PR #40982 完整报告

vllm-project/vllm

[DSV4] Support `max` reasoning effort

合并时间：2026-04-29 19:03

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/40982

执行摘要

- 一句话：支持 DeepSeek V4 的 max reasoning effort 及模型特定值传递
- 推荐动作：值得精读，展示了模型特定参数传递与 OpenAI 兼容性之间的设计权衡，以及在多模型系统中如何优雅地扩展枚举值。建议关注 Harmony parser 对 none 的处理，确保后续客户端升级无虞。

功能与动机

为了让 DeepSeek V4 用户能够直接使用 reasoning_effort="max"，而不是通过 chat_template_kwargs 传递，同时保持与 OpenAI API 的兼容性，并明确标注 max 是 DS V4 特有值。

实现拆解

1. 更新协议定义：在 vllm/entrypoints/openai/chat_completion/protocol.py 中将 reasoning_effort 字段类型从固定的 Literal 扩展为包含 minimal、xhigh、max 等值的 Literal，并添加详细描述说明 max 是 DS V4 特有。
2. 更新 DeepSeek V4 Tokenizer 映射逻辑：在 vllm/tokenizers/deepseek_v4.py 中修改 apply_chat_template 的推理 effort 处理逻辑：支持 none 禁用思考、xhigh 映射到 max、其他值（如 minimal）映射到 high。
3. 更新 Harmony Parser 类型和校验：在 vllm/entrypoints/openai/parser/harmony_utils.py 中将 get_system_message 的参数类型从 Literal["high","medium","low"] 改为 str | None，并添加运行时校验，确保只接受已知的 effort 值（high/medium/low），否则抛出明确的 ValueError。
4. 添加测试覆盖：在 tests/tokenizers_/test_deepseek_v4.py、tests/entrypoints/openai/chat_completion/test_chat.py 和 tests/entrypoints/openai/parser/test_harmony_utils.py 中增加针对新 effort 值的测试用例，包括 none 禁用思考、xhigh 映射、未知值拒绝等场景。

关键文件：

- vllm/entrypoints/openai/chat_completion/protocol.py（模块 请求协议；类别 source；类型 core-logic；符号 ChatCompletionRequest）：核心协议定义，扩展 reasoning_effort 字段支持模型特定值，是变更的入口点。

- vllm/tokenizers/deepseek_v4.py（模块 Tokenizer；类别 source；类型 core-logic；符号 _DeepseekV4Tokenizer.apply_chat_template）：DS V4 tokenizer 的核心逻辑变更，映射 OpenAI effort 值到参考内部值。
- vllm/entrypoints/openai/parser/harmony_utils.py（模块 话术解析；类别 source；类型 dependency-wiring；符号 get_system_message）：Harmony parser 的类型和校验调整，确保与协议层兼容。
- tests/tokenizers_/test_deepseek_v4.py（模块 测试；类别 test；类型 test-coverage；符号 test_deepseek_v4_none_reasoning_effort_disables_thinking, test_deepseek_v4_maps_compatible_thinking_reasoning_effort_values, test_deepseek_v4_maps_xhigh_to_reference_max_reasoning_effort）：添加了针对新 effort 值的测试，包括 none、xhigh、minimal 等，覆盖映射和 fallback 行为。
- tests/entrypoints/openai/chat_completion/test_chat.py（模块 测试；类别 test；类型 test-coverage；符号 test_chat_completion_request_accepts_model_specific_reasoning_effort, test_chat_completion_request_rejects_unknown_reasoning_effort）：测试协议层接受模型特定 effort 值和拒绝未知值。
- tests/entrypoints/openai/parser/test_harmony_utils.py（模块 测试；类别 test；类型 test-coverage；符号 test_unsupported_reasoning_effort_raises_clear_error）：测试 Harmony parser 对不支持 effort 值的错误提示。

关键符号：get_system_message, _DeepseekV4Tokenizer.apply_chat_template, ChatCompletionRequest.init, test_deepseek_v4_maps_compatible_thinking_reasoning_effort_values, test_chat_completion_request_accepts_model_specific_reasoning_effort

关键源码片段

vllm/entrypoints/openai/chat_completion/protocol.py

核心协议定义，扩展 reasoning_effort 字段支持模型特定值，是变更的入口点。

```
# file: vllm/entrypoints/openai/chat_completion/protocol.py

class ChatCompletionRequest(OpenAIBaseModel):
    # ...
    reasoning_effort: (
        Literal["none", "minimal", "low", "medium", "high", "xhigh", "max"] | None
    ) = Field(
        default=None,
        description=(
            "Constrains effort on reasoning for reasoning models. "
            "Currently supported values are none, minimal, low, medium, "
            "high, xhigh, and max. Reducing reasoning effort can result in "
            "faster responses and fewer tokens used on reasoning in a response. "
            "Note that 'max' is specific to the DeepSeek V4 series and is not "
            "part of the standard OpenAI API specification."
        ),
    )
    # ...
```

vllm/tokenizers/deepseek_v4.py

DS V4 tokenizer 的核心逻辑变更，映射 OpenAI effort 值到参考内部值。

```
# file: vllm/tokenizers/deepseek_v4.py

class _DeepseekV4Tokenizer(tokenizer.__class__):
    def apply_chat_template(self, messages, tools=None, **kwargs):
        # ... existing logic ...

        reasoning_effort = kwargs.get("reasoning_effort")
        # 处理 none 禁用思考，xhigh 映射为 max，其他未知值设为 high
        if not isinstance(reasoning_effort, str):
            reasoning_effort = None
        elif reasoning_effort == "none":
            thinking_mode = "chat"
            reasoning_effort = None
        elif reasoning_effort in ("max", "xhigh"):
            reasoning_effort = "max"
        else:
            reasoning_effort = "high"

        encode_config = dict(
            thinking_mode=thinking_mode,
            drop_thinking=kwargs.get("drop_thinking", True),
            reasoning_effort=reasoning_effort,
        )
        prompt_str = encode_messages(messages, **encode_config)
        # ... rest of logic ...
```

vllm/entrypoints/openai/parser/harmony_utils.py

Harmony parser 的类型和校验调整，确保与协议层兼容。

```
# file: vllm/entrypoints/openai/parser/harmony_utils.py

REASONING_EFFORT = {
    "high": ReasoningEffort.HIGH,
    "medium": ReasoningEffort.MEDIUM,
    "low": ReasoningEffort.LOW,
}

def get_system_message(
    model_identity: str | None = None,
    reasoning_effort: str | None = None, # 放宽类型以支持任意字符串
    start_date: str | None = None,
    # ...
) -> Message:
    sys_msg_content = SystemContent.new()
    # ...
    if reasoning_effort is not None:
```

```
        # 校验：只有 high/medium/low 是 Harmony 支持的
        if reasoning_effort not in REASONING_EFFORT:
            supported_values = ", ".join(REASONING_EFFORT)
            raise ValueError(
                f"reasoning_effort={reasoning_effort!r} is not supported by "
                f"Harmony. Supported values are: {supported_values}."
            )
        sys_msg_content = sys_msg_content.with_reasoning_effort(
            REASONING_EFFORT[reasoning_effort]
        )
    # ...
```

评论区精华

1. 类型选择争议：reviewer chaunceyjiang 建议保留 Literal["none","low","medium","high"] 以保持文档清晰，但作者 BugenZhao 认为该字段应该模型无关，最终采用包含所有已知值的 Literal 并附加 description，平衡了文档性和扩展性。
2. Harmony Parser 的校验必要性：reviewer gemini-code-assist[bot] 指出如果传入 "none"，Harmony parser 会抛出 ValueError，建议处理为 no-op（等效于 None），避免破坏现有客户端；作者回应由于全局校验缺失，需要在各模型渲染器中单独校验。
3. DS V4 默认启用思考：reviewer zifeitong 提到 DeepSeek 官方 API 默认启用思考，因此 reasoning_effort="none" 必须能关闭思考；作者已在 tokenizer 逻辑中处理。
4. 未知值的 catch-all 行为： reviewer zifeitong 建议将 catch-all 设为 "max" 以便 minimal 映射到 high，但作者最终按与 OpenAI 兼容的映射将未知值设为 "high"，与官方行为一致。

- reasoning_effort 字段类型是保持 Literal 还是放宽为 str (design): 采用包含 none/minimal/low/medium/high/xhigh/max 的 Literal，且 description 说明 max 是 DS V4 特有。
- Harmony parser 对 none 的处理 (correctness): 当前在 harmony_utils.py 中未默认处理 none，会抛出错误；DS V4 tokenizer 中已处理 none 禁用思考。后续可考虑在 harmony_utils 中增加 none 的静默处理。
- DS V4 默认启用思考与 none 关闭思考 (correctness): 已在 tokenizer 逻辑中处理，none 将 thinking_mode 设为 chat 并清除 reasoning_effort。
- 未知值 catch-all 应该映射为 high 还是 max (design): 将未知值（如 minimal）映射为 high，与 OpenAI 兼容。

风险与影响

- 风险：
  1. 兼容性风险：之前 reasoning_effort 是固定 Literal，现在允许更多值，但旧的客户端如果发送未知值（如 "extra_high"），在协议层会通过校验，但在 DeepSeek tokenizer 中会被映射为 "high"（安全），而在 Harmony parser 中会抛出 ValueError（需注意 Harmony 用户）。风险等级：低。
  2. Harmony parser 未处理 none：如果 Harmony parser 收到 "none" 会报错，但该场景只发生在模型使用 Harmony 且客户端显式传递 "none"（之前允许）。已通过测试提醒，

但尚未在 harmony_utils 中默认处理（仅校验）。当前行为是抛出错误，可能影响 Hybrid Harmony + DS V4 用户。风险等级：中，建议后续修复。

3. DS V4 tokenizer 行为变化：之前 "none" 被当作无效值设为 None（相当于 high），现在明确禁用思考；之前 "xhigh" 被当作无效值，现在映射为 max。这是预期行为，但需注意对已有工作流的影响。风险等级：低。 - 影响：影响所有使用 DeepSeek V4 且设置 reasoning_effort 的客户端，以及使用 Harmony parser 的模型。对于 DS V4 用户，现在可以直接使用 max 而无需额外配置；对于 Harmony 用户，传入非高 / 中 / 低的值会收到明确错误。整体影响范围中等，但改进明确。 - 风险标记：Harmony parser 未处理 none 值导致报错，字段类型放宽可能导致隐式错误，DS V4 行为变化影响现有工作流

关联脉络

- PR #41198 [Bugfix] DSV32/V4 add missing type conversion for non-streaming tool calls: 同为 DeepSeek V4 相关 bugfix，涉及工具调用和类型转换，两者共同完善 DS V4 支持。
- PR #34668 [Reasoning] Support for speculative decoding with thinking budget: 涉及 reasoning_effort 和 thinking budget，与本文档的 effort 扩展有关联，共同构建 reasoning 参数体系。