PR #43401 完整报告

vllm-project/vllm

[Bugfix] Map reasoning_effort to enable_thinking in chat template kwargs

合并时间：2026-05-27 20:39

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/43401

执行摘要

- 一句话：修复 Gemma4 Responses API 思考未启用
- 推荐动作：值得合并。设计简洁，尊重用户显式设置，且通过 resolve_chat_template_kwargs 天然过滤不兼容模型的 kwarg，安全无侵入。文档同步更新清晰。

功能与动机

Issue #43395 报告：Gemma4 模型在 Responses API 中无法显示推理输出，即使传递了 reasoning: {"effort": "high"}。根因是 Gemma4 的聊天模板需要显式设置 enable_thinking=True，而 Responses API 仅传递了 reasoning_effort 未做转换。

实现拆解

1. 修改 vllm/entrypoints/openai/chat_completion/protocol.py：在 ChatCompletionRequest.build_chat_params 中，当 self.reasoning_effort 不为 None 且用户未显式传入 enable_thinking 时，根据 reasoning_effort 是否为 "none" 注入 enable_thinking。
2. 修改 vllm/entrypoints/openai/responses/protocol.py：在 ResponsesRequest.build_chat_params 中执行相同逻辑，从 self.reasoning 中提取 reasoning_effort。
3. 新增测试文件 tests/entrypoints/openai/test_reasoning_enable_thinking.py：覆盖 Chat Completions 和 Responses API 的各种 reasoning_effort 输入，包括有效值、"none"、未设置和用户显式覆盖。
4. 更新文档 docs/features/reasoning_outputs.md：新增“Automatic enable_thinking Activation”章节，说明自动映射行为，并将 Gemma4 加入支持列表。

关键文件：

- tests/entrypoints/openai/test_reasoning_enable_thinking.py（模块 测试；类别 test；类型 test-coverage；符号 _build_chat_request, _build_responses_request, TestChatCompletionReasoningEffort, test_non_none_effort_injects_enable_thinking_true）：新增测试文件，系统覆盖所有映射场景，验证核心逻辑正确性。
- vllm/entrypoints/openai/responses/protocol.py（模块 API 协议；类别 source；类型 core-logic）：核心修改文件之一，在 Responses API 的 build_chat_params 中注入

enable_thinking。

- vllm/entrypoints/openai/chat_completion/protocol.py（模块 API 协议；类别 source；类型 core-logic）：另一个核心修改文件，Chat Completions API 的 build_chat_params 执行相同逻辑。
- docs/features/reasoning_outputs.md（模块 文档；类别 docs；类型 documentation）：更新文档解释自动映射行为，确保用户了解新特性。

关键符号：ChatCompletionRequest.build_chat_params, ResponsesRequest.build_chat_params

关键源码片段

vllm/entrypoints/openai/responses/protocol.py

核心修改文件之一，在 Responses API 的 build_chat_params 中注入 enable_thinking。

```
# vllm/entrypoints/openai/responses/protocol.py ( 部分 )
def build_chat_params(
    self,
    default_template: str | None,
    default_template_content_format: ChatTemplateContentFormatOption,
) -> ChatParams:
    from .utils import should_continue_final_message

    continue_final = should_continue_final_message(self.input)
    reasoning = self.reasoning
    reasoning_effort = None if reasoning is None else reasoning.effort

    extra_kwargs: dict[str, Any] = dict(
        add_generation_prompt=not continue_final,
        continue_final_message=continue_final,
        reasoning_effort=reasoning_effort,
    )

    # 当请求推理时，自动激活 enable_thinking
    # 对于需要显式 opt-in 的模型（如 Gemma4，默认 enable_thinking=False）
    # 对于未声明该变量的模板，resolve_chat_template_kwargs 会无害过滤
    user_kwargs = self.chat_template_kwargs or {}
    if reasoning_effort is not None and "enable_thinking" not in user_kwargs:
        # 非 "none" 的 effort 注入 True，"none" 注入 False
        extra_kwargs["enable_thinking"] = reasoning_effort != "none"

    return ChatParams(
        chat_template=default_template,
        chat_template_content_format=default_template_content_format,
        chat_template_kwargs=merge_kwargs(
            self.chat_template_kwargs,
            extra_kwargs,
        ),
```

```
            media_io_kwargs=self.media_io_kwargs,
        )
```

vllm/entrypoints/openai/chat_completion/protocol.py

另一个核心修改文件，Chat Completions API 的 build_chat_params 执行相同逻辑。

```
    # vllm/entrypoints/openai/chat_completion/protocol.py ( 部分 )
    def build_chat_params(
        self,
        default_template: str | None,
        default_template_content_format: ChatTemplateContentFormatOption,
    ) -> ChatParams:
        extra_kwargs: dict[str, Any] = dict(
            add_generation_prompt=self.add_generation_prompt,
            continue_final_message=self.continue_final_message,
            documents=self.documents,
            reasoning_effort=self.reasoning_effort,
        )

        # 自动注入 enable_thinking，与 Responses API 保持一致
        user_kwargs = self.chat_template_kwargs or {}
        if self.reasoning_effort is not None and "enable_thinking" not in user_kwargs:
            extra_kwargs["enable_thinking"] = self.reasoning_effort != "none"

        return ChatParams(
            chat_template=self.chat_template or default_template,
            chat_template_content_format=default_template_content_format,
            chat_template_kwargs=merge_kwargs(
                self.chat_template_kwargs,
                extra_kwargs,
            ),
            media_io_kwargs=self.media_io_kwargs,
        )
```

评论区精华

- Reviewer chaunceyjiang 指出（评论在 PR #43401 上）：本 PR 并非 Gemma4 专属，所有模型都会受到影响，因此需要更新文档解释 reasoning_effort 会自动启用 enable_thinking。
- 作者响应：接受建议并更新了文档（commit fb8691c），将描述泛化为模型的无害过滤，并在 reviewer 建议下补充了 DeepSeek-V4-Pro 等示例。
- 讨论结论：评审人点赞并 approve。
- 泛化性讨论：PR 是否特定于 Gemma4 (question): 作者同意并更新了文档，将说明泛化，并添加 DeepSeek-V4-Pro 等示例。
- 文档改进：添加 DeepSeek-V4-Pro 和 Anthropic Messages API (documentation): 作者在 commit 7633dd3 中采纳了这些建议。

风险与影响

- 风险：低风险。变更局限在 request 级别的 build_chat_params 方法，仅当 reasoning_effort 被设置且用户未显式提供 enable_thinking 时才注入。下游 resolve_chat_template_kwargs 会自动过滤模型模板未声明的 kwarg，因此对不支持 enable_thinking 的模型（如 DeepSeek R1）无影响。添加了全面的单元测试覆盖边界情况。
- 影响：
  - 用户 / 模型: Gemma4 用户现在可以正常通过 Responses API 获得推理输出。Chat Completions 用户的行为也保持一致（之前可能也未自动注入，但 Chat API 部分模型可能已通过其他方式工作，现在统一）。
  - 系统: 无性能影响，仅增加少量条件判断。
  - 团队: 维护成本低，测试覆盖齐全。
  - 风险标记：核心路径变更（build_chat_params），用户显式设置不受影响，未知模板兼容性好（依赖下游过滤）

关联脉络

- PR #43395 Issues #43395: Responses API does not surface reasoning output with --reasoning-parser gemma4: 直接关联的 issue，描述了 bug 和复现步骤，是此 PR 的动机。
- PR #38100 PR #38100 (open): maps reasoning_effort="none" → enable_thinking=False for Qwen3: 相关但未合并的 PR，与本 PR 互补：本 PR 激活 thinking，该 PR 禁用 thinking。本 PR 的 reasoning_effort="none" 处理方式与其一致。