PR #21551 完整报告

sgl-project/sglang

[MPS] Fix Triton stub sub-module imports on Python 3.12+

合并时间：2026-04-01 11:26

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/21551

执行摘要

本 PR 修复了在 macOS/MPS 环境中，由于 Python 3.12 移除 find_module() 回退机制，导致 SGLang 的 Triton stub 无法导入 triton.* 子模块（如 triton.compiler.compiler）的关键 bug。通过将 _TritonFinder 类的导入协议更新为 PEP 451 的 find_spec()，确保了在 Python 3.9+ 上的兼容性，解决了 torch 2.9+ 用户启动失败的问题。变更影响范围有限但重要，风险较低，已通过测试验证。

功能与动机

该 PR 旨在修复 issue #21548 中报告的问题：在 macOS 上，SGLang 使用 Triton stub 模拟 triton 模块，以便在没有真实 Triton 的环境下运行。但随 Python 3.12 发布，CPython 移除了 sys.meta_path 中 find_module() 的回退机制（当 finder 未实现 find_spec() 时），导致旧 stub 无法处理点分 triton.* 子模块导入。具体表现为，当 PyTorch 2.9+ 尝试导入 triton.compiler.compiler 时，会引发 ModuleNotFoundError，进而使 bench_one_batch 等功能失败。PR body 中明确指出："On Python 3.12, CPython removed the sys.meta_path fallback to find_module() when a finder does not implement find_spec(), so the old stub no longer materialized those sub-modules."

实现拆解

实现集中在 python/sglang/_triton_stub.py 文件中的 _TritonFinder 类。关键改动如下：

- 移除过时方法：删除了 find_module 和 load_module 方法，这些是 Python 遗留导入协议。
- 新增 find_spec 方法：实现 PEP 451 的 find_spec()，动态处理 triton 及 triton.* 子模块导入。代码逻辑包括：
  - 检查 sys.modules 中是否已存在模块。
  - 若不存在，创建 _MockModule 实例并注册到 sys.modules。
  - 建立父子模块关系，例如当导入 triton.compiler 时，将其设置为 triton 的子模块。
- 移除冗余 mock：删除了代码中显式注册的 triton.compiler 和 triton.compiler.compilermock，因为 find_spec() 已能动态处理所有子模块，简化了代码结构。

评论区精华

review 讨论中，核心交锋围绕更新导入协议和验证修复：

- yeahdongcn在 Issue 评论中提问："I don't see this issue in my environment. It might be related to Python 3.12..." 并建议测试 Python 3.11 和 3.12 兼容性。随后在 review

中强调："Could you please remove find_module/load_module, as described in the PRdescription?"

- karanb192回应确认问题是 Python 3.12 特有，并通过测试矩阵验证修复有效。在后续提交中移除了旧方法，说明："The latest revision removes find_module / load_module entirely and uses only find_spec"。
- 讨论还涉及移除冗余 mock，karanb192 解释："find_spec() on _TritonFinder already handles dotted triton.* sub-module imports dynamically, so the explicit triton.compiler mocks and their old inline comment are no longer needed."

风险与影响

风险分析：

- 回归风险：find_spec 方法需确保正确处理所有子模块导入；代码中通过动态创建 mock 和检查 sys.modules 来规避，但缺少单元测试覆盖具体逻辑。
- 兼容性风险：依赖于 PEP 451 的 find_spec，SGLang 要求 Python 3.9+，find_spec 自 Python 3.4 可用，因此风险可控。移除过时方法未发现副作用。

影响评估：

- 对用户：修复了 macOS 上 Python 3.12+ 用户在使用 torch 2.9+ 时无法运行 SGLang 的问题，提升了系统可用性。
- 对系统：改进导入系统健壮性，遵循现代 Python 标准，降低未来版本升级的兼容性问题。
- 对团队：提供从遗留协议迁移的示例，变更集中，易于维护。

关联脉络

从提供的近期历史 PR 分析中，未发现直接相关的 PR；但 issue #21548 是本 PR 的直接源头。变更属于 macOS/MPS 环境下的基础设施修复，可能与仓库中其他 bugfix 或 infra 类 PR（如 PR 21797 修复 CI 脚本）有间接关联，但无明确跨 PR 脉络。该修复凸显了对 Python 版本演进的适应，强调了在维护兼容性时更新导入协议的重要性。