

PR #29286 完整报告

sgl-project/sglang

[sgl-kernel/cpu]: exclude amx gemm source from arm build

合并时间: 2026-06-26 08:23

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/29286>

1. 执行摘要

该 PR 是一个小型的跨平台构建梳理, 将 Intel AMX 专用的 `gemm_int8.cpp` 从 ARM 构建中排除, 并添加了对应的 ARM 存根函数。变更简单、风险低, 不影响 x86 功能, 使构建系统更清晰。

2. 功能与动机

PR 标题明确指出了目标: "exclude amx gemm source from arm build"。动机是 ARM 架构没有 AMX 指令集, 这些源码在 ARM 上既无用又可能导致编译问题。通过将源码归类为 "x86-only", 并在 ARM 侧提供存根, 可以确保代码干净编译, 并为将来 ARM 原生 GEMM 实现预留接口。

3. 实现拆解

实现分为三个并行的步骤:

1. 构建配置 (`sgl-kernel/csrc/cpu/CMakeLists.txt`): 将 `gemm_int8.cpp` 追加到 `SGLANG_CPU_X86_ONLY_SOURCES` 变量中。这个变量的内容在 ARM 构建时不会被编译, 从而在构建系统层面直接排除该文件。
2. ARM 存根 (`sgl-kernel/csrc/cpu/aarch64/gemm_int8.cpp`): 在该文件中新增 `per_token_quant_int8_cpu` 和 `int8_scaled_mm_cpu` 两个函数。它们不执行任何实际计算, 直接抛出 `TORCH_CHECK(false, "not implemented yet")` 异常。它们的存在仅仅是为了满足链接器的符号解析需求——ARM 代码中其他模块可能引用了这些符号。
3. 移除条件编译守卫 (`sgl-kernel/csrc/cpu/gemm_int8.cpp`): 删除了包围 `int8_scaled_mm_with_quant` 函数定义的 `#ifndef __aarch64__` 和 `#endif`。现在该函数在所有架构下都参与编译。在 ARM 上, 它被编译后仍然会调用上一步的存根, 从而在运行时抛出异常, 但这比完全无法编译要好, 因为运行时错误更容易定位和调试。

`sgl-kernel/csrc/cpu/aarch64/gemm_int8.cpp`

为 ARM 平台添加了两个存根函数, 确保链接器可以解析符号。

```
// sgl-kernel/csrc/cpu/aarch64/gemm_int8.cpp (after PR)
```

```
// 在 anonymous namespace 的 int8_scaled_mm_impl 模板实例化之后
```

```
// 新增以下两个存根函数, 用于在 ARM 构建中提供符号定义
```

```
// 存根函数, 仅用于满足编译器链接需求, 实际调用会抛出异常
```

```
std::tuple<at::Tensor, at::Tensor> per_token_quant_int8_cpu(at::Tensor& /*A*/) {
    TORCH_CHECK(false, "not implemented yet");
    return {at::Tensor(), at::Tensor()};
}
```

// 存根函数，所有参数均注释掉以避免 -Wunused-parameter 警告

```
at::Tensor int8_scaled_mm_cpu(
    at::Tensor& /*mat1*/,
    at::Tensor& /*mat2*/,
    at::Tensor& /*scales1*/,
    at::Tensor& /*scales2*/,
    const std::optional<at::Tensor>& /*bias*/,
    at::ScalarType /*out_dtype*/,
    bool /*is_vnni*/) {
    TORCH_CHECK(false, "not implemented yet");
    return at::Tensor();
}
```

// 原有的 int8_scaled_mm_with_quant 函数（未改动）

// 该函数在 ARM 上也会编译，但内部会调用上方的存根，运行时异常

5. 评论区精华

- gemini-code-assist[bot]建议将存根参数名注释掉以避免编译器警告。
- cyb70289回复：“Dummy functions, just to satisfy compiler. Will implement if to be used.” 并采纳了参数注释的建议。

讨论简洁高效，核心人员对存根设计意图达成一致。

6. 风险与影响

- 风险：极低。x86 功能完全不变；ARM 上之前无法编译的代码现在可编译，但运行时会抛出异常。如果任何现有 ARM 用户依赖 int8_scaled_mm_with_quant 的编译版本（过去不可能），则可能遇到运行时错误。但基于历史，这是不可能的。
- 影响：正面影响。构建系统更清晰，为未来 ARM 原生 GEMM 实现铺平了道路。

7. 关联脉络

此 PR 是 sgl-kernel 跨平台构建系列的一部分。同仓库最近的 PR [28967] 和 [29103] 也涉及 AMD 和 ARM 平台支持，但本 PR 专注于 ARM 构建的基础设施改进。目前没有发现与本 PR 直接功能关联的其他 PR。