PR #44512 完整报告

vllm-project/vllm

[Frontend] Consolidate scale out entrypoints

合并时间：2026-06-29 18:18

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/44512

执行摘要

- 一句话：整合 scale-out 入口点，分离 Render/Derender API
- 推荐动作：值得精读：展示了如何通过将紧密耦合的模块拆分为独立模块，并统一工厂化初始化的设计模式。尤其适合需要重构入口层代码的团队参考。建议关注 factories.py 的工厂函数设计和 api_server.py 的初始化流程变更。

功能与动机

PR 背景：为避免 'Disaggregated Everything' 与 Disaggregated P/D Inference 混淆，需要将 Tokens IN/OUT、Renderer、Derenderer APIs 统一归类到 'scale out entrypoints' 下，而不是分散在 'serve' 或 'disagg' 中。作者在 PR body 和评论中说明这是为了保持相关 API 在一起，同时解决命名分歧。

实现拆解

1. 目录迁移与创建新模块：将 vllm/entrypoints/serve/render/ 和 vllm/entrypoints/serve/disagg/ 中的文件移动到 vllm/entrypoints/scale_out/，并新建 derender/ 子目录，将原本耦合在 ServingRender 中的 derender 逻辑抽离为独立的 ServingDerender。
2. 重构 ServingRender：在 scale_out/render/serving.py 中，ServingRender 移除了 OnlineDerenderer 依赖和所有 derender_* 方法，仅保留渲染功能。
3. 新建 ServingDerender：在 scale_out/derender/serving.py 中新增 ServingDerender 类，实现 derender_chat_response 和 derender_completion_response 方法，将原先散落的反渲染逻辑集中。
4. 分离 API 路由：从 scale_out/render/api_router.py 中删除 derender_chat_completion、derender_completion 端点以及 attach_router，在 scale_out/derender/api_router.py 中重新实现这些端点，并挂载到独立的 APIRouter 实例上。
5. 统一初始化与注册：新增 scale_out/factories.py，提供 init_render_state、init_scale_out_state 和 register_scale_out_api_routers 函数，被 api_server.py 和生成接口的 init_app_state、init_render_app_state 调用，替代原先直接初始化 ServingRender 和张贴路由的方式。
6. 调整入口点：修改 vllm/entrypoints/openai/api_server.py，移除 ServingRender 的直接导入和初始化，改为调用 factories.init_scale_out_state 和

register_scale_out_api_routers；同时更新导入路径。

7. 测试与示例：同步迁移和更新测试文件（如 test_mm_serde.py、test_chat_error.py），并更新 examples/scale_out/ 中的示例代码。

关键文件：

- vllm/entrypoints/scale_out/derender/serving.py（模块 反渲染；类别 source；类型 dependency-wiring；符号 ServingDerender, init, derender_chat_response, derender_completion_response）：新增核心文件，实现独立的反渲染服务 ServingDerender，承载 derender_chat_response 和 derender_completion_response 方法，是模块拆分的关键产出。
- vllm/entrypoints/scale_out/derender/api_router.py（模块 反渲染路由；类别 source；类型 entrypoint；符号 derender, derender_chat_completion, derender_completion）：新增文件，定义独立的 derender 端点 (POST /v1/chat/completions/derender 和 /v1/completions/derender)，并挂载到单独的路由器上，实现与 render 端点的彻底分离。
- vllm/entrypoints/scale_out/render/api_router.py（模块 渲染路由；类别 source；类型 rename-or-move；符号 derender_chat_completion, derender_completion, attach_router）：重命名文件，从 render 路由中删除 derender 端点，仅保留 render 端点，并调整导入路径，完成职责分离。
- vllm/entrypoints/scale_out/factories.py（模块 初始化工厂；类别 source；类型 core-logic；符号 init_render_state, init_scale_out_state, register_scale_out_api_routers）：新增工厂函数文件，统一管理 scale_out 各模块的初始化（ServingRender、ServingDerender、ServingTokens）和 API 路由注册，是初始化的总入口。
- vllm/entrypoints/openai/api_server.py（模块 服务器入口；类别 source；类型 entrypoint）：修改文件，移除 ServingRender 的直接初始化，改为调用 factories 的函数，是重构后的集成点。

关键符号：ServingDerender.init, ServingDerender.derender_chat_response, ServingDerender.derender_completion_response, init_render_state, init_scale_out_state, register_scale_out_api_routers, derender (in api_router), derender_chat_completion (handler), derender_completion (handler)

关键源码片段

vllm/entrypoints/scale_out/render/api_router.py

重命名文件，从 render 路由中删除 derender 端点，仅保留 render 端点，并调整导入路径，完成职责分离。

```
# vllm/entrypoints/scale_out/render/api_router.py ( 从 serve/render/ 迁移 )
# 移除了原先的 derender 端点和 attach_router 函数，仅保留 render 相关端点。

from http import HTTPStatus
from fastapi import APIRouter, Depends, Request
from fastapi.responses import JSONResponse
from vllm.entrypoints.openai.chat_completion.protocol import ChatCompletionRequest
```

```
from vllm.entrypoints.openai.completion.protocol import CompletionRequest
from vllm.entrypoints.openai.engine.protocol import ErrorResponse
from vllm.entrypoints.serve.utils.api_utils import validate_json_request
from vllm.logger import init_logger

# 调整导入路径，使用相对导入
from ..token_in_token_out.protocol import GenerateRequest
from .serving import ServingRender

logger = init_logger(__name__)
router = APIRouter()


def render(request: Request) -> ServingRender | None:
    return getattr(request.app.state, "serving_render", None)


# 仅保留 Render 端点
@router.post(
    "/v1/chat/completions/render",
    dependencies=[Depends(validate_json_request)],
    response_model=GenerateRequest,
    ...
)
async def render_chat_completion(request: ChatCompletionRequest, raw_request: Request):
    # 处理逻辑（与之前相同，只是 import 路径变了）
    ...

# derender 端点已完全删除，移至 scale_out/derender/api_router.py
```

vllm/entrypoints/scale_out/factories.py

新增工厂函数文件，统一管理 scale_out 各模块的初始化（ServingRender、ServingDerender、ServingTokens）和 API 路由注册，是初始化的总入口。

```
# vllm/entrypoints/scale_out/factories.py

from argparse import Namespace
from typing import TYPE_CHECKING

from fastapi import FastAPI
from vllm.engine.protocol import EngineClient
from vllm.tasks import SupportedTask

if TYPE_CHECKING:
    from starlette.datastructures import State
    from vllm.entrypoints.serve.utils.request_logger import RequestLogger
else:
    RequestLogger = object
```

```
def init_render_state(
    state: "State",
    request_logger: RequestLogger | None,
):
    """初始化 render 和 derender 的 serving 实例，挂载到 app.state。"""
    from .derender.serving import ServingDerender
    from .render.serving import ServingRender

    # 创建 ServingRender 实例
    state.serving_render = ServingRender(
        state.openai_serving_models,
        state.online_renderer,
        request_logger=request_logger,
    )

    # 创建独立的 ServingDerender 实例
    state.serving_derender = ServingDerender(
        state.openai_serving_models,
        state.online_derenderer,
        request_logger=request_logger,
    )


def init_scale_out_state(
    state: "State",
    args: "Namespace",
    engine_client: "EngineClient",
    request_logger: RequestLogger | None,
):
    """初始化 scale_out 模块的全部状态（包含 render 和 token-in/out）。"""
    init_render_state(state, request_logger)

    from vllm.entrypoints.scale_out.token_in_token_out.serving import ServingTokens

    state.serving_tokens = ServingTokens(
        engine_client,
        state.openai_serving_models,
        state.online_renderer,
        request_logger=request_logger,
        return_tokens_as_token_ids=args.return_tokens_as_token_ids,
        enable_prompt_tokens_details=args.enable_prompt_tokens_details,
        enable_log_outputs=args.enable_log_outputs,
        force_no_detokenize=args.tokens_only,
    )


def register_scale_out_api_routers(
    app: FastAPI,
    supported_tasks: tuple["SupportedTask", ...],
```

```
):
    """注册所有 scale-out API 路由（render 和 derender）。"""
    from .render.api_router import router as render_render
    app.include_router(render_render)

    from .derender.api_router import router as derender_render
    app.include_router(derender_render)

    if "generate" in supported_tasks:
        from .token_in_token_out.api_router import (
            attach_router as attach_disagg_router,
        )
        attach_disagg_router(app)
```

评论区精华

仅在目录命名上存在分歧：DarkLight1337 认为 disagg 也应属于 scale_out，但作者指出该目录已删除，最终达成一致。

- disagg 目录是否应属于 scale_out (design): 作者指出 disagg 已移除，双方认可。

风险与影响

- 风险：风险：1）大规模文件移动可能导致导入路径遗漏，部分内部模块或测试未及时更新。2）ServingDerender 独立后需确保与原有行为完全一致，避免回归。3）测试覆盖可能不全，特别是组合场景。4）外部代码若直接导入旧路径（如 vllm.entrypoints.serve.render.serving）会报错，但预计无外部依赖。
- 影响：对用户：无直接功能变化，所有 API 端点路径未变。对系统：代码结构更清晰，降低了 ServingRender 的复杂性和耦合度。对团队：后续添加新的 scale-out 功能更为方便，且避免了与拆分式推理的概念混淆。CI 受到影响，需要确保测试通过。
- 风险标记：大规模文件移动，导入路径变更，测试覆盖风险

关联脉络

- PR #22817 Disaggregated Everything: PR body 引用为其基础，提出 'Disaggregated Everything' 概念，本 PR 为此演进。
- PR #41907 未知标题（依据 body 引用）: PR body 引用此 issue/PR 作为前置工作。注：未在提供的历史中显示，但 body 提及。
- PR #44285 未知标题（依据 body 引用）: PR body 引用此 issue/PR 作为前置工作。注：未在提供的历史中显示，但 body 提及。