PR #34321 完整报告

sgl-project/sglang

[Fix] Pin `cuda-tile` to 1.6.0rc5 to unblock Python 3.10 x86_64 installs

合并时间：2026-08-11 06:09

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/34321

执行摘要

- 一句话：钉死 cuda-tile 1.6.0rc5，修复 Python 3.10 x86_64 安装失败
- 推荐动作：值得快速浏览，作为 ' 传递依赖导致安装链路破裂 ' 的典型案例。核心设计决策是：用 == 精确 pin 临时止血，并在代码注释中记录原因和解除条件，避免后续维护者无从下手。建议关注合并后 CI 是否全绿，并跟踪 cuda-tile 1.6.0rc6 是否补发 wheel，以便及时解除 pin。

功能与动机

PR body 明确指出：cuda-tile 是传递依赖（sglang → flashinfer_python[cu13]==0.6.15.post1 → cuda-tile>=1.4.0），在 PyPI 上它是 wheel_stub 占位 sdist，构建时从 pypi.nvidia.com 拉取真实 wheel；1.6.0rc6 发布时缺少 cp310 和 cp313 linux x86_64 wheel，导致 stub 抛出 RuntimeError: Didn't find wheel for cuda-tile 1.6.0rc6。CI 使用 --prerelease allow 安装，因此每个 Python 3.10 x86_64 的 CPU shard 都在 Install dependencies 阶段死亡，一个测试都没跑。

实现拆解

1. 定位根因：通过依赖链分析确认失败源头是 flashinfer_python[cu13]==0.6.15.post1 的传递依赖 cuda-tile>=1.4.0 被解析到了缺失 wheel 的 1.6.0rc6 预发布版。
2. 选定版本：在 python/pyproject.toml 的 dependencies 列表中新增 "cuda-tile==1.6.0rc5"，并添加一行注释解释这是 flashinfer-python 的传递依赖、因 1.6.0rc6 缺 cp310 wheel 而钉死。版本选择依据是 1.6.0rc5 拥有完整 wheel 矩阵（cp310-cp314，linux x86_64/aarch64、win_amd64），且满足 flashinfer 的 >=1.4.0 约束。
3. 验证与合入：提交后用 /tag-and-rerun-ci 触发 CI 重跑验证；最终由作者直接合入。PR 卡片上 CI 状态显示 Base 无记录、Extra 失败，读者应留意合并后的实际 CI 结果。
4. 后续维护：这是 == 精确 pin，需要在上游重新发布完整 1.6.0rcN 后手动 bump 或移除；若 flashinfer 未来提高 cuda-tile 版本下限，此 pin 也可能需要同步调整。没有新增测试或部署配套改动，属于纯配置修复。

关键文件：

- python/pyproject.toml（模块 依赖配置；类别 config；类型 configuration）：唯一变更文件，在依赖列表钉死 cuda-tile==1.6.0rc5，直接修复 Python 3.10 x86_64 安装失败并附

带原因注释，是整个 PR 的核心。

关键符号：未识别

关键源码片段

python/pyproject.toml

唯一变更文件，在依赖列表钉死 cuda-tile==1.6.0rc5，直接修复 Python 3.10 x86_64 安装失败并附带原因注释，是整个 PR 的核心。

```
# python/pyproject.toml —— dependencies 中的关键配置（节选）
dependencies = [
  "aiohttp",
  "anthropic>=0.20.0",
  # ... 其他依赖 ...
  "cuda-python>=13.0",

  # `cuda-tile` 是 `flashinfer_python[cu13]==0.6.15.post1` 的传递依赖，
  # flashinfer 侧约束为 `>=1.4.0`。
  # PyPI 上 `cuda-tile` 是 `wheel_stub` 占位 sdist，
  # 构建时会从 `pypi.nvidia.com` 拉取真实 wheel，
  # 而 `1.6.0rc6` 发布时缺少 cp310 linux x86_64 的 wheel，
  # 导致 stub 抛出 `RuntimeError: Didn't find wheel for cuda-tile 1.6.0rc6`。
  # `1.6.0rc5` 是目前 wheel 矩阵完整（cp310-cp314，含 linux x86_64/aarch64、win_amd64）
  # 且满足 `>=1.4.0` 约束的最新版本，故用 `==` 钉死以恢复安装。
  "cuda-tile==1.6.0rc5",

  "datasets",
  # ... 其他依赖 ...
]
```

评论区精华

该 PR 没有 review 评论，唯一的评论是作者自己的 /tag-and-rerun-ci 指令，用于触发 CI 重跑。本质上没有技术争论，决策依据全部来自 PR body 中对依赖链和 wheel 矩阵的分析。值得记住的是：作者选择了 wheel 矩阵最完整且满足约束的 1.6.0rc5，而不是简单的 ' 最新可用版本 '，这体现了依赖修复中 ' 可用性优先于版本号 ' 的权衡。

- CI 重跑指令 (other): 无技术结论，属于运维性操作；依赖修复的验证依赖 CI 安装阶段是否通过。

风险与影响

- 风险：
  1. 依赖冲突风险：==1.6.0rc5 精确钉死，若 flashinfer 升级要求 cuda-tile>=1.6.0rc6，会导致解析失败，需要人工联动更新。
  2. 平台覆盖面：虽然 1.6.0rc5 有完整 wheel 矩阵，但该 pin 是全局的，会同时影响 linux x86_64/aarch64 和 win_amd64 的所有安装路径；若 rc5 在某些平台存在隐性问

题，会影响全部用户。

3. 上游修复滞后：一旦 NVIDIA 补发 1.6.0rc6 的完整 wheel，该 pin 会阻止用户自动享受新版本，需人工解除；若无人及时处理，会长期停留在旧版。
4. CI 验证不确定：PR 状态显示 Extra CI 仍为失败状态，合并后主分支 CI 是否全绿需要后续观察；插件式依赖修复最容易在真实安装环境暴露问题。 - 影响：直接影响：修复所有 Python 3.10 x86_64 环境（尤其是 CPU CI shard）安装 sglang 时因 cuda-tile 1.6.0rc6 缺 wheel 导致的 Install dependencies 阶段失败，属于 CI 与用户安装链路的阻断性修复。影响范围：所有通过 PyPI 安装 sglang 的用户和全部 CI 作业都会使用到该 pin，但用户在 Python 3.11-3.14 平台上几乎无感知。对团队而言，避免了 CI 长期全红，并为后续 flashinfer/cuda-tile 升级保留了清晰的待办事项。 - 风险标记：传递依赖钉死需手动升级，上游补发 wheel 后需解除 pin，全局多平台安装链路受影响，合并时 CI Extra 尚未全绿

关联脉络

- 暂无明显关联 PR