

PR #51184 完整报告

vllm-project/vllm

[Docker] Cache test dependencies before vLLM install

合并时间: 2026-08-11 01:34

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/51184>

执行摘要

- 一句话: Docker 测试镜像分层缓存优化, 源码变更不再重装依赖
- 推荐动作: 这是一次小范围但收益明确的 CI 优化, 不必精读; 核心设计决策是“把稳定依赖与 per-commit 产物分层”以最大化 BuildKit 层缓存复用, 对维护 Docker 镜像的团队有参考价值。建议合并后观察 Buildkite 测试镜像构建时间是否如预期下降, 并关注 test 阶段 wheel 安装是否稳定复现。

功能与动机

PR body 指出, 在缓存命中的 Build image job 中, 测试镜像在安装 Git 和开发依赖之前就 ADD 了整个仓库, 源码变更会让两个稳定层同时失效: Git 安装 18.9 秒 / 65,523,093 压缩字节, 开发依赖 26.0 秒 / 894,989,258 压缩字节, 合计约 45 秒执行时间和 916 MiB 重新生成 / 推送。将 test-deps 分支提前到 per-commit 产物之前, BuildKit 可以复用这两层直到其真实输入 (依赖文件) 发生变化。blobless checkout 优化被单独跟踪在 [vllm-project/ci-infra#455](#)。

实现拆解

1. 拆分运行时基座: 将 docker/Dockerfile 中原来的 FROM \${FINAL_BASE_IMAGE} AS vllm-base 改为 FROM \${FINAL_BASE_IMAGE} AS vllm-runtime-base, 保留原有系统依赖安装; 随后新增 FROM vllm-runtime-base AS vllm-base, 把 vLLM 与 EP kernel wheel 的安装收在 vllm-base 内, 生产镜像构建路径不变。
2. 新增 test-deps 稳定层: FROM vllm-runtime-base AS test-deps, 先 COPY requirements/common.txt、requirements/lint.txt、requirements/test/cuda.in、requirements/test/cuda.txt, 再安装 Git 与 requirements/dev.txt (开发依赖), 这些层只随依赖文件变化而失效。
3. 重写 test 阶段: FROM test-deps AS test, 先通过 bind mount 从 build 阶段安装 vLLM wheel (含 torch nightly 分支) 与 EP kernel wheel, 设置 LD_LIBRARY_PATH, 最后 ADD . /vllm-workspace/ 加入源码并安装 tests/vllm_test_utils。
4. 配套工具与文档: tools/pre_commit/update-dockerfile-graph.sh 与 docs/contributing/dockerfile/dockerfile.md 在 dockerfilegraph 命令中增加 --concentrate, 重新生成 docs/assets/contributing/dockerfile-stages-dependency.png, 只改可视化、不改依赖关系。

5. 验证：本地执行 `pytest tests/tools/test_docker_build_metadata_args.py` (7 passed)、`docker buildx bake --print test`、`docker buildx build --call=check --target test` 均通过；完整 CUDA 镜像构建未在本本地运行，交由 Buildkite CI 验证。

关键文件：

- `docker/Dockerfile` (模块 镜像构建；类别 infra；类型 infrastructure；符号 `vllm-runtime-base`, `vllm-base`, `test-deps`, `test`)：核心变更：重排测试镜像 stage，新增 `vllm-runtime-base`、`test-deps` 并将 `test` 阶段改为先装 wheel 后 ADD 源码，实现稳定依赖层缓存复用。
- `tools/pre_commit/update-dockerfile-graph.sh` (模块 图表生成；类别 other；类型 core-logic)：为 `dockerfilegraph` 命令增加 `--concentrate` 参数，合并依赖图中的平行边，保证新增 `test` 分支可读。
- `docs/contributing/dockerfile/dockerfile.md` (模块 贡献指南；类别 docs；类型 documentation)：记录新 stage 划分与依赖图生成命令变化，更新贡献者文档。
- `docs/assets/contributing/dockerfile-stages-dependency.png` (模块 文档配图；类别 docs；类型 documentation)：重新生成的 Docker stage 依赖图，展示新增 `test-deps` 分支结构。

关键符号：`vllm-runtime-base`, `test-deps`, `vllm-base`, `test`

关键源码片段

`docker/Dockerfile`

核心变更：重排测试镜像 stage，新增 `vllm-runtime-base`、`test-deps` 并将 `test` 阶段改为先装 wheel 后 ADD 源码，实现稳定依赖层缓存复用。

```
# vllm-runtime-base 阶段只安装运行时系统依赖，作为稳定基座，
# 后续 vllm-base 与 test-deps 都从它派生，避免重复安装。
FROM ${FINAL_BASE_IMAGE} AS vllm-runtime-base
# （此处保留原有系统依赖安装步骤，未在本 PR 中改动）
```

```
# vllm-base 阶段在稳定基座上继续安装 vLLM 与 EP kernel wheel，
# 生产镜像构建路径的行为保持不变。
FROM vllm-runtime-base AS vllm-base
# （wheel 安装逻辑保持原有实现）
```

```
# test-deps 阶段先把 requirements 文件 COPY 进镜像，再安装 Git 与开发依赖，
# 这样只有依赖文件变化时这些层才失效，源码变更不会触发重装。
FROM vllm-runtime-base AS test-deps
COPY requirements/common.txt requirements/common.txt
COPY requirements/lint.txt requirements/lint.txt
COPY requirements/test/cuda.in requirements/test/cuda.in
COPY requirements/test/cuda.txt requirements/test/cuda.txt
RUN apt-get update -y \
    && apt-get install -y git ... \
    && --mount=type=cache,target=/opt/uv/cache \
    uv pip install --system -r requirements/dev.txt ...
```

```
# test 阶段先安装 per-commit 的 vLLM/EP wheel, 再把仓库源码 ADD 进来,
# 让最易变的源码处于最上层, 最大化 BuildKit 层缓存复用。
FROM test-deps AS test
COPY --from=build /workspace/dist/wheel.sha256 /tmp/vllm-wheel.sha256
RUN --mount=type=bind,from=build,src=/workspace/dist,target=/vllm-workspace/dist \
    --mount=type=cache,target=/opt/uv/cache \
    uv pip install --system dist/*.whl --verbose # 安装 vLLM wheel
ADD . /vllm-workspace/ # 源码最后加入, 避免污染稳定层
RUN --mount=type=cache,target=/opt/uv/cache \
    uv pip install --system -e tests/vllm_test_utils
```

评论区精华

本 PR 几乎没有技术讨论: claude[bot] 因 PR 来自 fork 而禁用自动 review; 唯一的人类批准来自 Harry-Chen, 仅回复 "Thanks!". mergify 曾提示存在 merge conflict, 作者 rebase (66b69d3、f82ef3b) 后重新触发 Buildkite CI (#82548、#82558) 并解决。第三个 commit 补上 requirements/common.txt 的 COPY, 说明依赖文件拷贝顺序是容易被遗漏的点。

- 暂无高价值评论线程

风险与影响

- 风险:

1. 构建顺序变化: test 阶段现在从 test-deps 派生, wheel 安装依赖 build 阶段的 bind mount (wheel.sha256、torch_lib_versions.txt), 若 CI 中该产物路径变化或缺失会直接失败; PR 未本地跑完整 CUDA 镜像, 需 Buildkite 全量验证。
2. stage 重命名兼容性: 原 vllm-base 保留但拆出新 vllm-runtime-base, 任何直接引用旧 stage 的外部脚本不受影响; 但新基座名称需要团队知晓, 避免后续维护时误用。
3. 依赖文件完整性: test-deps 需要同时 COPY 多份 requirements 文件, 且 cuda.in 与 cuda.txt 有生成关系; 第三次 commit 补上 requirements/common.txt 说明该处容易遗漏。
4. 缓存语义变化: 源码变更不再触发依赖重装, 但若开发依赖需要跟随源码变化 (如 tests/vllm_test_utils), 该包已在 ADD 之后安装, 逻辑正确; 反过来若误把源码相关文件放入依赖层, 会引入缓存污染风险。- 影响: 影响集中在 CI/Docker 镜像构建: 对 source-only commit, 预计每次构建减少约 45 秒执行时间和 916 MiB 层生成 / 推送量, 缓存命中场景收益显著; 生产 vllm-base 镜像路径与开发者镜像保持不变, 对最终用户无影响。需要团队维护 Dockerfile 的人员理解新的 stage 划分 (vllm-runtime-base / test-deps / test), 避免后续改动破坏分层缓存策略。- 风险标记: 镜像构建流程变更, 未完成全量镜像构建验证, 阶段重命名兼容性, 依赖层缓存策略

关联脉络

- PR #51424 [Build] Skip precompiled wheel fetch during metadata hooks: 同为构建基础设施优化, 减少 CI 镜像 / 依赖拉取开销, 与本 PR 构成 CI 缓存优化系列。

- PR #51604 [CI][XPU] Add VLLM_DISABLE_COMPILE_CACHE=1 for other random failed cases in Intel GPU CI: 同属 CI 构建基础设施与缓存稳定性相关优化，反映仓库近期在构建层缓存上的持续投入。