PR #34635 完整报告

sgl-project/sglang

[CI] Use default installer for B300 tests

合并时间：2026-08-13 07:21

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/34635

执行摘要

- 一句话：B300 CI 改用默认安装器并删除 Kimi-K3 专用脚本
- 推荐动作：建议快速浏览即可，重点确认两点：一是实际 CI 镜像中 transformers 版本是否已包含 PR #46618 修复；二是 test_kimi_k3_b300.py rerun 失败是否与本次安装路径切换相关，若持续失败应及时回退或保留补丁。对 CI 基础设施维护者而言，这是一个典型的 " workaround 到期清理 " 案例，值得参考其验证方法（配置解析测试 + 路径存在性校验 + 残留引用检查）。

功能与动机

PR body 明确说明：B300 测试套件应使用标准 CUDA CI 依赖安装路径，而不是维护 Kimi-K3 专用 wrapper（"The B300 test suite should use the standard CUDA CI dependency installation path instead of maintaining a Kimi-K3-specific wrapper."）。此前为绕开 Transformers 5.12.1 的 custom-code symlink 解析问题，专门维护了该 wrapper；上游 PR #46618 修复后，该 workaround 应当可安全移除。

实现拆解

1. 切换安装路径：修改 scripts/ci/runner_configs.yml，将 8-gpu-b300 的 install 从 *kimi_k3 锚点改为 *default，并删除 kimi_k3_install 锚点定义，使 B300 与 1-gpu-small、8-gpu-h200 等机型共用 scripts/ci/cuda/ci_install_dependency.sh。
2. 删除专用脚本：移除 scripts/ci/cuda/ci_install_kimi_k3.sh（58 行）。该脚本先 source 标准安装脚本保持 Python/venv 选择状态，再通过 Python 内联脚本对 transformers.dynamic_module_utils 打 symlink 热补丁，并自带 HF cache 布局自测。删除后不再有单机型专用安装逻辑。
3. 配套验证：执行 python3 -m unittest scripts.ci.test_list_stage_models（34 个测试通过）、python3 scripts/ci/runner_configs.py 8-gpu-b300（解析为标准安装器）、校验全部 10 个 runner 配置的安装路径存在、确认无残留 Kimi-K3 installer 引用，并通过 pre-commit。
4. 硬件验证留待 CI：PR body 声明实际 B300/Kimi-K3 模型启动需要 CI 硬件验证；作者随后 rerun 的 test_kimi_k3_b300.py 在 8-gpu-b300 上失败一次，结果未定性。

关键文件：

- scripts/ci/runner_configs.yml（模块 CI 配置；类别 infra；类型 infrastructure）：B300 安装路径切换的配置入口，直接决定 8-gpu-b300 走标准安装器还是专用 wrapper。
- scripts/ci/cuda/ci_install_kimi_k3.sh（模块 安装脚本；类别 infra；类型 deletion；符号 apply_transformers_symlink_patch）：被删除的专用安装脚本，包含 Transformers symlink 热补丁，是本 PR 的主要清理对象。

关键符号：apply_transformers_symlink_patch

关键源码片段

scripts/ci/runner_configs.yml

B300 安装路径切换的配置入口，直接决定 8-gpu-b300 走标准安装器还是专用 wrapper。

scripts/ci/runner_configs.yml

```
# 8-gpu-b300 回归标准 CUDA CI 安装路径，不再为单一机型维护专用 wrapper
_anchors:
  default_install: &default scripts/ci/cuda/ci_install_dependency.sh
  # kimi_k3_install 锚点已在本 PR 中删除

runner_configs:
  8-gpu-b300: { install: *default, artifact_version: v6, install_timeout: "20", runs_on: 8-gpu-b300
  }
```

scripts/ci/cuda/ci_install_kimi_k3.sh

被删除的专用安装脚本，包含 Transformers symlink 热补丁，是本 PR 的主要清理对象。

scripts/ci/cuda/ci_install_kimi_k3.sh（本 PR 已删除）

```
#!/bin/bash
# 原用途：安装标准 CUDA CI 依赖后，再修复 Transformers 的 custom-code
# symlink 解析问题（上游 PR #46618 修复前），本 PR 整体删除该 wrapper。
set -euxo pipefail

SCRIPT_DIR="$(cd "$(dirname "${BASH_SOURCE[0]}" )" && pwd)"

# source 而非直接执行，保留通用安装脚本对 Python/venv 的选择状态
source "${SCRIPT_DIR}/ci_install_dependency.sh" "$@"

apply_transformers_symlink_patch() {
  # transformers 5.12.1 会把 custom-code symlink 解析到 blobs/ 目录，
  # 再按文件名找相对导入失败；此处镜像 docker/rocm.Dockerfile 的热补丁。
  python3 - <<'PY'
import importlib
import pathlib
import tempfile

import transformers.dynamic_module_utils as dynamic_module_utils

marks = ["Path(resolved_module_file).resolve()", "Path(source_file).resolve()"]
```

```
path = pathlib.Path(dynamic_module_utils.__file__)
src = path.read_text()
if not any(mark in src for mark in marks):
    print("transformers dynamic_module_utils already fixed; no patch needed")
else:
    patched = src.replace("Path(resolved_module_file).resolve()", "Path(resolved_module_file)")
    patched = patched.replace("Path(source_file).resolve()", "Path(source_file)")
    assert patched != src, "FATAL: transformers symlink patch matched nothing"
    path.write_text(patched)
    print("patched transformers dynamic_module_utils.py (symlink hash fix)")

# 端到端自测：复现 HF cache 布局，确认 symlink 哈希计算正确
dynamic_module_utils = importlib.reload(dynamic_module_utils)
with tempfile.TemporaryDirectory() as tmp:
    cache = pathlib.Path(tmp) / "models--org--model"
    blobs = cache / "blobs"
    snapshot = cache / "snapshots" / "revision"
    blobs.mkdir(parents=True)
    snapshot.mkdir(parents=True)
    (blobs / "modeling-blob").write_text("from .media_utils import VALUE\n")
    (blobs / "media-blob").write_text("VALUE = 1\n")
    (snapshot / "modeling.py").symlink_to("../../blobs/modeling-blob")
    (snapshot / "media_utils.py").symlink_to("../../blobs/media-blob")
    source_hash = dynamic_module_utils._compute_local_source_files_hash(snapshot, snapshot /
    "modeling.py")
    assert len(source_hash) == 16, source_hash
PY
}

apply_transformers_symlink_patch
```

评论区精华

该 PR 没有 review 评论或设计讨论。唯一的交互是作者发起的 rerun：/rerun-test test/registered/models_e2e/test_kimi_k3_b300.py，github-actions 返回 😕（8-gpu-b300 上 1 个测试失败）。PR body 已提前声明实机启动需 CI 硬件验证，因此该失败与本变更的因果关系尚未确认，也没有后续讨论跟进。

- B300 Kimi-K3 rerun 测试失败 (testing): 未确认是否与本 PR 的安装路径切换相关；PR body 声明实机启动需 CI 硬件验证，失败结果未进一步归因。

风险与影响

- 风险：
  1. Transformers 热补丁移除风险：删除的 symlink 补丁依赖上游 PR #46618 已进入实际安装的 transformers 版本；若 CI 镜像仍使用旧版本，B300 上 Kimi-K3 custom-code 模型加载可能失败。

2. 实机回归信号：test/kimi_k3_b300.py rerun 失败一次，虽然不能直接归因于本变更，但需确认失败原因不是安装路径切换导致（例如依赖缺失或补丁缺失）。
3. 配置锚点影响面：kimi_k3_install 锚点删除前需确认无其他 runner 引用，PR 已通过脚本校验全部 10 个路径存在，但未列出校验脚本细节。
4. 影响范围限定：变更仅涉及 CI 依赖安装，不触碰推理路径，对线上服务无回归风险。 - 影响：对用户无直接行为影响。对系统而言，8-gpu-b300 后续所有 PR 测试和夜间任务的依赖安装将走标准 CUDA 脚本，安装时间与行为可能变化，且不再享受 Transformers symlink 补丁保护。对团队而言，减少了一个专用脚本的维护负担，符合 #33997 之后持续清理 Kimi-K3 workaround 的方向；但 B300 上 Kimi-K3 测试的稳定性将取决于上游修复是否已随依赖落地。 - 风险标记：删除了 Transformers 热补丁，B300 实机验证失败，CI 基础设施变更

关联脉络

- PR #33997 Bump FlashInfer to 0.6.17 and remove Kimi K3 workarounds: 同一清理脉络：该 PR 移除 Kimi-K3 的 cubin pool 与 DCP 补丁，本 PR 继续移除 Kimi-K3 的 CI 安装 wrapper，两者目标一致。
- PR #33465 [Kimi-K3][NPU] Support Kimi-K3 on NPU: Kimi-K3 模型支持的来源 PR，本 PR 删除的 wrapper 服务于 B300 上的 Kimi-K3 测试，属于同一模型功能线。