

PR #34825 完整报告

sgl-project/sglang

[diffusion] Bound overlong weight lock filenames

合并时间: 2026-08-15 17:20

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/34825>

执行摘要

- 一句话: 限制 diffusion 权重锁文件名长度, 避免 ENAMETOOLONG
- 推荐动作: 值得精读。改动虽小, 但展示了如何在保持向后兼容的同时处理文件系统边界问题; get_lock 中通过 os.fsencode 以字节尺度判断文件名长度、只对超限路径回退摘要的写法, 可作为类似命名约束场景的参考。

功能与动机

PR body 指出: diffusion weight loader builds lock filenames from a SHA-256 digest plus the full sanitized model path, 因此 Deep absolute Hugging Face snapshot/component paths can exceed the common 255-byte per-filename limit and fail with ENAMETOOLONG before weight loading begins。

实现拆解

1. 在 python/sglang/multimodal_gen/runtime/loader/weight_utils.py 的 get_lock 中, 保留原有 hash_name + model_name + ".lock" 的命名。
2. 在构造 lock_file_name 后, 用 os.fsencode 检查其字节长度是否超过 255; 超限时回退为 hash_name + ".lock" (64 字节 SHA-256 摘要加后缀, 远低于上限)。
3. 在 python/sglang/multimodal_gen/test/unit/test_weight_utils.py 新增 TestDiffusionWeightLock, 用超长 snapshot 风格路径构造锁, 断言文件名长度 ≤ 255 并实际 with lock 获取锁确认落盘。

关键文件:

- python/sglang/multimodal_gen/runtime/loader/weight_utils.py (模块 权重加载; 类别 source; 类型 core-logic; 符号 get_lock): 修复核心: get_lock 增加超长锁文件名回退逻辑
- python/sglang/multimodal_gen/test/unit/test_weight_utils.py (模块 单元测试; 类别 test; 类型 test-coverage; 符号 TestDiffusionWeightLock, test_long_snapshot_path_uses_bounded_lock_filename): 新增回归测试覆盖长路径锁文件名

关键符号: get_lock, test_long_snapshot_path_uses_bounded_lock_filename

关键源码片段

python/sglang/multimodal_gen/runtime/loader/weight_utils.py

修复核心：get_lock 增加超长锁文件名回退逻辑

```
def get_lock(model_name_or_path: str | Path, cache_dir: str | None = None):
    lock_dir = cache_dir or temp_dir
    model_name_or_path = str(model_name_or_path)
    os.makedirs(os.path.dirname(lock_dir), exist_ok=True)
    model_name = model_name_or_path.replace("/", "-")
    hash_name = hashlib.sha256(model_name.encode()).hexdigest()

    # 摘要加模型名组成锁文件名，可避免与旧用户的历史锁文件冲突
    lock_file_name = hash_name + model_name + ".lock"

    # Linux 文件系统通常限制单个文件名为 255 字节，
    # 绝对快照路径即使整体合法，也可能超过该限制。
    # 摘要本身已具备抗碰撞性，因此超限时只使用摘要作文件名，
    # 同时保留常规路径下的历史文件名不变。
    if len(os.fsencode(lock_file_name)) > 255:
        lock_file_name = hash_name + ".lock"

    # 使用 0o666 权限，保证多个用户可共享同一把锁
    lock = filelock.FileLock(os.path.join(lock_dir, lock_file_name), mode=0o666)
    return lock
```

python/sglang/multimodal_gen/test/unit/test_weight_utils.py

新增回归测试覆盖长路径锁文件名

```
class TestDiffusionWeightLock(unittest.TestCase):
    def test_long_snapshot_path_uses_bounded_lock_filename(self):
        # 模拟 Hugging Face 深快照路径：包含长仓库名、64 字符 commit 与子组件
        component_path = os.path.join(
            "/scratch",
            "models--" + "very-long-repository-name-" * 8,
            "snapshots",
            "a" * 64,
            "transformer",
            "config.json",
        )

        with tempfile.TemporaryDirectory() as lock_dir:
            lock = get_lock(component_path, lock_dir)
            lock_filename = os.path.basename(lock.lock_file)

            # 锁文件名必须落在 255 字节的文件名上限之内
            self.assertLessEqual(len(os.fsencode(lock_filename)), 255)
            # 真正获取锁并确认锁文件落盘，覆盖文件系统层面的可用性
            with lock:
                self.assertTrue(os.path.exists(lock.lock_file))
```

评论区精华

本 PR 没有 review 评论。作者 BBuf 仅触发了 /tag-and-rerun-ci 并附了 CI 链接，未出现设计争议或未解决疑虑。

- 暂无高价值评论线程

风险与影响

- 风险：风险点集中在 get_lock 的文件名回退逻辑：对于超长路径，锁文件名变为纯摘要，若不同路径产生相同 SHA-256 摘要（概率极低）会造成锁冲突；常规路径行为完全不变。另外 os.fsencode 按文件名编码检查长度，与文件系统实际限制一致，但不同文件系统可能有不同限制，此处按常用 255 字节处理。测试仅覆盖了临时目录场景，未覆盖并发获取同一锁的互斥性。
- 影响：影响范围仅限 diffusion 权重加载路径中的锁文件名生成。对常规模型路径无行为变化；对使用深层 Hugging Face 快照路径的用户，修复了权重加载前即失败的 ENAMETOOLONG 问题。改动小，回归风险低。
- 风险标记：文件名长度边界，依赖 SHA-256 抗碰撞性

关联脉络

- 暂无明显关联 PR