PR #46769 完整报告

vllm-project/vllm

[CPU] Fix macOS/Apple Silicon hang by enabling OpenMP in the build

合并时间：2026-06-27 02:32

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/46769

执行摘要

此 PR 修复了 macOS CPU 上因构建未传递 -fopenmp 标志导致的 attention split-KV 死锁问题。核心方案是：通过 cmake 添加 -Xpreprocessor -fopenmp 启用 OpenMP，并将所有 CPU 内核的线程数获取封装为 cpu_utils::get_max_threads()，在无 OpenMP 时安全降级为串行。同时改进了 CI 烟雾测试以覆盖该路径，并调整了 CI 矩阵。变更涉及 10 个文件，已合入 main。

功能与动机

macOS CPU 构建在 #16086 后未传递 -fopenmp，导致 #pragma omp parallel 被编译器忽略，但 omp_get_max_threads() 仍报告全部核心。attention 的 split-KV 路径在 barrier 上等待从未启动的线程团队，从而死锁。短 prompt 可侥幸逃过，但长 prompt 必然触发。此 PR 从根本修复，替代了之前的 workaround #45649（强制 OMP_NUM_THREADS=1，但导致 attention 单线程化）。

实现拆解

1. 添加编译标志：在 cmake/cpu_extension.cmake 中为 Apple Clang 添加 -Xpreprocessor -fopenmp，无需额外链接，_C 通过 dynamic_lookup 从 torch 解析 libomp。
2. 封装线程数获取：在 csrc/cpu/cpu_types.hpp 的 cpu_utils 命名空间中新增 get_max_threads()：有 OpenMP 时返回 omp_get_max_threads()，否则返回 1 并警告。
3. 替换内核调用：在 6 个源文件（cpu_attn_impl.hpp、cpu_fused_moe.cpp、cpu_wna16.cpp、dnnl_kernels.cpp、mla_decode.cpp）中共 7 处将 omp_get_max_threads() 改为 cpu_utils::get_max_threads()。
4. 改进 CI 烟雾测试：.github/workflows/macos-smoke-test.yml 发送约 260 token 的 prompt 以触发 split-KV 路径；CI 矩阵固定为门控 macos-15（Clang 16）和非阻塞 macos-26（Clang 17）；.github/actionlint.yaml 允许预览标签 macos-26。
5. 更新文档：docs/getting_started/installation/cpu.apple.inc.md 补充 OpenMP 依赖说明。

csrc/cpu/cpu_types.hpp

新增 cpu_utils::get_max_threads() 封装函数，这是本次修复的核心抽象，确保非 OpenMP 构建安全降级。

```
#ifndef CPU_TYPES_HPP
#define CPU_TYPES_HPP
```

```
#if defined(__x86_64__)
  #include "cpu_types_x86.hpp"
#elif defined(__aarch64__)
  #include "cpu_types_arm.hpp"
// ... 其他架构分支省略 ...
#else
  #include "cpu_types_scalar.hpp"
#endif

#ifdef _OPENMP
  #include <omp.h>
#endif

#include <c10/util/Exception.h>

namespace cpu_utils {
// 当没有 OpenMP 时，`#pragma omp parallel` 区域会编译为串行循环，
// 如果内核基于线程数做屏障（barrier），就会导致死锁。
// 因此返回 1，并给出一次性警告。
inline int get_max_threads() {
#ifdef _OPENMP
  return omp_get_max_threads();
#else
  TORCH_WARN_ONCE(
      "vLLM CPU was built without OpenMP; running single-threaded.");
  return 1;
#endif
}
} // namespace cpu_utils

#endif
```

评论区精华

> WindChimeRan: "Solution LGTM ... One risk: this PR has only been tested on macOS 26 / clang 21, and no macOS CI runs on this PR. It may break on macos-15 clang 17, and if so, it will break as a loading error, worse than today's hang, and `cpu_utils::get_max_threads()` cannot prevent loading error."
>
> mgoin: "Ran the smoke test on this branch before merge — both legs green, including macos-15 / Apple Clang 16. So the load-error worst case doesn't show up."

风险与影响

- 兼容性风险：-Xpreprocessor -fopenmp 在更早 Apple Clang 上可能失效，但 CI 覆盖了 macos-15（Clang 16）并通过。
- 降级风险：非 OpenMP 构建串行运行但不会死锁。

- 加载风险：libomp 加载失败可能导致 _C 导入错误，但 torch 始终提供 libomp，且加载路径与已有 omp_get_max_threads 一致。
- 正面影响：macOS CPU 用户可处理长 prompt 且 attention 多线程加速；移除了临时 workaround，降低维护成本。

关联脉络

此 PR 直接替代了 #45649（强制单线程的 workaround），并修复了 #16086 引入的回归（添加 omp.h 但未传递 -fopenmp）。相关 issue #15941 讨论了 macOS 上的 OpenMP 兼容性问题。这些改进是 vLLM CPU 后端持续完善的一部分，确保在 Apple Silicon 上的可用性和性能。