PR #44266 完整报告

vllm-project/vllm

[Bugfix][CI] Normalize NIXL connector CUDA wheel installs

合并时间：2026-06-02 10:34

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/44266

执行摘要

- 一句话：修复 CI 中 NIXL 连接器 CUDA wheels 安装
- 推荐动作：值得快速合并，修复明确的 CI bug。建议后续关注：若出现新的 CUDA 版本（如 cu14），需更新脚本；也可考虑将 CUDA 版本检测逻辑放入共享的 CI 基础镜像中。

功能与动机

在 CI 中，`uv pip install -r requirements/kv_connectors.txt` 会安装所有 CUDA 变体的 nixl 包（如 nixl-cu12、nixl-cu13）。在 CUDA 13 镜像中，`import nixl_ep` 可能错误加载 CUDA 12 扩展，导致 `ImportError: libcudart.so.12: No such file or directory`。此问题最初在 PR#44083 的 CI 运行中被发现。

实现拆解

1. 创建安装辅助脚本(buildkite/scripts/install-kv-connectors.sh)：新脚本首先从 requirements/kv_connectors.txt 安装所有 NIXL 依赖（包括所有 CUDA 变体），然后通过 Python 内联脚本读取 torch.version.cuda 获取当前 CUDA 主版本号，以及 nixl 包的版本号。接着卸载所有不匹配的 nixl-cu* 包（如 nixl-cu12、nixl-cu13），最后安装仅匹配当前 CUDA 主版本的 wheel，且版本与之前安装的 nixl 一致，从而确保 nixl_ep 链接正确的 libcudart。
2. 修改 CI 配置：在 .buildkite/test_areas/disaggregated.yaml 中，将原本 9 处分散的 uv pip install --system -r /vllm-workspace/requirements/kv_connectors.txt 统一替换为 bash /vllm-workspace/.buildkite/scripts/install-kv-connectors.sh；在 .buildkite/test_areas/misc.yaml 中同样替换了 V1 Core + KV + Metrics 任务中的对应行。
3. 保持影响范围最小：仅在已手动安装连接器的 CI job（disaggregated 和 misc 中的特定步骤）中使用新脚本，不将 NIXL 依赖加入共享 CI 镜像，避免无关 job 引入非必需包。

关键文件：

- .buildkite/scripts/install-kv-connectors.sh（模块 CI 脚本；类别 other；类型 dependency-wiring）：核心修复脚本：规范化 NIXL wheel 安装，避免 CUDA 版本错配。
- .buildkite/test_areas/disaggregated.yaml（模块 CI 配置；类别 config；类型 configuration）：更新 9 个测试步骤，用新脚本替换内联 pip install 命令。
- .buildkite/test_areas/misc.yaml（模块 CI 配置；类别 config；类型 configuration）：更新 'V1 Core + KV + Metrics' 步骤，使用新脚本。

关键符号：未识别

关键源码片段

.buildkite/scripts/install-kv-connectors.sh

核心修复脚本：规范化 NIXL wheel 安装，避免 CUDA 版本错配。

```
# install-kv-connectors.sh
# SPDX-License-Identifier: Apache-2.0

set -euo pipefail

# 允许通过环境变量覆盖 requirements 文件路径，默认指向工作空间中的文件
REQUIREMENTS_FILE="${KV_CONNECTORS_REQUIREMENTS:-/vllm-workspace/requirements/kv_connectors.txt}"

# 第一步：安装所有 NIXL 依赖（包括所有 CUDA 变体）
uv pip install --system -r "${REQUIREMENTS_FILE}"

# 第二步：检测当前 CUDA 主版本号和已安装的 nixl 版本
# 使用 Python 内联脚本获取，避免依赖外部工具
NIXL_METADATA=$(python3 - <<'PY'
import importlib.metadata as metadata
import torch

cuda_version = torch.version.cuda
if cuda_version is None:
    # 若 torch 无法检测 CUDA，则退出（适用于 CPU 镜像）
    raise SystemExit("torch.version.cuda is not set")

# 输出 CUDA 主版本和 nixl 版本，后续用 read 解析
print(cuda_version.split(".", 1)[0], metadata.version("nixl"))
PY
)
read -r CUDA_MAJOR NIXL_VERSION <<<"${NIXL_METADATA}"

# nixl>=1.1.0 会安装多个 CUDA 变体，此处移除所有不匹配的变体
# 这样 nixl_ep_cpp 只会链接当前可用的 libcudart
uv pip uninstall --system nixl-cu12 nixl-cu13 2>/dev/null || true

# 第三步：仅安装与当前 CUDA 主版本匹配的 wheel，版本锁为刚检测到的 nixl 版本
uv pip install --system --no-deps "nixl-cu${CUDA_MAJOR}==${NIXL_VERSION}"

# 第四步：打印最终安装状态，便于 CI 日志排查
python3 - <<'PY'
import importlib.metadata as metadata
for package_name in ("nixl", "nixl-cu12", "nixl-cu13"):
    try:
        version = metadata.version(package_name)
```

```
    except metadata.PackageNotFoundError:
        version = "not installed"
    print(f"{package_name}: {version}")
PY
```

评论区精华

该 PR 无 review 评论。合并者 njhill 直接批准。

- 暂无高价值评论线程

风险与影响

- 风险：低风险。变更仅限于 CI 配置和新脚本，不涉及生产代码。但需注意：
  - 若 torch.version.cuda 返回 None（如 CPU 环境），脚本会以非零退出，可能需在 CI 中处理此异常路径。
  - 依赖 nixl 包的版本必须与 nixl-cu* 严格对应，否则版本不匹配可能导致其他问题。
  - 影响：影响范围限于 CI 中运行 NIXL 连接器测试的 job（disaggregated 测试和 V1 Core + KV + Metrics 测试）。修复后，这些 job 不会再因 CUDA 版本错配而失败，提高 CI 稳定性。对其他 job 和用户无影响。
  - 风险标记：依赖版本锁定，CUDA 检测退化风险

关联脉络

- PR #44083 [Bugfix][CI] ...: 导致了本 PR 修复的问题，PR body 明确提及。
- PR #44192 [CI] ...: 提供了另一种更宽泛的修复思路（将 NIXL 依赖加入共享镜像），本 PR 是更窄的替代方案。