PR #33932 完整报告

sgl-project/sglang

Install DeepEP from release wheels

合并时间：2026-08-08 06:38

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/33932

执行摘要

- 一句话：CI 改用 sgl-deep-ep wheel，移除 DeepEP 源码构建
- 推荐动作：值得精读，尤其是 CI 负责人。这是一个典型的「临时构建脚本 -> 正式发布制品」收敛案例：不仅替换安装方式，还同步清理了 runner 配置、workflow job、slash 命令字段与测试注册，31 个文件跨度但模式统一。值得关注的设计点：按 GPU 数量条件安装 GDRCopy、双发行名（deep-ep/sgl-deep-ep）卸载防混合安装、CUDA 12/13 双通道 wheel 解析、以及用测试夹具保护 runner label 聚合逻辑。建议合并后观察 1-2 周 CUDA 12/13 多卡 CI 稳定性，尤其是 H200/B200/GB 系列 runner。

功能与动机

PR body 明确说明动机：SGLang now publishes sgl-deep-ep wheels, so CUDA CI should consume the released wheel instead of rebuilding DeepEP from source in dedicated runner configurations and test suites. 旧路径需要 git clone 固定 commit、检测 CUDA arch、编译 GDRCopy 并打 setup.py 补丁，依赖安装耗时约 4 分钟；改用发布 wheel 后固定版本、直接安装，降至 30 秒。PR 同时清理了 FORCE_REBUILD_DEEPEP、grace_blackwell 等只为源码构建服务的 CI 控制键。

实现拆解

1. 依赖声明与 wheel 安装通道：python/pyproject.toml 新增 "sgl-deep-ep==0.1.0"（review 中由 0.1.0rc2 收尾修正为正式版）。CUDA 13 runner 直接消费 PyPI 公共 wheel；CUDA 12 runner 由 scripts/ci/cuda/ci_install_dependency.sh 新增的 install_cuda12_deepep_wheel() 从 https://docs.sglang.ai/whl/${CU_VERSION}/ index 安装带本地版本后缀的 wheel（如 0.1.0+cu126），并配合 --force-reinstall --no-deps 保证与 pyproject pin 严格一致；setup_pip_toolchain() 中扩展卸载逻辑，同时卸载 deep-ep 与 sgl-deep-ep 两个发行名，避免 sdist 与 wheel 混合安装残留。
2. 删除源码构建路径：整体删除 scripts/ci/cuda/ci_install_deepep.sh（177 行），包含 GDRCopy 源码编译、DeepEP git clone 固定 commit、CUDA arch 检测、CUDA 13 cccl include 补丁等逻辑；同时删除 pr-test.yml / pr-test-extra.yml 中的 FORCE_REBUILD_DEEPEP 环境变量与 6 个专用 *-deepep-* job，并为归并目标 job 补齐 warmup_deep_gemm_models、timeout_per_file 等原 deepep job 独有的参数。
3. GDRCopy 条件安装：ci_install_dependency.sh 新增 install_gdrcopy()，通过 nvidia-smi 统计 GPU 数量，仅在 >= 4 GPU 的 runner 上执行 GDRCopy 的 DKMS/ 包构

建（DeepEP 测试只在 4+ GPU 上运行），并用 ldconfig 探测 libgdrapi.so 做幂等保护，避免 1/2 GPU job 上无谓的编译开销。

4. 测试注册与 runner 归并：16 个 *-deepep 注册测试的 register_cuda_ci(..., runner_config=...) 从 deepep-4-gpu-b200 / deepep-8-gpu-h200 等改为标准 4-gpu-b200 / 8-gpu-h200；scripts/ci/runner_configs.yml 删除 deepep 条目，test/run_suite.py 的套件表删除 6 个 deepep 套件；scripts/ci/utils/slash_command_handler.py 与 .github/workflows/rerun-test.yml 移除 grace_blackwell 字段。NPU 的 DeepEP 安装路径保持不变。
5. 测试与验证配套：scripts/ci/test_list_stage_models.py 用 alternate-1-gpu 替换 fake runner 中的 deepep-1-gpu，继续验证「多个 runner_config 共享同一 label 时的 union 聚合」逻辑；PR 验证了 workflow YAML/pyproject 解析、base/extra 分区计算、34 个 CI model inventory 单测，以及 16 个迁移注册的套件并集与 stage 元数据等价。

关键文件：

- scripts/ci/cuda/ci_install_dependency.sh（模块 安装脚本；类别 infra；类型 infrastructure；符号 install_gdrcopy, install_cuda12_deepep_wheel, setup_pip_toolchain）：CI 依赖安装的唯一入口，本 PR 的核心新增逻辑（CUDA 12 wheel 安装、GDRCopy 条件安装、双发行名卸载）都集中在这里。
- scripts/ci/cuda/ci_install_deepep.sh（模块 安装脚本；类别 infra；类型 deletion）：177 行 DeepEP 源码构建脚本被整体删除，是本 PR 意图的最直接体现。
- .github/workflows/pr-test.yml（模块 CI 工作流；类别 infra；类型 infrastructure）：标准 CI 主工作流，删除 3 个 base-c-test-deepep-* 专用 job 与 FORCE_REBUILD_DEEPEP 环境变量，并为归并目标 job 补齐 warmup 与超时参数。
- .github/workflows/pr-test-extra.yml（模块 CI 工作流；类别 infra；类型 infrastructure）：extra 工作流同步删除 3 个 extra-b-test-deepep-* 专用 job，保持 base/extra 分区结构一致。
- test/registered/models_e2e/test_deepseek_v4_flash_fp4_b200.py（模块 模型测试；类别 test；类型 test-coverage；符号 register_cuda_ci）：具代表性的 e2e 注册迁移：runner_config 从 deepep-4-gpu-b200 改回标准 4-gpu-b200，测试并入标准 B200 套件。
- scripts/ci/utils/slash_command_handler.py（模块 命令处理；类别 infra；类型 infrastructure；符号 _resolve_runner_config, detect_suite, _dispatch_batch, handle_rerun_test）：/rerun-test 的分派配置与 runner_configs.yml 强耦合，本 PR 移除了 grace_blackwell 字段在解析、分派、日志全链路的传递。

关键符号：install_cuda12_deepep_wheel, install_gdrcopy, setup_pip_toolchain, register_cuda_ci, _resolve_runner_config

关键源码片段

scripts/ci/cuda/ci_install_dependency.sh

CI 依赖安装的唯一入口，本 PR 的核心新增逻辑（CUDA 12 wheel 安装、GDRCopy 条件安装、双发行名卸载）都集中在这里。

```
# 从 python/pyproject.toml 解析 sgl-deep-ep 的固定版本号，拼上 CUDA 版本后缀，
```

```
# 并从 SGLang 专用 wheel index 安装与本地 CUDA 工具链匹配的 wheel。
# CUDA 13 直接使用 PyPI 公共 wheel；CUDA 12 的 wheel 只发布在 SGLang index 上，
# 且带本地版本后缀（如 0.1.0+cu126），本地版本满足 pyproject 的公共版本 pin。
install_cuda12_deepep_wheel() {
    if [ "$CU_MAJOR" = "13" ]; then
        # CUDA 13 走公共 PyPI wheel，无需专用 index，函数到此返回。
        echo "CUDA 13 uses the public sgl-deep-ep wheel declared in python/pyproject.toml"
        mark_step_done "${FUNCNAME[0]}"
        return
    fi

    local version
    version=$(grep -Po -m1 '"sgl-deep-ep==\K[^"]+' python/pyproject.toml || true)
    if [ -z "$version" ]; then
        # 硬保护：pyproject 必须显式 pin，否则后续 editable 安装会解析到错误版本。
        echo "ERROR: python/pyproject.toml must pin sgl-deep-ep"
        exit 1
    fi

    # --force-reinstall --no-deps：确保旧 wheel 被替换，且不连带改动其他依赖；
    # 后续的可编辑 SGLang 安装会复用这个 wheel，而不会解析到错误的版本。
    $PIP_CMD install "sgl-deep-ep==${version}+${CU_VERSION}" \
        --index-url "https://docs.sglang.ai/whl/${CU_VERSION}/" \
        --force-reinstall --no-deps $PIP_INSTALL_SUFFIX

    mark_step_done "${FUNCNAME[0]}"
}
```

test/registered/models_e2e/test_deepseek_v4_flash_fp4_b200.py

具代表性的 e2e 注册迁移：runner_config 从 deepep-4-gpu-b200 改回标准 4-gpu-b200，测试并入标准 B200 套件。

```
# B200 per-commit CI: DeepSeek-V4-Flash FP4（LowLatency recipe）。
# 迁移前该测试注册在 deepep-4-gpu-b200 专用 runner 上，CI 需先源码编译 DeepEP；
# 迁移后 DeepEP 由 sgl-deep-ep wheel 随标准依赖安装，测试并回通用 4-gpu-b200 runner。
register_cuda_ci(est_time=465, stage="base-c", runner_config="4-gpu-b200")

MODEL = "deepseek-ai/DeepSeek-V4-Flash"
SERVER_LAUNCH_TIMEOUT = 3600

# DeepEP dispatch/combine 的 SM 数配置保留在测试中，作为运行时调优参数，
# 与依赖来源（wheel 或源码）无关。
DEEPEP_CONFIG = '{"normal_dispatch":{"num_sms":96},"normal_combine":{"num_sms":96}}'
```

评论区精华

唯一的 review comment 来自作者本人 Fridge003，在 python/pyproject.toml 的 diff 上提出将 sgl-deep-ep==0.1.0rc2 收尾为 sgl-deep-ep==0.1.0 正式版，最终提交采纳。除此之外没有实质性技术争论；Issue 侧只有两次 /rerun-test test/registered/ep/test_deepep_large.py

的 CI 重跑记录，均在 8-gpu-h200 上通过（runs 31152732664 / 31222443590），说明迁移后 EP 大测试在标准 runner 上表现正常。

- sgl-deep-ep 版本 pin 从 rc 收尾为正式版 (question): 已采纳，最终提交 Update python/pyproject.toml 将 pin 定为 0.1.0 正式版，CI 消费正式发布 wheel。

风险与影响

- 风险：

  1. wheel 二进制兼容性：旧源码构建脚本中有 Grace-Blackwell 专用分支（hybrid-ep 分支、NUM_CPU_TIMEOUT_SECS 100->1000 补丁、sm_103 arch）。改为统一 wheel 后，这些细分场景是否被发布 wheel 覆盖完全依赖发布通道，若缺失会导致相关 runner 上 kernel 缺失或运行期错误。
  2. 版本耦合：install_cuda12_deepep_wheel 用 grep 从 python/pyproject.toml 解析 pin，格式一改就会 exit 1（有硬保护）；但 wheel index 上的本地版本（+cu126 等）与公共版本 0.1.0 的兼容性依赖发布流程，任何一侧掉线都会让 CUDA 12 runner 安装失败。
  3. CI 参数逐项对应：原 base-c-test-deepep-4-gpu-b200 / base-c-test-deepep-8-gpu-h200 有 timeout_per_file: '1800' 和 warmup 参数，归并后需确认所有目标 job 都补齐（patch 显示仅部分 job 补充），否则长测试可能超时。
  4. GDRCopy 探测逻辑：nvidia-smi 计数异常（驱动问题、容器限制）会静默跳过 GDRCopy 安装，问题延迟到 DeepEP 运行期才暴露。
  5. 无运行时影响：不涉及模型代码，运行时行为与旧版一致，主要风险集中在 CI 基础设施层。 - 影响：对 CI 运维：所有 CUDA 多卡 runner 的初始化提速（依赖安装 4 min -> 30 s），净删约 212 行脚本与配置，runner_configs.yml、workflow 与 /rerun-test 分派逻辑全面简化。对测试体系：16 个 DeepSeek V4 FP4/FP8 e2e 测试从专用 runner 迁入标准硬件套件（base-c/extra-b），覆盖等价但 job 数量减少 6 个，CI 队列更短。对用户与运行时：零影响，DeepEP 的 dispatch/combine SM 配置等行为不变。对团队：DeepEP 分发正式收敛到 wheel 通道，源码构建脚本不再需要维护，但新增了对「先发布 wheel 再跑 CI」的外部依赖。 - 风险标记：依赖来源切换，CI 覆盖面收缩，wheel 兼容性依赖发布通道，grep 解析 pyproject 强耦合

关联脉络

- PR #33532 [CP]: Support CP V2 Strategy for dsv4: 同属 DeepSeek-V4 功能线，且本 PR 修改了该 PR 引入的 test/registered/cp/test_deepseek_v4_flash_fp4_b200_cp.py 注册配置。
- PR #33616 feat: Add flashinfer mHC fusion for DSV4: 同属 DeepSeek-V4 性能与服务化演进，运行时依赖 DeepEP。
- PR #32556 Autotune flashinfer extend buckets at warmup: CI warmup 参数（warmup_deep_gemm_models）在本次归并的 job 中被显式保留，说明 CI 正在为 DeepSeek 模型做统一的依赖与 warmup 治理。