

PR #49387 完整报告

vllm-project/vllm

Add `sm\_107` for Rubin

合并时间: 2026-07-25 02:59

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/49387>

执行摘要

- 一句话: 为 Rubin GPU 添加 SM107 构建支持
- 推荐动作: 值得精读, 尤其是 CMake 函数调优的方法。关注架构匹配优先级的调整 and 外部项目架构列表的维护方式, 这为后续新硬件支持提供了可复用的模式。

功能与动机

需要支持 NVIDIA Vera Rubin (SM107) GPU 的编译。CUDA 13.4 引入了原生 SM107 目标, vLLM 应能针对该架构构建并充分利用与 SM100 的内核兼容性。同时修复了架构选择中精确匹配被家族回退掩盖的问题。

实现拆解

1. 修改 cmake/utils.cmake: 在 `cuda_archs_loose_intersection` 函数中将精确匹配检查提前至家族回退之前, 防止 10.0f 误匹配 10.7 目标。并将 10.7f 添加到 `cuda_archs_sm90plus` 的架构列表中。
2. 修改 CMakeLists.txt: 当 CUDA 版本 ≥ 13.4 时, 设置 `CUDA_SUPPORTED_ARCHS` 包含 10.7; 在多个 kernel 编译目标中加入 10.7f 架构 (如 `COOPERATIVE_TOPK_ARCHS`、`DSV3_FUSED_A_GEMM_ARCHS`、`SCALED_MM_ARCHS`、`CUTLASS_MOE_DATA_ARCHS` 等)。
3. 更新外部项目 CMake 文件: 在 `flashmla.cmake`、`deepgemm.cmake`、`cutlass.cmake` 中根据 CUDA 版本条件加入 10.7f 架构, 确保这些库也为 Rubin 编译。
4. 添加测试 `tests/test_cmake_utils.py`: 新增 `test_exact_family_arch_precedes_generic_family_fallback` 测试, 验证架构选择逻辑的修正正确。
5. 更新注释: 在 CMakeLists.txt 中为 Rubin 和 SM107 添加说明。

关键文件:

- `tests/test_cmake_utils.py` (模块 构建测试; 类别 test; 类型 test-coverage; 符号 `test_exact_family_arch_precedes_generic_family_fallback`): 新增测试验证架构精确匹配优先于家族回退的修复, 确保 `cuda_archs_loose_intersection` 函数正确性。
- `cmake/utils.cmake` (模块 构建系统; 类别 infra; 类型 core-logic; 符号 `cuda_archs_loose_intersection`, `cuda_archs_sm90plus`): 核心逻辑变更: 修改 `cuda_archs_loose_intersection` 函数, 将精确匹配检查提前至家族回退之前, 修复因 10.0f 覆盖 10.7 的问题。

- `cmake/external_projects/qutlass.cmake` (模块 QuTLASS 集成; 类别 `infra`; 类型 `configuration`) : 根据 CUDA 版本条件加入 10.7f 架构, 确保 QuTLASS 为 Rubin 编译。
- `cmake/external_projects/flashmla.cmake` (模块 FlashMLA 集成; 类别 `infra`; 类型 `configuration`) : 根据 CUDA 版本条件加入 10.7f 架构, 确保 FlashMLA 为 Rubin 编译。
- `cmake/external_projects/deepgemm.cmake` (模块 DeepGEMM 集成; 类别 `infra`; 类型 `configuration`) : 根据 CUDA 版本条件加入 10.7f 架构, 确保 DeepGEMM 为 Rubin 编译。
- `CMakeLists.txt` (模块 主构建文件; 类别 `docs`; 类型 `documentation`) : 设置 `CUDA_SUPPORTED_ARCHS` 包含 10.7, 并在多个 `kernel` 编译目标中加入 10.7f, 同时添加注释文档。

关键符号: `cuda_archs_loose_intersection`, `cuda_archs_sm90plus`,
`test_exact_family_arch_precedes_generic_family_fallback`

关键源码片段

`tests/test_cmake_utils.py`

新增测试验证架构精确匹配优先于家族回退的修复, 确保 `cuda_archs_loose_intersection` 函数正确性。

```
# SPDX-License-Identifier: Apache-2.0
# SPDX-FileCopyrightText: Copyright contributors to the vLLM project
```

```
import subprocess
from pathlib import Path
```

```
def test_exact_family_arch_precedes_generic_family_fallback(tmp_path: Path):
    repo_root = Path(__file__).parents[1]
    script = tmp_path / "test_cuda_archs.cmake"
    # 写入 CMake 脚本, 调用 cuda_archs_loose_intersection
    # 传入源列表 "10.0f;10.7f" 和目标 "10.7"
    # 预期结果是 "10.7f" (精确匹配) 而非 "10.0f" (家族回退)
    script.write_text(
        f"""
cmake_minimum_required(VERSION 3.26)
include("{repo_root} / "cmake" / "utils.cmake")
cuda_archs_loose_intersection(
    actual "10.0f;10.7f" "10.7")
if(NOT "${{actual}}" STREQUAL "10.7f")
    message(FATAL_ERROR "Expected 10.7f, got '${{actual}}')
endif()
"""
    )

    subprocess.run(["cmake", "-P", script], check=True)
```

`cmake/utils.cmake`

核心逻辑变更：修改 `cuda_archs_loose_intersection` 函数，将精确匹配检查提前至家族回退之前，修复因 10.0f 覆盖 10.7 的问题。

```
function(cuda_archs_loose_intersection OUT_CUDA_ARCHS SRC_CUDA_ARCHS TGT_CUDA_ARCHS)
# ... 其他代码 ...
set(_CUDA_ARCHS)
# 第一步：优先处理精确的架构名匹配（如 10.7f）
# 在家族回退之前，先尝试精确匹配
foreach(_arch ${_SRC_CUDA_ARCHS})
    if(_arch MATCHES "[af]$")
        string(REGEX REPLACE "[af]$" "" _base "${_arch}")
        if("${_base}" IN_LIST _TGT_CUDA_ARCHS)
            list(REMOVE_ITEM _SRC_CUDA_ARCHS "${_arch}")
            list(REMOVE_ITEM _TGT_CUDA_ARCHS "${_base}")
            list(APPEND _CUDA_ARCHS "${_arch}")
        endif()
    endif()
endforeach()

# 第二步：然后处理家族级别匹配（如 10.0f 匹配 10.x 家族）
foreach(_arch ${_SRC_CUDA_ARCHS})
    if(_arch MATCHES "[af]$")
        list(REMOVE_ITEM _SRC_CUDA_ARCHS "${_arch}")
        string(REGEX REPLACE "[af]$" "" _base "${_arch}")
        if("${_base}a" IN_LIST _TGT_CUDA_ARCHS)
            list(REMOVE_ITEM _TGT_CUDA_ARCHS "${_base}a")
            list(APPEND _CUDA_ARCHS "${_base}a")
        elseif("${_base}f" IN_LIST _TGT_CUDA_ARCHS)
            list(REMOVE_ITEM _TGT_CUDA_ARCHS "${_base}f")
            list(APPEND _CUDA_ARCHS "${_base}f")
        endif()
    endif()
endforeach()
# ... 其余代码 ...
endfunction()

function(cuda_archs_sm90plus OUT_CUDA_ARCHS TGT_CUDA_ARCHS)
    if(${CMAKE_CUDA_COMPILER_VERSION} VERSION_GREATER_EQUAL 13.0)
        # 添加 10.7f 以支持 Rubin
        cuda_archs_loose_intersection(_archs "9.0a;10.0f;10.7f;11.0f;12.0f" "${TGT_CUDA_ARCHS}")
    else()
        cuda_archs_loose_intersection(_archs "9.0a;10.0a;10.1a;10.3a;12.0a;12.1a" "${TGT_CUDA_ARCHS}")
    endif()
endfunction()
```

评论区精华

审阅者 mgoin 对 qutlass.cmake 的修改评论表达了对运行时平台检查是否需要更新的担忧 ("I hope I don't need to update how current_platform.is_device_capability_family(100) works")，但最终认为无需修改并批准了 PR。他还称赞了新增的 CMake 测试 ("We should be doing a lot more cmake testing, I like this")。

- 运行时平台检查是否需更新 (question): 最终审阅者确认无需修改，PR 被批准
- 新增 CMake 测试 (testing): 无异议，测试通过

风险与影响

- 风险：主要风险在于 Rubin 架构可能与 SM100 内核不完全兼容，需要验证共享内核的行为。CMake 逻辑变更可能意外影响其他架构的选择。通过条件化版本检查 (≥ 13.4) 限制范围，降低了影响。新增测试覆盖了核心匹配逻辑，但运行时兼容性仍需后续验证。
- 影响：用户可在 Rubin GPU 上无缝使用 vLLM，无需手动指定架构。开发者需在添加新架构时参考本 PR 的模式。影响范围仅限于构建系统，运行时逻辑未变。对系统性能无直接影响，但可能增加构建时间（新增一个目标）。
- 风险标记：构建脚本变更，硬件兼容性风险

关联脉络

- 暂无明显关联 PR