

PR #56599 完整报告

vllm-project/vllm

[CI] Update DeepSeek V4.1 MegaMoE routing test

合并时间: 2026-09-13 02:31

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/56599>

执行摘要

- 一句话: 对齐 pad-free 图像 token, 修复 MegaMoE 测试收集失败
- 推荐动作: 不需要精读, 属于可直接略过的 CI 修复类 PR; 但有两处值得作为经验留存:
 1. tests/models/test_deepseek_v4_mega_moe.py 内联复刻 image_mask 判定的写法是典型的“镜像实现”, 建议后续把“图像位置判定”抽成 vllm/models/deepseek_v4_1/common/mm_preprocess.py 侧的可复用 helper (如 is_image_position(input_ids)), 让测试调用生产函数而不是复制逻辑, 从根上避免本类常量漂移。
 2. 遇到“上游删常量导致测试 collect 失败”的同类问题时, 可参考本 PR 的处理顺序: 先确认新契约 (pad-free 图像流)、再保持夹具语义等价 (两个哨兵位仍代表 image span)、最后用硬件 CI 验证而非本地 skip 结果作为合并依据。

功能与动机

PR body 直接给出根因: “PR #56554 removed `IMAGE_PAD_ID`, but `tests/models/test_deepseek_v4_mega_moe.py` still imported and used it. B200 DeepSeek V4 Kernels now fails during collection before this test can run.” 也就是说, 这不是功能缺陷, 而是测试夹具与上游常量契约脱节造成的 CI 收集期阻塞: 测试连 import 都过不去, MegaMoE 的 VL router bias 回归覆盖实际处于失效状态。作者同时说明新夹具 “represents image-span positions with the checkpoint's single `IMAGE_SENTINEL_BASE_ID`, matching the current preprocessing contract while preserving the test's router-bias coverage”, 并做了 duplicate check (搜索 #56554、`IMAGE_PAD_ID` 与该测试, 确认没有其他在飞 PR 处理该失败), 同时把与 PR #56594 重叠的 CuTeDSL bounds-test 修复从本 PR 中剥离。

实现拆解

本 PR 全部改动集中在 `tests/models/test_deepseek_v4_mega_moe.py` (+3/-6), 不涉及任何生产代码、配置或 kernel, 属于典型的“上游契约变更后同步测试夹具”的维护动作。

变更入口: 失败点在模块级 import。上游 #56554 从 `vllm/models/deepseek_v4_1/common/mm_preprocess.py` 移除 `IMAGE_PAD_ID` 后, 测试文件顶层的 `from vllm.models.deepseek_v4_1.common.mm_preprocess import (IMAGE_PAD_ID, IMAGE_SENTINEL_BASE_ID)` 直接抛 ImportError, pytest 在 collect 阶段即中止, `test_deepseek_v41_moe_routes_without_hash_table` 根本没有机会执行。

- 1. 收窄导入契约：把多行导入改写为单行 `from vllm.models.deepseek_v4_1.common.mm_p`
`reprocess import IMAGE_SENTINEL_BASE_ID`，让测试只依赖当前仍存在的常量，恢复
`collect`。
- 2. 对齐夹具 token 流：在 `test_deepseek_v41_moe_routes_without_hash_table` 的
`vision=True` 分支里，`input_ids` 由 `[42, IMAGE_SENTINEL_BASE_ID, IMAGE_PAD_ID,`
`129257]` 改为 `[42, IMAGE_SENTINEL_BASE_ID, IMAGE_SENTINEL_BASE_ID,`
`129257]`。序列长度仍为 4，与 `hidden_states = torch.randn(4, config.hidden_size)` 以
及 `scheduler_config.max_num_batched_tokens=4` 保持一致；两个相邻图像位用于模拟
`image span`，而不是单个孤立图像 token。
- 3. 对齐期望值计算：`image_mask` 从 `(input_ids == IMAGE_SENTINEL_BASE_ID) |`
`(input_ids == IMAGE_PAD_ID)` 简化为 `input_ids == IMAGE_SENTINEL_BASE_ID`。由
于两处图像位现在同为哨兵值，`mask` 结果（第 1、2 位为 `True`）与旧版完全等价，
`torch.where(image_mask[:, None], moe.gate.bias_vl, bias)` 仍能把 VL 偏置叠加到图像
位置，`expected_ids / expected_weights` 的重算路径不受影响。
- 4. 覆盖度维持：`moe.gate.tid2eid` is `None`、`moe.experts.top_k == top_k`、
`DeepseekV4MegaMoEExperts.w13_weight` shape、`check_routing` 中 `activation_clamp`
`== config.swiglu_limit` 以及 `moe(hidden_states, input_ids) == 3 * hidden_states` 等断
言全部保留，说明本 PR 只修夹具不降断言强度。
- 5. 验证与配套：PR body 记录 `pytest tests/models/test_deepseek_v4_mega_moe.py -q` 可
干净收集（本地无 CUDA，28 个用例 `skip`）、`pre-commit run --files`
`tests/models/test_deepseek_v4_mega_moe.py` 通过；作者明确注明“Exact B200
hardware CI remains required”，最终由 `Isotr0py` 触发 `/ci run`（Buildkite CI #88541）
完成硬件验证。

修改点	旧写法	新写法	影响
import	同时导入 <code>IMAGE_PA</code> <code>D_ID</code> 与 <code>IMAGE_SE</code> <code>NTINEL_BASE_ID</code>	仅导入 <code>IMAGE_SENT</code> <code>INEL_BASE_ID</code>	解除 <code>collect</code> 期 <code>ImportError</code>
<code>input_id</code> <code>s</code>	哨兵位 + 填充位各一个	两个位置都用哨兵值	契合 <code>pad-free</code> 图像流，长度与 <code>shape</code> 不变
<code>image_</code> <code>mask</code>	两个常量的或	单条件比较	VL <code>bias</code> 命中位置与旧版一致

关键文件：

- `tests/models/test_deepseek_v4_mega_moe.py`（模块 测试用例；类别 `test`；类型
`test-coverage`；符号 `test_deepseek_v41_moe_routes_without_hash_table,`
`v41_moe_config`）：本 PR 唯一改动文件。修复因上游 #56554 删除 `IMAGE_PAD_ID` 而
导致的模块级 `import` 失败，并把夹具图像 token 流与 `image_mask` 计算对齐到 `pad-free`
语义，是 B200 DeepSeek V4 Kernels CI 解阻的关键。

关键符号: test_deepseek_v41_moe_routes_without_hash_table, v41_moe_config, check_routing

关键源码片段

tests/models/test_deepseek_v4_mega_moe.py

本 PR 唯一改动文件。修复因上游 #56554 删除 IMAGE_PAD_ID 而导致的模块级 import 失败, 并把夹具图像 token 流与 image_mask 计算对齐到 pad-free 语义, 是 B200 DeepSeek V4 Kernels CI 解阻的关键。

```
# tests/models/test_deepseek_v4_mega_moe.py (本 PR 唯一改动文件)
# 上游 PR #56554 删除 IMAGE_PAD_ID 后, 模块顶层 import 直接 ImportError,
# 导致 B200 上该用例在 pytest collect 阶段就失败。
from vllm.models.deepseek_v4_1.common.mm_preprocess import IMAGE_SENTINEL_BASE_ID

@pytest.mark.parametrize("vision", [False, True])
@pytest.mark.parametrize("layer_id,num_experts,top_k", [(0, 8, 2), (2, 4, 3)])
def test_deepseek_v41_moe_routes_without_hash_table(
    v41_moe_config, monkeypatch, vision, layer_id, num_experts, top_k
):
    """Main and draft layers select experts by score and preserve image routing."""
    config = v41_moe_config.model_config.hf_config
    config.vision_n_layers = int(vision) # 打开视觉层, gate 才会启用 bias_vl 分支
    with torch.device("cuda"):
        moe = DeepseekV41MoE(v41_moe_config, prefix=f"model.layers.{layer_id}.ffn")
        hidden_states = torch.randn(4, config.hidden_size)
        # 4 个 token 位 = 文本 + 2 个图像位 + 文本, 长度与 hidden_states、
        # scheduler_config.max_num_batched_tokens=4 对齐。
        # 旧版第二个图像位写作 IMAGE_PAD_ID; pad-free 图像流下两处都用哨兵基址。
        input_ids = (
            torch.tensor([42, IMAGE_SENTINEL_BASE_ID, IMAGE_SENTINEL_BASE_ID, 129257])
            if vision
            else None
        )
    assert moe.gate.tid2eid is None # 无 hash 表, 完全依赖打分选专家
    assert moe.gate.weight.shape == (num_experts, config.hidden_size)
    assert isinstance(moe.experts, DeepseekV4MegaMoEExperts)
    assert moe.experts.w13_weight.shape == (num_experts, 256, 64)
    assert moe.experts.top_k == top_k
    assert (config.n_routed_experts, config.num_experts_per_tok) == (8, 2)

    with torch.no_grad():
        moe.gate.weight.normal_(std=0.01)
        moe.gate.e_score_correction_bias.copy_(torch.arange(num_experts, device="cuda"))
    if vision:
        # 让 VL 偏置与主偏置取值反向, 便于断言图像位置确实走了 bias_vl
        moe.gate.bias_vl.copy_(-moe.gate.e_score_correction_bias)
```

```

# 在测试内镜像生产逻辑，独立复算期望的 top-k 专家与权重
scores = torch.nn.functional.softplus(
    torch.nn.functional.linear(hidden_states, moe.gate.weight)
).sqrt()
bias = moe.gate.e_score_correction_bias
if vision:
    # 旧写法是 (input_ids == IMAGE_SENTINEL_BASE_ID) | (input_ids == IMAGE_PAD_ID);
    # 常量合并后只需单条件比较，掩码命中位置（第 1、2 位）保持不变
    image_mask = input_ids == IMAGE_SENTINEL_BASE_ID
    bias = torch.where(image_mask[:, None], moe.gate.bias_vl, bias)
expected_ids = (scores + bias).topk(top_k, dim=-1).indices
expected_weights = scores.gather(1, expected_ids)
expected_weights *= config.routed_scaling_factor / expected_weights.sum(
    dim=-1, keepdim=True
)

def check_routing(x, weights, ids, *, activation_clamp):
    # 用 monkeypatch 截获 experts.forward，直接比对路由结果与期望值
    torch.testing.assert_close(ids, expected_ids)
    torch.testing.assert_close(weights, expected_weights)
    assert activation_clamp == config.swiglu_limit
    return x.clone()

monkeypatch.setattr(moe.experts, "forward", check_routing)
monkeypatch.setattr(moe.shared_experts, "forward", lambda x: 2 * x)
# 共享专家贡献 2x，路由专家贡献 1x，合计 3x
torch.testing.assert_close(moe(hidden_states, input_ids), 3 * hidden_states)

```

评论区精华

本 PR 的 review 讨论非常轻量，没有 review comment (`review_comments_count = 0`)，主要信号来自 PR body 的自证和 CI 触发记录：

- Isotr0py 以空 body 直接 APPROVED，随后在评论中发出 /ci run，github-actions 机器人回复 “🔔 Triggered Buildkite CI #88541 for commit 2cae8b5620b6”，说明评审焦点是“夹具改动是否只影响测试 + 硬件 CI 是否恢复”，而非实现细节。
- claude[bot] 留下的是流程性说明：“This pull request is from a fork — automated review is disabled.”，即自动评审对 fork PR 不生效，人工 reviewer 需要承担全部把关。
- 作者在 body 中主动做了 duplicate check 与范围裁剪：“PR #56594 separately fixes the CuTeDSL bounds-test failure from PR #53566, so that overlapping change was removed from this PR.”，这是本 PR 唯一值得记录的协作决策——避免与并行修复重叠导致冲突。
- 作者还披露了 AI 参与程度：“OpenAI Codex assisted with failure analysis and preparing this change.”，并强调“A human maintainer must review every changed line and validate the hardware CI”。
- 硬件 CI 作为唯一有效验证：本地无 CUDA 只能 skip (testing)：以 B200 硬件流水线作为合并前的唯一真实验证，本地仅验证 collect 与 pre-commit，符合纯测试改动的验证预期。

- 与 PR #56594 的范围切分，避免重叠修复冲突 (design): 本 PR 收敛为单一职责——只修 MegaMoE 夹具；CuTeDSL bounds-test 交给 #56594，避免同一 CI 失败被两个 PR 重复修复。
- fork PR 自动评审被禁用，人工把关责任加重 (other): 未触发增量自动评审，最终由 Isotr0py 人工 approve；对这类 3 行测试修复而言风险可控，但流程上缺少第二双眼睛。

风险与影响

- 风险：
 1. 测试语义区分度下降（低危，测试内）：把 IMAGE_PAD_ID 全部替换为 IMAGE_SENTINEL_BASE_ID 后，用例不再覆盖“哨兵 ID 之外的填充区域”这一分支语义。当前上游已确认是 pad-free 图像流，因此是正确收敛；但若未来重新引入独立的 pad 常量，本测试会静默失去对该位置的断言。
 2. 镜像实现漂移（中低危）：image_mask 是在测试内部复刻的生产判定逻辑，而不是调用 vllm/models/deepseek_v4_1/nvidia/model.py 中的真实 gate 代码。一旦生产侧改用其他方式（例如通过 config.vision_n_layers 或 position metadata）标记图像位，测试会继续用陈旧规则自证通过。test_deepseek_v4_mega_moe.py 与生产代码的常量契约是紧耦合的。
 3. 验证依赖硬件 CI（中危，流程性）：本地环境无 CUDA，PR body 自述“28 skipped locally without CUDA”，所有真实路由断言只在 B200 上执行；因此该修复的有效性完全押在 Buildkite #88541 上，任何硬件队列波动都会延长验证周期。
 4. 回归风险极低：改动仅在测试文件，无 source/config/schema 变更，对推理路径、吞吐、精度零影响，不存在用户侧回归面。
- 影响：影响范围限定在 CI 与测试可信度：
- 对系统：DeepSeek V4.1 MegaMoE 的 router bias 回归覆盖从“collect 即失败”恢复为可执行，B200 DeepSeek V4 Kernels 流水线解阻；运行时推理逻辑、KV cache、注意力内核均不受影响。
- 对团队：修复采用 3 行级别的极小改动，review 成本低（1 个 reviewer 一次 approve），但再次暴露“上游常量删除 → 测试顶层 import 崩溃”这一 vLLM 中高频的耦合模式，提示跨模块常量的消费者需要集中登记。
- 对用户：无直接可见影响，既不改 API 也不改模型行为；间接价值是保持 DSv4.1 混合图像 / 文本路由相关回归网有效，降低后续 PR 引入路由错误而不被发现的概率。
- 风险标记：仅测试改动，依赖硬件 CI 验证，测试内镜像实现易漂移，上游常量强耦合

关联脉络

- PR #56554（标题未在上下文中提供）移除 DeepSeek V4.1 的 IMAGE_PAD_ID，引入 pad-free 图像 token 流：本 PR 的直接上游动因。该 PR 从 vllm/models/deepseek_v4_1/common/mm_preprocess.py 删除 IMAGE_PAD_ID，使本测试的顶层 import 失效，本 PR 即为其配套的测试对齐修复。
- PR #56594（标题未在上下文中提供）修复 PR #53566 引入的 CuTeDSL bounds-test 失败：PR body 明确说明该 PR 独立修复 CuTeDSL bounds-test 失败，因此相关改动已从本

PR 中移除，两者是同一次 CI 故障下的并行修复，范围上互补而非重叠。

- PR #53566 [5/N][warmup][DSv4] Migrate NVIDIA CuTeDSL attention kernels: 关联 issue，也是 CuTeDSL bounds 测试失败的来源；它与本 PR 同属 DeepSeek V4.1 NVIDIA 内核与测试基础设施的维护序列，说明本 PR 是 DSv4 内核迁移后遗症清理链条的一环。