

PR #34231 完整报告

sgl-project/sglang

[CI] Keep the torch compilation cache instead of wiping it on install

合并时间: 2026-08-10 16:21

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/34231>

执行摘要

- 一句话: CI 安装不再清空 Torch 编译缓存, 消除并发竞态
- 推荐动作: 该 PR 是一个小而精准的基础设施修复, 值得 CI 相关维护者关注。它体现了“缓存按内容哈希寻址则无需手动清理”这一设计原则, 并利用注释记录决策原因, 对后续维护者友好。建议合并, 并可在未来的 CI 磁盘管理策略中再统一考虑缓存清理策略。

功能与动机

PR body 指出, 在将多个 GPU runner 打包到一台宿主机上的 runner pool 中, 缓存目录是共享挂载, 原先安装步骤中的清空操作会 unlink 并发任务正在编译的文件, 导致 Inductor 报 `OSError: source code not available -> ValueError: @jit functions should be defined in a Python file`。同时说明 Inductor 和 Triton 缓存条目是按内容哈希寻址的, 陈旧条目永远不会被复用, 因此清空操作没有益处。

实现拆解

变更集中在单个 CI 脚本内, 可按以下步骤拆解:

1. 定位清理逻辑: `scripts/ci/cuda/ci_install_dependency.sh` 中的 `clean_site_packages` 函数原先通过一段内联 Python 脚本尝试清空多个缓存位置 (`TORCHINDUCTOR_CACHE_DIR`、`/tmp/torchinductor_<user>`、`~/.cache/sglang/inductor`)。
2. 删除清空代码: 移除该 Python 脚本, 不再对 Torch 编译缓存目录执行 `shutil.rmtree`, 保留函数内对损坏 `dist-info` 目录的清理逻辑 (按 PEP 376 处理缺失 METADATA 的目录)。
3. 补充设计注释: 在原本清理位置添加注释, 明确说明缓存不清空的理由: 缓存按内容哈希寻址、旧条目不会被复用, 以及共享挂载下并发编译文件可能被误删的风险。
4. 无测试配套: 这是纯 CI 基础设施脚本变更, 不涉及源码逻辑或测试文件, 未增加新测试。

关键文件:

- `scripts/ci/cuda/ci_install_dependency.sh` (模块 CI 脚本; 类别 `infra`; 类型 `infrastructure`): 唯一变更文件, 移除了 CI 安装时清空 Torch 编译缓存的逻辑, 并补充了设计注释。

关键符号: `clean_site_packages`

关键源码片段

scripts/ci/cuda/ci_install_dependency.sh

唯一变更文件，移除了 CI 安装时清空 Torch 编译缓存的逻辑，并补充了设计注释。

```
clean_site_packages() {  
    # torch 编译缓存故意不清空：  
    # - 条目按内容哈希寻址，过期条目不会被复用，清空无收益；  
    # - 多 runner 共享宿主机时，该目录是共享挂载，  
    # 清空会 unlink 并发任务正在编译的文件，  
    # 引发 `torch._inductor.exc.InductorError: SubprocException`。  
    # 仍然清理损坏的 dist-info 目录（PEP 376 缺失 METADATA）。  
    SITE_PACKAGES=$(python3 -c "import site; print(site.getsitepackages()[0])")  
    # ... 其余清理逻辑保持不变  
}
```

评论区精华

PR 没有收到人工 review 评论，唯一的 review 来自 [copilot-pull-request-reviewer\[bot\]](#)，自动摘要了变更内容：停止删除共享 Torch 编译缓存，防止并发 GPU 任务之间的竞态，并说明缓存保留的原因。未出现任何反对意见或未解决的疑虑。

- 暂无高价值评论线程

风险与影响

- 风险：主要风险在于移除缓存清空逻辑后，CI 工作节点上的 Torch 编译缓存目录可能随时间累积，占用磁盘空间。但 CI 环境通常是短暂运行的容器或被定期清理，且缓存条目按内容哈希寻址，旧条目不会被复用，因此影响有限。另一个潜在风险是，如果未来缓存目录结构发生变化，旧残留可能导致混淆，但概率极低。整体风险很低。
- 影响：影响范围为所有使用 scripts/ci/cuda/ci_install_dependency.sh 的 CUDA CI 任务，尤其是多个 GPU runner 共享宿主机（如自托管 runner pool）的场景。变更消除了并发编译时缓存文件被误删导致的随机失败，提升了 CI 的稳定性和可重复性。对用户和运行时功能无直接影响，属于纯内部基础设施改进。
- 风险标记：共享缓存并发，磁盘占用增长，CI 脚本无测试

关联脉络

- PR #34186 [CI] Key scheduled CUDA suites by runner_config instead of hand-written jobs: 同属 CI 基础设施演进，涉及 CUDA 测试工作流的统一调度，与本 PR 都在改善 CI 稳定性和可维护性，但未修改同一文件。