PR #52881 完整报告

vllm-project/vllm

[BugFix] Revert incorrect MM keep_on_cpu=True changes

合并时间：2026-08-19 15:07

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/52881

执行摘要

- 一句话：回滚错误的 MM keep_on_cpu=True 变更，修复 CI 回归
- 推荐动作：该 PR 是紧急修复，改动直接，建议快速合入以稳定 CI。值得关注的是回滚时对 non_blocking=True 的补充，这是一个性能折中策略，可以借鉴。但需在合入前确认相关模型的测试覆盖。

功能与动机

PR #52827 在修改 vllm launch 参数校验时，错误地给多个多模态模型的字段配置添加了 keep_on_cpu=True，导致 CI 回归。该 PR 的目的是回滚这些不正确的变更，恢复原有行为，保证 CI 稳定。PR body 明确说明 "Fix CI regression introduced by #52827"。

实现拆解

回滚操作分两步：

1. 对 MultiModalFieldConfig 中 keep_on_cpu=True 参数进行还原，涉及文件：vllm/model_executor/models/minicpmv4_6.py（tgt_sizes、video_tgt_sizes）、llava_onevision2.py（patch_positions、patch_positions_videos）、fireredlid.py（speech_lengths）、qwen2_audio.py（feature_attention_mask）。
2. 对部分张量 .to() 调用添加 non_blocking=True 以补偿移除 keep_on_cpu 带来的潜在同步开销，涉及文件：ernie45_vl.py 的 encoder_eager_forward 和 postprocess_encoder_output，以及 isaac.py 的 _process_image_input。
3. 删除 moss_transcribe_diarize.py 中错误的 keep_on_cpu=True 配置。

关键文件：

- vllm/model_executor/models/ernie45_vl.py（模块 模型执行器；类别 source；类型 data-contract；符号 encoder_eager_forward, postprocess_encoder_output）：修改了 eager forward 和 postprocess 中的张量传输方式，添加 non_blocking=True，属于需要验证的性能补偿改动。
- vllm/model_executor/models/isaac.py（模块 模型执行器；类别 source；类型 data-contract；符号 _process_image_input）：修改了 spatial_grids 的传输方式，同样添加 non_blocking=True 作为性能补偿。

- vllm/model_executor/models/minicpmv4_6.py（模块 模型执行器；类别 source；类型 data-contract；符号 _minicpmv4_6_field_config）：回滚了 tgt_sizes 和 video_tgt_sizes 字段的 keep_on_cpu=True 设置，是回滚的核心文件之一。
- vllm/model_executor/models/llava_onevision2.py（模块 模型执行器；类别 source；类型 data-contract；符号 _field_config）：回滚了 patch_positions 和 patch_positions_videos 字段的 keep_on_cpu=True 设置。
- vllm/model_executor/models/fireredlid.py（模块 模型执行器；类别 source；类型 data-contract；符号 _get_mm_fields_config）：回滚了 speech_lengths 字段的 keep_on_cpu=True 设置。
- vllm/model_executor/models/qwen2_audio.py（模块 模型执行器；类别 source；类型 data-contract；符号 _qwen2audio_field_config）：回滚了 feature_attention_mask 字段的 keep_on_cpu=True 设置，影响 Qwen2-Audio 模型。
- vllm/model_executor/models/moss_transcribe_diarize.py（模块 模型执行器；类别 source；类型 data-contract）：删除了某处错误的 keep_on_cpu=True 配置（具体字段未展示）。

关键符号：encoder_eager_forward, postprocess_encoder_output, _process_image_input, _minicpmv4_6_field_config, _field_config, _get_mm_fields_config, _qwen2audio_field_config

关键源码片段

vllm/model_executor/models/ernie45_vl.py

修改了 eager forward 和 postprocess 中的张量传输方式，添加 non_blocking=True，属于需要验证的性能补偿改动。

```
# vllm/model_executor/models/ernie45_vl.py ( 部分 )
def encoder_eager_forward(
    self, mm_kwargs: dict[str, Any], path: str = "default"
) -> torch.Tensor:
    # 异步拷贝 grid_thw，避免阻塞，提升 eager 路径性能。
    pixel_values = mm_kwargs["pixel_values"].type(self.vision_model.dtype)
    grid_thw = mm_kwargs["image_grid_thw"].to(
        self.vision_model.device, non_blocking=True
    )
    image_features = self.vision_model(pixel_values, grid_thw)
    return self.resampler_model(image_features, grid_thw)

def postprocess_encoder_output(
    self,
    outputs: dict[str, torch.Tensor],
    indices: list[int],
    per_item_out_tokens: list[int],
    dest,
    clone: bool = False,
    batch_mm_kwargs: dict[str, Any] | None = None,
```

```
) -> None:
    # 利用 CPU 上的 grid_thw 直接计算有效 token 数，再异步拷贝到 GPU。
    output = outputs["default"]
    grid_thw_cpu = batch_mm_kwargs["image_grid_thw"]
    grid_thw = grid_thw_cpu.to(output.device, non_blocking=True)
    num_valid = int(
        (grid_thw_cpu[:, 0] * grid_thw_cpu[:, 1] * grid_thw_cpu[:, 2]).sum()
    )
    image_embeds = self.resampler_model(output[:num_valid], grid_thw)
    scatter_output_slices(image_embeds, indices, per_item_out_tokens, dest, clone)
```

vllm/model_executor/models/isaac.py

修改了 spatial_grids 的传输方式，同样添加 non_blocking=True 作为性能补偿。

```
# vllm/model_executor/models/isaac.py ( 部分 )
def _process_image_input(
    self,
    image_input: IsaacImagePixelInputs,
) -> tuple[torch.Tensor, ...]:
    pixel_values = image_input["pixel_values"]
    image_grid_thw = image_input["image_grid_thw"]
    if pixel_values.numel() == 0:
        return ()
    device = next(self.language_model.parameters()).device
    dtype = self.vision_embedding.linear_fc1.weight.dtype
    pixel_values = pixel_values.to(device=device, dtype=dtype)
    # 非阻塞传输，减少 H2D 同步开销。
    spatial_grids = image_grid_thw[:, 1:3].to(
        device, dtype=torch.int32, non_blocking=True
    )
    vision_embeddings = self.vision_embedding((pixel_values, spatial_grids))
    merge_size = self.config.vision_config.pixel_shuffle_scale_factor
    sizes = spatial_grids.prod(-1) // (merge_size * merge_size)
    return tuple(vision_embeddings.split(sizes.tolist()))
```

vllm/model_executor/models/minicpmv4_6.py

回滚了 tgt_sizes 和 video_tgt_sizes 字段的 keep_on_cpu=True 设置，是回滚的核心文件之一。

```
# vllm/model_executor/models/minicpmv4_6.py ( 部分 )
def _minicpmv4_6_field_config(hf_inputs: Mapping[str, torch.Tensor]):
    fields = dict(
        pixel_values=MultiModalFieldConfig.batched("image"),
        tgt_sizes=MultiModalFieldConfig.batched("image"), # 回滚：不再 keep_on_cpu
        image_embeds=MultiModalFieldConfig.batched("image"),
        video_pixel_values=MultiModalFieldConfig.batched("video"),
        video_image_sizes=MultiModalFieldConfig.batched("video", keep_on_cpu=True),
        video_tgt_sizes=MultiModalFieldConfig.batched("video"), # 回滚：不再 keep_on_cpu
        video_embeds=MultiModalFieldConfig.batched("video"),
```

```
    )
    if "use_vit_merger" in hf_inputs:
        fields["use_vit_merger"] = MultiModalFieldConfig.batched(
            "image", keep_on_cpu=True
        )
    return fields
```

评论区精华

无实质性 review 讨论，仅有一个来自 claude[bot] 的自动提示（由于来自 fork 而禁用自动审查）和 AndreasKaratzas 的批准。没有内容需要提炼。

- 暂无高价值评论线程

风险与影响

- 风险：回滚本身风险较低，但需要注意：
  1. keep_on_cpu 的移除意味着这些字段将保留在 GPU 上，可能增加显存占用或改变数据流路径，对于 MiniCPM-V 和 Qwen2-Audio 等模型可能影响性能，需要回归测试验证。
  2. 添加 non_blocking=True 的改动（ernie45_vl.py、isaac.py）属于新增改动，需要确保相关张量在异步传输时没有被提前修改，否则可能引入并发问题。
  3. 回滚未包含测试变更，可能缺乏对回滚后行为的自动化验证，存在回归风险。 - 影响：影响范围：涉及 6 个多模态模型处理器（MiniCPM-V4.6、LLaVA-OneVision2、FireRedLID、Qwen2-Audio、Ernie45-VL、Isaac、Moss），影响这些模型的输入处理逻辑，尤其是字段的 CPU/GPU 放置。对用户而言，如果这些模型在 CI 中受回归影响，回滚将修复其功能；对系统而言，消除了 CI 的不稳定因素。影响程度中等，但由于改动分散，需确保各模型回归测试通过。 - 风险标记：回滚未包含测试，异步传输引入并发风险，显存占用可能增加

关联脉络

- PR #52827 [Bugfix][Frontend] Run the serve arg checks for vllm launch too: 本 PR 是回滚由此 PR 引入的错误的 keep_on_cpu=True 变更，需关联。