

PR #45161 完整报告

vllm-project/vllm

Deprecate Transformers v4 support

合并时间: 2026-06-11 11:04

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/45161>

执行摘要

- 一句话: 移除 Transformers v4 支持, 强制使用 v5
- 推荐动作: 此 PR 值得精读, 它展示了如何系统性地移除遗留依赖。关键设计决策是将弃用警告转为硬错误, 避免用户忽视。注意 review 中发现的类型提示问题, 警示在大型清理中需要全面检查相关常量。

功能与动机

如 PR 描述所述, Transformers v4 支持将在 vLLM 0.24.0 中完全移除。当前是从 main 分支切出 v0.24.0 版本的正确时机。本 PR 移除了安装 Transformers v4 的支持, 但对使用 Transformers v4 创建的模型应保持兼容性。这还允许在 [requirements/common.txt](#) 中使用更灵活的 Transformers 版本锁定, 以便用户可以安装新版本的 Transformers 以获得更好的 Day-0 模型支持。

实现拆解

1. 版本检查硬化: 在 `vllm/transformers_utils/config.py` 中, 将原有的 `try-except` 导入替换为直接导入 `ALLOWED_LAYER_TYPES`, 并将 Transformers 版本检查从 `logger.warning` 改为 `raise ImportError`, 使得使用 Transformers v4 时直接报错。
2. 移除兼容分支: 在多个配置文件和模型文件中删除针对 Transformers v4 的条件分支。例如 `vllm/model_executor/models/transformers/base.py` 中移除 `dtype` 传播逻辑和旧的权重映射处理; `vllm/model_executor/model_loader/weight_utils.py` 中删除 `enable_hf_transfer` 函数及相关条件, 统一使用 `enable_xet_high_performance`。
3. 清理模型特定兼容代码: 在 `qwen3_omni_moe_thinker.py`、`ultravox.py` 等模型中移除对 Transformers 版本号的运行时检查和针对旧版本的 hack 代码。
4. 简化配置类定义: 在 `deepseek_vl2.py`、`qwen3_5.py` 等配置文件中, 统一使用 Transformers v5 的 `strict` 装饰器, 删除 v4 的 `fallback` 路径。
5. 依赖约束调整: 更新 `requirements/common.txt`, 放宽 Transformers 版本限制, 不再固定为 v4 系列。

关键文件:

- `vllm/transformers_utils/config.py` (模块 配置层; 类别 `source`; 类型 `dependency-wiring`; 符号 `patched_validate_rope`): 核心配置入口, 修改导入方式并将 Transformers 版本检查从警告升级为硬错误, 直接影响所有用户启动行为。

- `vllm/model_executor/models/transformers/base.py` (模块 模型执行器; 类别 source; 类型 data-contract) : Transformer 模型基类, 移除了大量 Transformers v4 的兼容代码, 并引入了新的导入用于 Transformers v5 的权重重命名支持。
- `vllm/model_executor/model_loader/weight_utils.py` (模块 模型加载; 类别 source; 类型 data-contract; 符号 `enable_hf_transfer`) : 权重加载入口, 移除了 `hf_transfer` 的兼容代码, 统一启用 xet 高性能模式。
- `vllm/model_executor/models/qwen3_omni_moe_thinker.py` (模块 模型实现; 类别 source; 类型 data-contract) : 移除了针对 Transformers 版本的运行时检查和旧版本兼容代码, 简化音频处理逻辑。
- `vllm/transformers_utils/configs/deepseek_vl2.py` (模块 模型配置; 类别 source; 类型 core-logic; 符号 `DeepseekVLV2TextConfig`) : 简化了 `DeepseekVLV2TextConfig` 的定义, 统一使用 Transformers v5 的 `@strict` 装饰器。

关键符号: `patched_validate_rope`, `enable_xet_high_performance`, `_patch_config`, `_create_hf_to_vllm_mapper`, `DeepseekVLV2TextConfig`

关键源码片段

`vllm/transformers_utils/config.py`

核心配置入口, 修改导入方式并将 Transformers 版本检查从警告升级为硬错误, 直接影响所有用户启动行为。

```
# vllm/transformers_utils/config.py ( 关键变更 )

# 之前: 通过 try-except 兼容 v4/v5
# try:
# from transformers.configuration_utils import ALLOWED_ATTENTION_LAYER_TYPES
# except ImportError:
# from transformers.configuration_utils import (
#     ALLOWED_LAYER_TYPES as ALLOWED_ATTENTION_LAYER_TYPES,
# )

# 现在: 直接使用 Transformers v5 的导出
from transformers.configuration_utils import ALLOWED_LAYER_TYPES

# 启动时检查 Transformers 版本, v4 直接报错
if Version(version("transformers")) < Version("5.0.0"):
    raise ImportError(
        "Support for Transformers v4 is deprecated and was removed in vLLM v0.24.0. "
        "Please upgrade to Transformers v5: pip install --upgrade transformers"
    )
```

`vllm/model_executor/models/transformers/base.py`

Transformer 模型基类, 移除了大量 Transformers v4 的兼容代码, 并引入了新的导入用于 Transformers v5 的权重重命名支持。

```
# vllm/model_executor/models/transformers/base.py ( 关键变更 )
```

```

# 新增 Transformers v5 特有的导入
from transformers.conversion_mapping import (
    WeightRenaming,
    get_model_conversion_mapping,
)

class TransformersModel(VllmModel):
    def _patch_config(self):
        """不再需要为 v4 的 sub_config dtype 传播"""
        self.text_config._attn_implementation = "vllm"
        self.config.dtype = torch.get_default_dtype()
        # 移除以下 v4 兼容代码:
        # for sub_config_name in getattr(self.config, "sub_configs", {}):
        #     sub_config = getattr(self.config, sub_config_name)
        #     if sub_config.dtype != (dtype := self.config.dtype):
        #         sub_config.dtype = dtype

    def _create_hf_to_vllm_mapper(self):
        """统一使用 Transformers v5 的 WeightRenaming 机制"""
        self.hf_to_vllm_mapper = WeightsMapper()
        orig_to_new_regex = self.hf_to_vllm_mapper.orig_to_new_regex
        # 直接调用 v5 的 API, 移除 v4 的 _checkpoint_conversion_mapping 分支
        for mapping in get_model_conversion_mapping(self.model):
            if isinstance(mapping, WeightRenaming):
                compiled_sources = re.compile(
                    mapping.compiled_sources.pattern, mapping.compiled_sources.flags
                )
                target_pattern = mapping.target_patterns[0]
                orig_to_new_regex[compiled_sources] = target_pattern

```

vllm/model_executor/model_loader/weight_utils.py

权重加载入口, 移除了 hf_transfer 的兼容代码, 统一启用 xet 高性能模式。

```

# vllm/model_executor/model_loader/weight_utils.py ( 关键变更 )

# 移除了 enable_hf_transfer 函数
# def enable_hf_transfer():
#     """automatically activates hf_transfer"""
#     if "HF_HUB_ENABLE_HF_TRANSFER" not in os.environ:
#         try:
#             import hf_transfer
#             huggingface_hub.constants.HF_HUB_ENABLE_HF_TRANSFER = True
#         except ImportError:
#             pass

# 直接启用 xet 高性能模式 (Transformers v5 特性)
def enable_xet_high_performance():
    """automatically activates xet high performance mode"""
    if "HF_XET_HIGH_PERFORMANCE" not in os.environ:

```

```
huggingface_hub.constants.HF_XET_HIGH_PERFORMANCE = True
```

```
# 移除版本检查，无条件启用
# 之前: if hasattr(huggingface_hub.constants, "HF_XET_HIGH_PERFORMANCE"):
# enable_xet_high_performance()
# else:
# enable_hf_transfer()
# 现在:
enable_xet_high_performance()
```

评论区精华

Review 中主要有一处讨论: tlrchlsmth 指出在 `vllm/model_executor/models/transformers/`
`utils.py` 的 `Style` 类型字面量中仍然保留 `colwise_rep` 和 `rowwise_rep`, 但对应的映射已被移
除, 可能导致类型检查通过但实际静默回退到默认行为。作者已通过后续 commit 修复此问题。

- 类型提示未清理可能导致静默行为变化 (correctness): 作者在后续 commit 中修复, 可能是移除了这两个字面量。review 已批准。

风险与影响

- 风险:
 1. 依赖硬切换风险: 用户如果仍在使用 Transformers v4, 启动 vLLM 时会直接因 `ImportError` 退出, 需要明确告知用户升级。虽然这是预期行为, 但可能影响尚未来得及升级的用户。
 2. 潜在兼容性问题: 部分自定义模型可能依赖已被移除的 v4 兼容代码 (如 `_patch_hf_transformers_validate_rope` 中的版本检查), 但这些代码已在多个版本中标记为弃用, 风险较低。
 3. 类型提示残留: review 中发现的 `Style` 类型字面量未同步清理, 虽已修复, 但类似问题可能在其它文件中存在。- 影响: 所有使用 vLLM 的用户必须将 Transformers 升级到 v5 ($\geq 5.0.0$), 否则无法运行。对内部开发而言, 代码库显著简化, 移除了大量条件分支, 降低了维护成本。`requirements/common.txt` 中 Transformers 版本限制放宽, 用户可自由安装新版 Transformers 以支持新模型。- 风险标记: 依赖硬切换, 潜在兼容性问题

关联脉络

- 暂无明显关联 PR