PR #49498 完整报告

vllm-project/vllm

[Frontend] Add cache_salt support to Anthropic Messages API

合并时间：2026-08-01 15:36

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/49498

执行摘要

- 一句话：Anthropic Messages API 新增 cache_salt 透传支持
- 推荐动作：值得快速阅读。作为前端 API 字段透传的最小闭环样例，展示了如何让 Anthropic 协议复用 OpenAI 内部请求模型的校验逻辑，避免重复实现。关注点：字段描述与安全建议的对齐、CountTokens 分支的有意排除，以及透传方式与 kv_transfer_params 的一致性。

功能与动机

Issue #46688 的核心诉求是：cache_salt 参数在 OpenAI 兼容 API 中已可用于显式前缀缓存隔离，而 Anthropic Messages API 端点缺失该参数，用户只能被迫从 Anthropic 格式切换到 OpenAI 格式，这在深度集成 Anthropic schema（tool calling、system prompt 处理等）的多租户系统中意味着大量应用层重构。PR body 也明确说明目标是让 /v1/messages 接受 cache_salt，并镜像 OpenAI 路由的引擎行为。

实现拆解

1. 协议层扩展（vllm/entrypoints/anthropic/protocol.py）：在 AnthropicMessagesRequest 的 vLLM 扩展字段区（与 kv_transfer_params、ec_transfer_params 并列）新增 cache_salt: str | None 字段，默认 None，描述文案与 OpenAI chat 字段保持一致，说明其用于多用户环境下前缀缓存的随机化隔离。
2. 请求转换透传（vllm/entrypoints/anthropic/serving.py）：在 _build_base_request 的 Messages 分支构造 ChatCompletionRequest 时新增 cache_salt=anthropic_request.cache_salt，让字段进入 engine 的统一请求模型。该模型已自带 check_cache_salt_support 校验器，因此校验行为与 OpenAI 路由完全一致；CountTokens 分支不涉及生成与缓存隔离，保持不动。
3. 测试补充（tests/entrypoints/anthropic/test_anthropic_messages_conversion.py）：新增 TestCacheSalt 两个用例，验证 cache_salt 能透传至转换后的 ChatCompletionRequest、省略时保持 None 默认值。

关键文件：

- vllm/entrypoints/anthropic/protocol.py（模块 请求协议；类别 source；类型 core-logic；符号 AnthropicMessagesRequest）：定义请求 schema，新增 cache_salt 字段及安全描述，是功能入口。

- vllm/entrypoints/anthropic/serving.py（模块 服务转换；类别 source；类型 core-logic；符号 _build_base_request）：转换层透传字段，使 cache_salt 进入内部 ChatCompletionRequest 并复用既有校验。
- tests/entrypoints/anthropic/test_anthropic_messages_conversion.py（模块 转换测试；类别 test；类型 test-coverage；符号 TestCacheSalt, test_cache_salt_passed_through, test_cache_salt_defaults_to_none）：新增 TestCacheSalt 验证透传与默认值，保证行为没有回归。

关键符号：_build_base_request

关键源码片段

vllm/entrypoints/anthropic/protocol.py

定义请求 schema，新增 cache_salt 字段及安全描述，是功能入口。

```
class AnthropicMessagesRequest(BaseModel):
    """Anthropic Messages API 请求模型。"""

    # ... 其余字段 ...

    # cache_salt 为 vLLM 扩展字段，不在 Anthropic 官方规范中。
    # 用途：对前缀缓存加盐，实现多用户环境下的缓存隔离，
    # 防止攻击者通过缓存命中率猜测其他用户的提示词。
    cache_salt: str | None = Field(
        default=None,
        description=(
            "If specified, the prefix cache will be salted with the provided "
            "string to prevent an attacker to guess prompts in multi-user "
            "environments. The salt should be random, protected from "
            "access by 3rd parties, and long enough to be "
            "unpredictable (e.g., 43 characters base64-encoded, corresponding "
            "to 256 bit)."
        ),
    )

    # 已有的 vLLM 扩展字段，cache_salt 的透传模式与之一致
    kv_transfer_params: dict[str, Any] | None = Field(
        default=None,
        description="KVTransfer parameters used for disaggregated serving.",
    )
```

vllm/entrypoints/anthropic/serving.py

转换层透传字段，使 cache_salt 进入内部 ChatCompletionRequest 并复用既有校验。

```
@classmethod
def _build_base_request(
    cls,
    anthropic_request: AnthropicMessagesRequest | AnthropicCountTokensRequest,
    openai_messages: list[dict[str, Any]],
```

```
) -> ChatCompletionRequest:
    """构建基础 ChatCompletionRequest，Anthropic 请求中的字段在此统一映射。"""
    if isinstance(anthropic_request, AnthropicCountTokensRequest):
        # CountTokens 只统计 token，不涉及生成与缓存隔离，因此不透传 cache_salt
        return ChatCompletionRequest(
            model=anthropic_request.model,
            messages=openai_messages,
            chat_template_kwargs=anthropic_request.chat_template_kwargs,
        )

    # Messages 生成路径：所有 vLLM 扩展字段（cache_salt、kv_transfer_params 等）
    # 在此透传给内部请求模型，后续校验与 engine 行为与 OpenAI 路由完全一致
    return ChatCompletionRequest(
        model=anthropic_request.model,
        messages=openai_messages,
        max_tokens=anthropic_request.max_tokens,
        # 同时映射到 OpenAI 的 max_completion_tokens 字段
        max_completion_tokens=anthropic_request.max_tokens,
        stop=anthropic_request.stop_sequences,
        temperature=anthropic_request.temperature,
        top_p=anthropic_request.top_p,
        top_k=anthropic_request.top_k,
        cache_salt=anthropic_request.cache_salt,
        kv_transfer_params=anthropic_request.kv_transfer_params,
        ec_transfer_params=anthropic_request.ec_transfer_params,
        chat_template_kwargs=anthropic_request.chat_template_kwargs,
    )
```

评论区精华

Review 交互较少：claude[bot] 因 PR 来自 fork 而跳过自动化 review，提示维护者可通过 @claude review 触发一次性 review；AndreasKaratzas 在 issue 中表示“PR looks alright, but since I dont see any comments on the RFC, I will wait for @DarkLight1337 to give the go/no-go.”，即 PR 本身没有技术问题，但希望等维护者对 RFC 表态；最终 DarkLight1337 批准（APPROVED）。未出现实质性技术争议。

- RFC go/no-go 等待 (question): DarkLight1337 最终 APPROVE 该 PR，问题关闭。
- fork 分支自动 review 被禁用 (other): 未触发额外 bot review，由维护者人工批准。

风险与影响

- 风险：风险整体较低： 1) 兼容性：字段默认 None，未指定时行为不变，既有客户端无感知。 2) 安全语义：cache_salt 涉及前缀缓存隔离，若 salt 泄露或被猜测，可能影响多租户隔离；协议层字段描述已保留 OpenAI 侧的安全建议（随机、保密、足够长），实际安全性由调用方保证。 3) 测试未本地运行：作者声明无 CUDA/torch 环境，仅通过 py_compile，准确性依赖 CI 执行，但改动为纯透传，测试用例直接断言转换结果，可执行性风险小。 4) 校验一致性：依赖 ChatCompletionRequest.check_cache_salt_support 存在且行为与 OpenAI 路由一致，若该 validator 调整需同步关注。

- 影响：影响范围：Anthropic Messages API（/v1/messages）的请求协议与转换层，面向使用 Anthropic schema 的多租户 / 多用户部署。对现有用户无破坏性影响（默认 None）。对团队而言，此改动确立了 Anthropic 端新增 vLLM 扩展字段的透传模式（与 kv_transfer_params 相同），后续扩展字段可参考同一路径。影响程度较小、收益明确。
- 风险标记：新增 API 字段，安全语义，依赖 CI 验证

关联脉络

- 暂无明显关联 PR