 CUDA 版本 + 多架构镜像”构建的参考模板。重点观察三点: 运行时探测取代显式构建开关的取舍、第三方 fork 依赖的固定 commit 管理、以及滚动 latest 标签的所有权转移。若团队近期要支持 Blackwell 训练, 此 PR 是必要的基础设施前置。

功能与动机

PRbody 明确说明要 'Add a CUDA-13 (Blackwell) build path on the cu130 SGLang base'。

背景是 SGLang 已提供 cu130 基础镜像 (v0.5.15.post1-cu130), 而 Blackwell (B200/B300) 用户在 CUDA 13 环境下需要一条可复现的镜像构建路径; 同时

TransformerEngine 在 CUDA 13 上还没有预编译 wheel, 必须从源码构建, 因此需要单独维护依赖编译顺序和架构参数。

实现拆解

1. Dockerfile 引入 CUDA 版本分支: 新增 DEEPGEMM_COMMIT、DEEPEP_COMMIT、DEEPEP_CUDA_ARCH_LIST、FLASH_QLA_COMMIT、TRANSFORMER_ENGINE_COMMIT、TRANSFORMER_ENGINE_CUDA_ARCHS、TMS_COMMIT 等 ARG, 移除旧的 ENABLE_CUDA_13 开关, 改为在 RUN 中通过 `torch.version.cuda.split(".")[0]` 探测 CUDA 主版本并分支。
2. CUDA 12 路径保持原状: 当探测结果为 12 时, 卸载 SGLang 自带 FA4, 安装 `flash-attn==2.8.3` 并编译 FA3 (flash-attention hopper 子模块, commit 002cce0), 继续使用验证过的 TE 2.16 CP 训练栈。
3. CUDA 13 路径新增编译: 保留 FA4 不卸载; 安装 `nvidia-mathdx==26.6.0` 并从源码编译 TransformerEngine (pin 到 c9877be, TRANSFORMER_ENGINE_CUDA_ARCHS=100a; 103a); 随后编译 zhuzilin fork 的 torch_memory_saver、DeepGEMM、DeepEP, 并为 DeepEP 增加 CCCL include 路径以适配 CUDA 13 中迁移到 `<cuda/std/*>` 的头文件。
4. justfile 发布目标调整: release-primary 改为显式使用 cu129 基础镜像并发布 latest、latest-cu129、<version>-cu129; 新增 release-cu13 使用 cu130 基础镜像、传入 DEEPEP_CUDA_ARCH_LIST=10.0;10.3, 发布 latest-cu130、<version>-cu130; _release-raw 统一按 ARG_TAG_POSTFIX 补打滚动 latest 标签, ARG_TAG_LATEST 控制无后缀 latest 的所有权。

5. 文档与版本配套：docker/README.md 更新构建 / 发布命令、tag 后缀约定、pinned 组件清单与自定义 DEEPEP_CUDA_ARCH_LIST 的用法；docker/version.txt 从 nightly-dev-20260804a 更新为 nightly-dev-20260807a。

关键文件：

- docker/Dockerfile（模块 镜像构建；类别 infra；类型 infrastructure）：核心文件：新增 CUDA 版本分支逻辑、TE 源码编译、zhuzilin fork 组件安装和 CCCL include 路径，直接决定镜像内容。
- docker/justfile（模块 发布脚本；类别 infra；类型 infrastructure；符号 release-primary, release-cu13, _release-raw）：发布脚本：分离 CUDA 12/13 发布目标，调整 latest 标签所有权并新增 release-cu13。
- docker/README.md（模块 构建文档；类别 docs；类型 documentation）：文档同步：更新构建命令、tag 约定、pinned 组件清单和 arch 参数说明。
- docker/version.txt（模块 版本标记；类别 docs；类型 documentation）：版本号标记更新，标识构建镜像对应的 nightly 版本。

关键符号：未识别

关键源码片段

docker/Dockerfile

核心文件：新增 CUDA 版本分支逻辑、TE 源码编译、zhuzilin fork 组件安装和 CCCL include 路径，直接决定镜像内容。

```
# 新增构建参数固定第三方组件 commit，保证可复现构建。
ARG DEEPGEMM_COMMIT=b38a77cd193cf38f670caae192310521d24343be
ARG DEEPEP_COMMIT=6845ffd9d59126ec0030c13e0e155935a61e5b5a
ARG DEEPEP_CUDA_ARCH_LIST=
ARG TRANSFORMER_ENGINE_COMMIT=c9877beb87ad7e711e1869dd0b5062167ede447a
ARG TRANSFORMER_ENGINE_CUDA_ARCHS=100a;103a
ARG TMS_COMMIT=8d30c59ca12a68d9deccbc9c6599076a1218cbc5

# 以运行时探测到的 CUDA 主版本决定分支，取代旧的 ENABLE_CUDA_13 开关。
# CUDA 12 走验证过的 FA2 + FA3 训练栈并卸载 SGLang 自带 FA4；
# CUDA 13 (Blackwell) 保留 FA4，避免其推理 kernel 被覆盖。
RUN if [ "$(python -c 'import torch; print(torch.version.cuda.split(".")[0])' )" = "12" ]; then pip
uninstall -y flash-attn-4 flash_attn_4 || true; MAX_JOBS=64 pip -v install flash-attn==2.8.3 --no-
build-isolation; fi
```

docker/justfile

发布脚本：分离 CUDA 12/13 发布目标，调整 latest 标签所有权并新增 release-cu13。

```
# CUDA 12 主发布：基于 cu129 基础镜像，拥有无后缀 latest 标签。
release-primary:
  ARG_TAG_POSTFIX="-cu129"
  ARG_TAG_LATEST=1
  ARG_BUILD_EXTRA_ARGS='--build-arg SGLANG_IMAGE_TAG=v0.5.15.post1-cu129'
```

```
just _release-raw
```

CUDA 13 (Blackwell) 发布：基于 cu130 基础镜像，DeepEP 需要显式 Blackwell arch。

```
release-cu13:
```

```
ARG_TAG_POSTFIX="-cu130"
```

```
ARG_BUILD_EXTRA_ARGS="--build-arg DEEPEP_CUDA_ARCH_LIST=10.0;10.3 --build-arg
```

```
SGLANG_IMAGE_TAG=v0.5.15.post1-cu130"
```

```
just _release-raw
```

评论区精华

该 PR 没有任何 review 评论 (comments_count 与 review_comments_count 均为 0)，由作者 zhuzilin 直接合并，公开讨论记录为空。设计决策只能从 PR body、代码差异和提交历史推断。

- 暂无高价值评论线程

风险与影响

- 风险：
 - 第三方 fork 供应链风险：CUDA 13 路径编译的 torch_memory_saver、DeepGEMM、DeepEP 均来自 zhuzilin 个人 fork，其中 DeepEP 使用 align_fp8_quantization 分支（GLM-5 低延迟对齐），commit 虽已固定，但没有上游 review 背书，镜像内容审计需额外关注。
 - 运行时探测分支的脆弱性：Dockerfile 用 `python -c 'import torch; print(torch.version.cuda...)'` 探测 CUDA 主版本，一旦基础镜像的 PyTorch 行为变化（例如返回 13.0 之外的格式），分支可能静默落入错误路径；相比显式 ARG 更隐蔽，排障成本更高。
 - DeepEP arch 列表传递风险：release-cu13 以分号分隔的 10.0;10.3 作为 DEEPEP_CUDA_ARCH_LIST，依赖 PyTorch 解析器将其规范化为空格，且 _release-raw 中 \$ARG_BUILD_EXTRA_ARGS 未加引号展开；若 shell 分词或 Docker 参数解析与预期不符，会产生错误 arch 的 DeepEP kernel。
 - 缺少 CI/ 实机验证：CUDA 13 构建路径没有配套 CI 任务，README 中验证清单仍停留在 64xH100（CUDA 12），Blackwell 实机验证情况不透明，发布通道处于未充分验证状态。
 - latest 标签所有权变更：latest 现在由 release-primary（CUDA 12、带 cu129 后缀）独占，依赖旧 just release 无后缀构建的脚本或用户需要迁移到新命令。
- 影响：
 - 用户侧：Blackwell（B200/B300）用户可通过 latest-cu130 或 <version>-cu130 镜像在 CUDA 13 环境运行；CUDA 12 用户继续使用 latest/latest-cu129，无破坏性变更。
 - 系统侧：镜像发布矩阵从单通道扩展为 CUDA 12（cu129）+ CUDA 13（cu130）双通道，构建时长和镜像仓库存储开销增加；Dockerfile 复杂度上升，后续 SGLang/ 依赖升级需要同步维护两条路径。

- 团队侧：发布操作新增 `just release-cu13` 命令，README 明确了 tag 规则；后续升级需在 cu130 基础镜像上重新验证 TransformerEngine 源码编译与 DeepEP 架构参数。
- 风险标记：第三方 fork 依赖，运行时探测分支，缺少 CI 验证，发布流程变更

关联脉络

- PR #2252 [release] bump to v0.3.1: 同期发布流程与 Docker patch 对齐，本 PR 的 cu130 构建延续同一 SGLang 版本线。
- PR #2228 [docker] upgrade sglang to v0.5.15.post1: 本 PR 使用的 cu129/cu130 基础镜像 tag 均基于 v0.5.15.post1，构建参数与 patch 继承自该升级。
- PR #2248 Support PYTORCH_ALLOC_CONF: 同样修改 docker/Dockerfile 与 version.txt，属于 Docker 构建基础设施的连续演进。
- PR #2180 Add --release-train: 涉及 docker patch 与 release 流程调整，与本 PR 同属镜像发布基础设施的迭代。