PR #46806 完整报告

vllm-project/vllm

Remove mantis

合并时间：2026-07-01 15:13

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/46806

执行摘要

- 一句话：移除低使用率的 Mantis 多模态模型
- 推荐动作：此 PR 价值在于展示了基于数据驱动的技术债务清理方法。对于大多数开发者，无需深入阅读代码细节；但若需要了解如何安全移除模型架构，可作为参考。关注点：保持版本记录（registry.py 中添加了最后支持版本），确保用户可追踪兼容性。

功能与动机

此 PR 属于减少技术债务 sprint 的一部分，基于内部使用率追踪移除已废弃的模型架构。PR 描述指出 Mantis（以及 Grok）模型过去 3 个月使用不足 20 小时，经与 Tiger Research 的 Prof. Wenhu Chen 确认后移除。

实现拆解

1. 删除模型处理类：在 vllm/model_executor/models/llava.py 中移除了 MantisProcessingInfo、MantisMultiModalProcessor 及其 apply 方法和 get_replacement_mantis 辅助函数。这些类继承自 LLaVA 的处理类，仅用于 Mantis 特定的 prompt 格式（如 (image N: <Image>...)）。
2. 更新模型注册表：在 vllm/model_executor/models/registry.py 中从 _ArchitectureRegistry 中移除了 MantisForConditionalGeneration 条目（原本映射到 llava 模块），并在 _VLLM_VERSION_FOR_ARCH 字典中添加了 Mantis 的最后版本记录 0.24.0，表示该架构在后续版本被移除。
3. 删除示例和测试辅助代码：在 examples/generate/multimodal/vision_language_offline.py 中删除了 run_mantis 函数及其在 model_map 中的注册；在 tests/models/multimodal/generation/vlm_utils/model_utils.py 中删除了 mantis_vllm_to_hf_output 和 mantis_patch_hf_runner 函数。
4. 清理测试配置和 CI 配置：从 tests/models/multimodal/generation/test_common.py 中的 vlm_test_info 移除 Mantis 条目，从 tests/models/registry.py 移除 Mantis 的 _HfExamplesInfo。从三个 CI 配置文件（.buildkite/test_areas/models_multimodal.yaml、.buildkite/test-amd.yaml、.buildkite/intel_jobs/models_multimodal_intel.yaml）中移除 pip install mantis 行和相关的测试命令。
5. 更新文档：在 docs/models/supported_models.md 中将 Mantis 标记为已弃用（deprecated）状态。

关键文件：

- vllm/model_executor/models/llava.py（模块 模型执行器；类别 source；类型 data-contract；符号 MantisProcessingInfo, get_hf_processor, MantisMultiModalProcessor, apply）：核心删除变更：移除了 Mantis 模型处理类，包括 MantisProcessingInfo、MantisMultiModalProcessor 及其辅助函数，涉及约 97 行代码。
- tests/models/multimodal/generation/vlm_utils/model_utils.py（模块 测试工具；类别 test；类型 test-coverage；符号 mantis_vllm_to_hf_output, mantis_patch_hf_runner, _generate）：移除了 Mantis 专用的测试辅助函数，包括 mantis_vllm_to_hf_output 和 mantis_patch_hf_runner，涉及 32 行。
- examples/generate/multimodal/vision_language_offline.py（模块 示例；类别 source；类型 core-logic；符号 run_mantis）：删除了 Mantis 运行示例函数 run_mantis 及其在模型映射中的注册，涉及 23 行。
- vllm/model_executor/models/registry.py（模块 模型注册；类别 source；类型 data-contract）：从模型注册表中移除 MantisForConditionalGeneration 条目，并在版本追踪字典中添加最后支持版本 0.24.0。
- tests/models/multimodal/generation/test_common.py（模块 测试框架；类别 test；类型 test-coverage）：移除了 Mantis 的测试信息条目，包括模型列表、prompt 格式、tokenizer 后处理等。
- tests/models/registry.py（模块 测试注册；类别 test；类型 test-coverage）：移除了 Mantis 的 HF 示例信息，包括模型名称和 transformers 版本限制。
- .buildkite/test_areas/models_multimodal.yaml（模块 CI 配置；类别 config；类型 configuration）：移除了 Mantis 的测试命令和 pip install mantis 依赖。
- .buildkite/test-amd.yaml（模块 CI 配置；类别 config；类型 configuration）：移除了 AMD CI 中 Mantis 的 pip 安装依赖。
- .buildkite/intel_jobs/models_multimodal_intel.yaml（模块 CI 配置；类别 config；类型 configuration）：移除了 Intel GPU CI 中 Mantis 的测试命令，调整了并行步骤配置。
- docs/models/supported_models.md（模块 文档；类别 docs；类型 documentation）：将 Mantis 模型标记为 deprecated，减少文档中维护的信息。

关键符号：MantisProcessingInfo.get_hf_processor, MantisMultiModalProcessor.apply, get_replacement_mantis, mantis_vllm_to_hf_output, mantis_patch_hf_runner, run_mantis

关键源码片段

vllm/model_executor/models/llava.py

核心删除变更：移除了 Mantis 模型处理类，包括 MantisProcessingInfo、MantisMultiModalProcessor 及其辅助函数，涉及约 97 行代码。

```
# 以下类在 PR #46806 中被移除
class MantisProcessingInfo(LlavaProcessingInfo):
    def get_hf_processor(self, **kwargs: object):
        hf_config = self.get_hf_config()
```

```
        vision_info = self.get_vision_encoder_info()
        # 设置 patch_size 和 vision_feature_select_strategy 参数
        kwargs.setdefault("patch_size", vision_info.get_patch_size())
        kwargs.setdefault(
            "vision_feature_select_strategy",
            hf_config.vision_feature_select_strategy,
        )
        return self.ctx.get_hf_processor(LlavaProcessor, **kwargs)


class MantisMultiModalProcessor(LlavaMultiModalProcessor):
    def apply(
        self,
        inputs: ProcessorInputs,
        timing_ctx: TimingContext,
    ) -> MultiModalInput:
        hf_config = self.info.get_hf_config()
        image_token_id = hf_config.image_token_index
        # 假设不依赖图像尺寸
        num_image_tokens = self.info.get_num_image_tokens(
            image_width=-1,
            image_height=-1,
        )
        result = super().apply(inputs, timing_ctx)
        mm_item_counts = inputs.mm_data_items.get_all_counts()
        mm_kwargs = result["mm_kwargs"]
        mm_hashes = result["mm_hashes"]

        # 重新实现 MLlavaProcessor 中的 prompt 替换功能
        # 参考: https://github.com/TIGER-AI-Lab/Mantis.git
        def get_replacement_mantis(item_idx: int):
            return "".join(
                [
                    f"(image {item_idx + 1}: <Image>", # 7 个 token
                    "<image>" * num_image_tokens,
                    "</Image>)", # 3 个 token
                ]
            )

        mantis_mm_repls = self._bind_and_group_updates(
            [
                PromptReplacement(
                    modality="image",
                    target=[image_token_id] * num_image_tokens,
                    replacement=get_replacement_mantis,
                )
            ],
            mm_item_counts,
        )
```

```
        prompt_ids, _ = self._apply_prompt_updates(
            result["prompt_token_ids"],
            mantis_mm_repls,
        )
        # ... 剩余方法用于验证和返回 mm_input
        # 注意：该方法已从代码库中完全删除
```

tests/models/multimodal/generation/vlm_utils/model_utils.py

移除了 Mantis 专用的测试辅助函数，包括 mantis_vllm_to_hf_output 和 mantis_patch_hf_runner，涉及 32 行。

```
# 以下函数在 PR #46806 中被移除

def mantis_vllm_to_hf_output(vllm_output: RunnerOutput, model: str) -> RunnerOutput:
    """Sanitize vllm output [mantis] to compare with hf output."""
    output_ids, output_str, out_logprobs = vllm_output
    # Mantis 输出末尾需要添加 <|eot_id|> token
    hf_output_str = output_str + "<|eot_id|>"
    return output_ids, hf_output_str, out_logprobs


def mantis_patch_hf_runner(hf_model: HfRunner) -> HfRunner:
    from mantis.models.mllava import MLlavaProcessor
    # 替换 HF runner 的 processor 为 Mantis 自定义处理器
    hf_model.processor = MLlavaProcessor.from_pretrained(hf_model.model_name)
    orig_generate = hf_model.model.generate
    tokenizer = hf_model.processor.tokenizer

    def _generate(self, *args, **kwargs):
        # 在 generate 中额外加入 <|eot_id|> 作为 eos token
        return orig_generate(
            *args,
            **kwargs,
            eos_token_id=[
                tokenizer.eos_token_id,
                tokenizer.convert_tokens_to_ids("<|eot_id|>"),
            ],
        )

    hf_model.model.generate = types.MethodType(_generate, hf_model.model)
    return hf_model
```

评论区精华

此 PR 未引发实质性讨论。PR 提交后，由 DarkLight1337 和 jeejeelee 分别批准，无 review 评论。唯一评论来自 claude[bot] 表示来自 fork 的 PR 无法自动审查。

- 整体批准 (other): PR 被批准，无需修改。

风险与影响

- 风险：风险较低。移除的是独立模型架构，不共享核心逻辑。但存在以下考虑：
  - 兼容性风险：现有用户若仍依赖 Mantis 模型，升级后将无法使用，需改用其他替代或旧版本。
  - 测试覆盖丢失：移除了相关测试，但移除的是低使用率模型，影响很小。
  - 文档未及时反映：虽然文档已更新为 *deprecated*，但用户可能未及时看到。
  - 影响：
  - 用户影响：对绝大多数用户无影响。依赖 Mantis 的用户需要迁移或停止升级。
  - 系统影响：代码量减少约 200 行，降低了维护负担。
  - 团队影响：清理技术债务，减少后续开发中的干扰。
  - 风险标记：低风险清理，移除废弃模型，不影响 LLaVA 基础模型

关联脉络

- 暂无明显关联 PR